U N I V E R S I D A D N A C I O N A L
A U T Ó N O M A D E M É X I C O

F A C U L T A D D E C I E N C I A S

ESTRUCTURA DEL ADN POR MEDIO DEL JUEGO DEL CAOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN MATEMÁTICAS APLICADAS

P R E S E N T A:

BRENDA PAOLA QUINTANA SILVA

DIRECTORA DE TESIS:

DRA. LAURA CLEMENTINA ESLAVA FERNÁNDEZ

CIUDAD UNIVERSITARIA, CDMX, 2022

Hoja de Datos del Jurado

1. Datos del Alumno	2. Datos del Tutor
Quintana Silva Brenda Paola 77 13 51 55 67 Universidad Nacional Autónoma de México Facultad de Ciencias Matemáticas Aplicadas 314060707	Dra. Laura Clementina Eslava Fernández
3. Datos del Sinodal 1	**4. Datos del Sinodal 2**
Dra. María Clara Fittipaldi	Dra. Yalbi Itzel Balderas Martínez
5. Datos del Sinodal 3	**6. Datos del Sinodal 4**
Dra. Lizbeth Naranjo Albarrán	Dr. Jefferson Edwin King Dávalos
7. Datos del Trabajo	
Estructura del ADN Por Medio del Juego del Caos 141 p 2022	

ESTRUCTURA DEL ADN POR MEDIO DEL JUEGO DEL CAOS

Beautiful, damn hard, increasingly useful.
That's fractals.
Benoit Mandelbrot

Agradecimientos

Este trabajo es fruto de muchos meses de ardua labor, sin embargo, esta tesis pudo ser concluida gracias al apoyo de varias personas.

Agradezco a la Dra. Laura Clementina Eslava Fernández quien me guio y aconsejó pacientemente en todo momento durante la construcción de la presente investigación, además, por su compresión ante los percances sucedidos.

A la Facultad de Ciencias, pues en ella encontré a grandes profesores que me inspiraron e hicieron sentir pasión por la ciencia y me alentaron siempre a seguir adelante.

A mis padres, Gabriela Silva y Guillermo Quintana, por todo el apoyo y cariño que me han dado desde que era niña; su compromiso y dedicación ha sido una inspiración para mí. A mi hermano Guillermo A. Quintana por sus consejos y guía tanto en este trabajo como en la vida. A Kora, por ser la más leal compañera que siempre me dio ánimos y risas en de momentos frustración; quien pasó largas horas acostada a mi lado, durante el proceso de escritura de tesis.

A mis queridos amigos, Mario, Julia, Kevin, Gema y Jazmín, quienes han sido un pilar emocional durante gran parte de la carrera, además, compañeros de risas y llantos; siempre me alentaron a seguir adelante y dar lo mejor de mí.

Este logro no es solo mío, sino también suyo, gracias por estar a mi lado en este largo viaje.

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM, con número TA100820. Agradezco a la DGAPA-UNAM la beca recibida.

Resumen

En este trabajo se encuentran patrones genéticos o genómicos haciendo uso de un área reciente de las matemáticas: los Sistemas Dinámicos. En particular, se utilizarán los sistemas de funciones iteradas, que cuentan con un componente estocástico, para la construcción de fractales y, a partir, de ellos reconocer algún patrón o característica particular de la estructura de los genes o genomas.

Esta idea surge del artículo de Joel Jeffrey titulado "Chaos game representation of gene structure" donde se menciona una forma gráfica de representar secuencias de ADN, haciendo uso del famoso sistema de funciones iteradas denotado como el Juego del Caos. Sin embargo, en ese artículo se dejan muchas preguntas abiertas. El presente trabajo se inspira en ellas para formular y contestar la pregunta: *¿El patrón que se observa en el genoma varía con respecto al de los genes?*

Con las anotaciones que ofrece la base de datos de secuencias genéticas del National Institutes of Health (NIH) que es llamada "GenBank", resulta sencillo poder extraer las secuencias de ADN que son reconocidas como regiones codificantes; además, son las secuencias de ADN más actualizadas y completas. En total se tomaron 30 genomas provenientes de representantes de los 5 reinos en los que se dividen los seres vivos de acuerdo a la biología y se les aplicó el sistema de funciones iteradas.

Haciendo uso del lenguaje de programación Python se logró repetir el algoritmo que se menciona en el artículo de Jeffrey y se realizó una comparación cualitativa de las representaciones gráficas de los genes y genomas, las cuales revelan información sobre la estructura de ambas secuencias y además se hizo una comparación entre especies, lo cual, bajo un estudio más sistemático y especializado, podría revelar información sobre la taxonomía.

Índice general

Índice de figuras

Índice de cuadros

Introducción

Desde la década de los setenta se han desarrollado metodologías que permiten la realización de modificaciones genéticas. Estas son las llamadas técnicas recombinantes de ácido desoxirribonucleico (ADN), las cuales tienen por objetivo modificar el genoma para el beneficio del ser humano; consisten en cortar, unir y reintroducir ciertos fragmentos de ADN en forma de cromosomas. Las técnicas de recombinación genética son muy utilizadas en la actualidad, ya que la modificación genética es bastante útil e importante para el desarrollo de la medicina como la creación de fármacos, el estudio de enfermedades y tratamientos, trasplantes de órganos, etc.

La arquitectura genética se refiere al patrón de efectos genéticos que construyen y controlan un carácter observable de un organismo dado y sus propiedades variacionales; en la actualidad, aún no se comprenden las arquitecturas genéticas que hacen a un individuo más susceptible a determinadas enfermedades que a otros. Sin embargo, la predicción de la arquitectura genética representa un gran problema en el reconocimiento de patrones. Por lo cual se han implementado modelos de predicción que han tenido grandes avances en la identificación computacional de genes utilizando, por ejemplo, los Modelos Ocultos de Markov, los cuales son reconocidos ampliamente por tener una gran precisión.[25]

En esta investigación se utilizan herramientas de un área aparentemente distante, los sistemas dinámicos. Estos forman parte de una rama reciente de las matemáticas que surge en la última década del Siglo XIX, gracias al matemático francés Henri J. Poincaré. Esta teoría moderna tiene como objetivo entender las irregularidades o azares de la naturaleza utilizando sistemas continuos o discretos. El caos juega un papel muy importante en esta área, ya que varios sistemas naturales o mecánicos presentan trayectorias erráticas e impredecibles y se dice que tienen un comportamiento caótico.

La teoría del caos trata con sistemas dinámicos particulares, aquellos que cumplen la particularidad de ser sensibles a condiciones iniciales, es decir, dos soluciones muy similares tienen trayectorias muy diferentes. Sin embargo, lo que busca esta rama de la ciencia es el orden subyacente en sistemas aparentemente aleatorios. Tiene su origen en 1963, con el meteorólogo y matemático Edward Lorenz, quien trabajó en un sistema de ecuaciones que

predijera el clima. En cierta ocasión, Lorenz quiso volver a emplear un resultado que había obtenido e introdujo un número que había obtenido a mitad de la trayectoria solución, pero el resultado final fue totalmente diferente al que había obtenido por primera vez. El error se debió a que su computadora almacenaba seis decimales y él había utilizado solo tres decimales. De modo que, en los sistemas estudiados dentro de la teoría del caos, cualquier pequeña perturbación en las condiciones iniciales puede tener una gran influencia sobre el resultado final y esta característica es conocida como el famoso efecto mariposa *"el aleteo de las alas de una mariposa puede provocar un Tsunami al otro lado del mundo"*.

En 1989 H. Joel Jeffrey en su artículo *Chaos game representation of gene structure*, muestra una nueva forma de analizar patrones genéticos, la idea consiste fundamentalmente en, dada una secuencia de nucleótidos provenientes de un genoma, realizar una representación gráfica con un sistema de iteración de funciones. Sorprendentemente esta representación da origen a Figuras fractales, lo que indica, que se pueden interpretar los patrones genéticos a través de Sistemas Dinámicos.

El objetivo de este trabajo es encontrar patrones que sean característicos del genoma y hacer un análisis comparativo de su representación completa y de secuencias codificantes. Este análisis depende del tipo de célula ya que, por ejemplo, en los eucariontes dentro del gen se encuentran secciones no codificantes que pueden tener estructuras diferentes de las codificantes. De hecho, se pudo encontrar que en bacterias cuyo tipo de celular es procariota no hay una diferencia significativa entre la representación del genoma completo y las regiones codificantes. Por otro lado, en especies con células eucariotas se pudo observar una diferencia significativa entre el genoma y las regiones codificantes; esta se acentúa más cuando el porcentaje del genoma perteneciente a las regiones codificantes es menor.

El trabajo comienza exponiendo los conceptos básicos de biología molecular para la codificación genética; se parte dando una concisa descripción de la estructura química de las moléculas de ADN y ARN para posteriormente, hablar de fragmentos de ADN o ARN codificantes los cuales son llamados genes. La importancia de los genes radica en que contienen la información necesaria para la creación de distintas proteínas y determinan su función. Subsecuentemente se introducen las estructuras del genoma de las células procariontes y eucariontes; resaltando que la forma en la que se distribuyen las regiones codificantes en cada célula es distinta y se habla de las regiones no codificantes y su importancia, también se recalca algunas características vitales para su reconocimiento.

El objetivo de la bioinformática es analizar información biológica, por esta razón, no es de extrañarse querer trabajar con las regiones codificantes del ADN; y así se han desarrollado métodos estadísticos para la predicción genética. En el presente trabajo se mencionan algunos métodos de predicción genética para así dar a conocer la base de datos

de secuencias genéticas del *National Institutes of Health*(NIH) que es llamada '*GenBank*'.; aquí se pueden encontrar las secuencias genéticas más actualizadas y completas.

Para introducir la parte matemática del trabajo, se inicia hablando de los fractales. Un fractal es un objeto matemático que cumple con la particularidad de contar con una estructura que se repite a diferentes escalas. Este objeto abstracto fue considerado una monstruosidad geométrica, carente de utilidad, sin embargo, Benoit B. Mandelbrot en su libro *The Fractal Geometry of Nature*, muestra que los fractales pueden encontrarse en la naturaleza justo en los lugares menos esperados.

La ecuación logística es un sistema dinámico que describe el crecimiento de una población en un ecosistema con recursos limitados; lo interesante de este sistema es analizar el comportamiento a largo plazo dada una población inicial; el cual está determinado por un parámetro: la tasa de crecimiento de población. Los resultados a largo plazo varían, la población puede estabilizarse en una cantidad u oscilar, estas cantidades son llamadas puntos atractores. Al graficar los puntos atractores de acuerdo con la variación del parámetro se divisa un fractal, a estos curiosos atractores se le suele llamar atractores extraños.

En los sistemas dinámicos existen otros sistemas que poseen atractores que usualmente son fractales, estos son los nombrados Sistemas de Funciones de Iteradas los cuales son un conjunto de funciones contractivas. Para programar y así poder visualizar su atractor; generalmente se utilizan dos formas, el algoritmo determinista o el algoritmo aleatorio, los cuales son descritos, programados y probados para así poder hacer un análisis riguroso sobre su complejidad y su efectividad.

Un sistema de funciones singular es El Juego del Caos; este sencillo algoritmo fue propuesto por Michael Barnsley en 1980. Para ejecutar el algoritmo solo se necesita una secuencia de números aleatorios y un triángulo; el sistema cuenta con la singularidad de que la forma de su atractor es el triángulo de Sierpinski, un fractal reconocido en el mundo de las matemáticas. Este algoritmo se puede ejecutar con diversos polígonos, sin embargo, el caso de interés es para un cuadrado donde su atractor parece llenar uniformemente el interior de este, aunque se puede obtener distintos resultados si la secuencia que se utiliza no es completamente aleatoria.

Finalmente, se indaga y analiza meticulosamente los patrones observados en los fractales generados por medio del Juego del Caos eligiendo al cuadrado como el polígono a utilizar y usando secuencias con distintas distribuciones de probabilidad. Para así posteriormente con el método para representar los patrones del ADN a través del Juego del Caos propuesto por H. Joel Jeffrey poder ilustrar y justificar las formas fractales obtenidas al usar las secuencias de ADN originarios del genoma y de las regiones codificante. Esto se hace para

cada representante proveniente de los cinco reinos en los que se dividen los seres vivos de acuerdo con la biología.

Capítulo 1

Codificación Genética

Las moléculas de ADN y ácido ribonucleico (ARN) son probablemente las más importantes de la biología molecular, ya que trabajan juntas para una correcta transmisión de información genética, también se encargan de su almacenamiento y recombinación; dando como resultado mutaciones y con ello una diversidad genética importante para el proceso evolutivo de una especie.

El ácido desoxirribonucleico codifica la información que determina las características de un organismo y sus funciones, también se encarga de transmitir estos atributos a su descendencia y posee una estructura de doble hélice; por lo que actúa como portador de información genética expresable con ciertas particularidades. En las células eucariotas el ADN se encuentra en el núcleo de la célula comprimido en cromosomas, mientras que en las células procariotas tienen un solo cromosoma circular que se encuentra en el citoplasma. [34, Cap. 4.0]

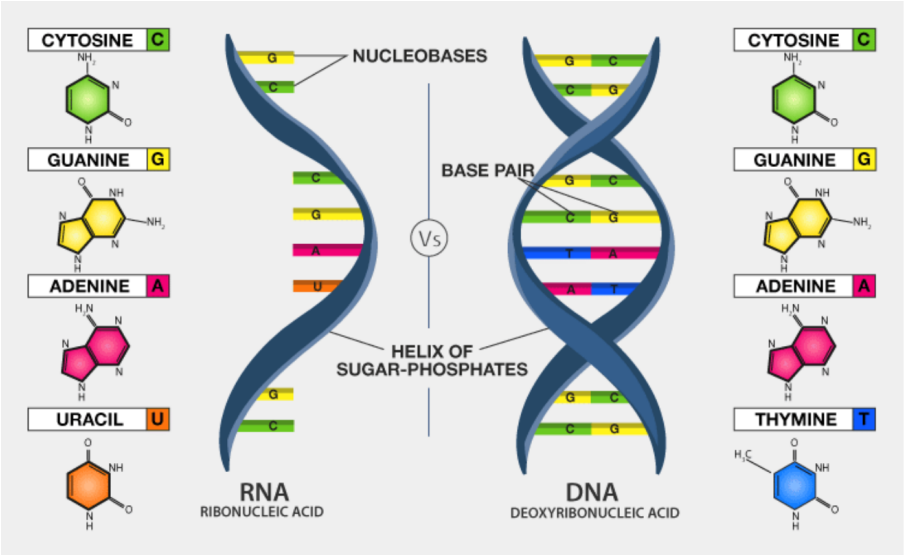


Figura 1.1: Comparación entre ARN y ADN.

1.1. Estructura del ADN

La principal función del ácido ribonucleico es la transmisión de información genética y es sintetizada a partir del ADN. Su estructura molecular es similar a la del ADN, pero la diferencia es que el ARN está constituido por una sola hebra o una hélice. En la Figura 1.1[1] se muestra la estructura de una hebra del ARN y de doble hebra que corresponde al ADN.[34, Cap. 4.0]

La estructura de la molécula del ADN es un polímero químico que está formado por dos cadenas antiparalelas que se enrollan entre sí, unidas por bases nitrogenadas con giro a la derecha, Esta estructura de doble hélice fue descubierta por Watson y Crick en 1953.

Las hélices están formadas por nucleótidos, compuestos por un azúcar, un grupo fosfato y una base nitrogenada. En el ADN las bases nitrogenadas son cuatro: La adenina (A) y la guanina (G) que son purinas, estructuras bicíclicas de dos anillos; la timina (T) y la citosina (C), que son pirimidinas, estructuras monocíclicas. En la Figura 1.2 [2], a la izquierda se encuentra la estructura base de las pirimidinas y purinas numeradas por sus carbonos, y en la derecha se encuentran las bases nitrogenadas utilizadas en el ADN y ARN. En el ARN la timina es remplazada por el uracilo (U), el cual también es una pirimidina. En el ADN, la molécula de azúcar se le denomina desoxirribosa, mientras que, en el ARN, la molécula de azúcar se le conoce como ribosa. La diferencia entre la desoxirribosa y la ribosa se encuentra en el carbono 2' con un enlace -H o -OH, respectivamente; sus estructuras se muestran en la Figura 1.3a [3]. [34, Cap. 4.1]

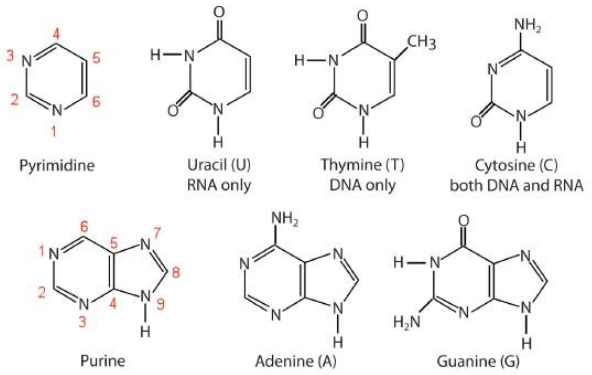
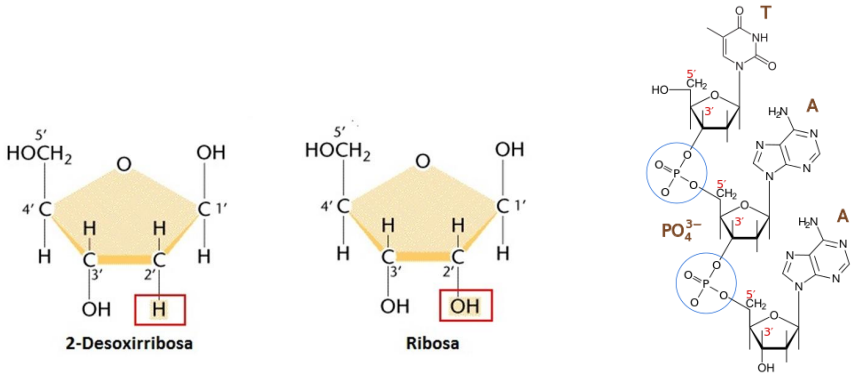


Figura 1.2: Estructura de las pirimidinas y purinas.

Las secuencias de ADN y ARN se suelen representar solamente con letras A, T, C, G y U. En química se suele numerar la cadena o ciclo carbonado, de manera que a los que tienen

[1] Figura tomada de [49].
[2] Figura tomada de [3].
[3] Figura tomada de [22].

sustituyente [4] les corresponda el número más bajo posible y son nombrados de acuerdo a los grupos funcionales que son responsables de la reactividad y propiedades químicas de los compuestos orgánicos [21]. En el ADN y ARN las moléculas de azúcar se unen entre sí a través de grupos fosfato que forman enlaces fosfodiéster en los átomos de carbono 3' y 5' como se ve en la Figura 1.3b [5]. La síntesis de proteínas procede de 5' a 3', por esta razón surge la convención de leer y escribir las secuencias de 5' a 3'.[34, Cap. 4.1]



(a) Desoxirribosa y Ribosa.

(b) Enlace fosfodiéster en la posición.

Figura 1.3: Estructuras químicas.

Como ya se mencionó, el ADN está formado por dos cadenas. Cada cadena o hebra dextrógira [6] está formada por azúcares unidos por un grupo fosfato y, unido a cada azúcar, se encuentra una de las cuatro bases nitrogenadas de ADN; estas hebras se enrollan alrededor de un eje formando una doble hélice. Estas son antiparelelas y están orientadas de 5' a 3' [7]; en la Figura 1.4 se ilustra el sentido de la cadena y su unión con los grupos fosfatos.[Cap. 4.1][34]

En el exterior de la doble hélice, dos cadenas de azúcares unidas a través de grupos fosfatos y en el interior se proyectan las bases nitrogenadas de las dos cadenas unidas a través de puentes de hidrógeno. Una purina de una de las cadenas se encuentra enfrentada y unida a una pirimidina en la otra cadena. De esta forma surge una complementariedad de bases que es consecuencia del tamaño, forma y composición química de las bases. La adenina (A) se une con la timina (T) con dos puentes de hidrógeno y la citosina (C) se une con la guanina (G) mediante tres puentes de hidrógeno en el ADN. En la Figura 1.4[8] se aprecia

[4] Grupo funcional o grupo alquilo, que ocupa el lugar de un átomo de un grupo saliente de un compuesto orgánico en general.

[5] Figura tomada de [57].

[6] Del lat. *dexter, -tra, -trum*: derecha, derecho. Que gira a la derecha.

[7] Posición del grupo fosfato en la ribosa.

[8] Figura tomada de [54].

esta estructura. En que la estructura del ARN se cambia la unión de adenina-timina por adenina-uracilo. [54]

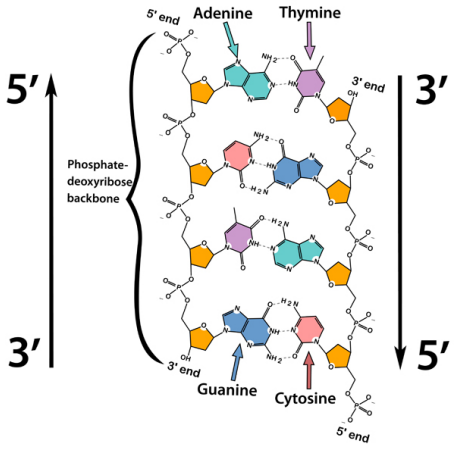


Figura 1.4: Representación en dos dimensiones del ADN.

1.1.1. Genes

Oculto en el ADN se encuentran los genes, que son fragmentos de ADN cuya secuencia nucleotídica codifica la estructura y función de cada proteína. Estos tienen la información para construir todas las células y tejidos del organismo, además, también son responsables de la regulación de la expresión génica. Las células que conforman un organismo tienen el mismo genoma, pero la diferencia entre los distintos tipos de células son los genes que se expresan. *'Por ejemplo, aunque una célula de la piel tiene toda la información genética al igual que la célula del hígado, en la piel solo se expresarán aquellos genes que den características de piel, mientras que los genes que dan características de hígado, estarán allí "apagados". Por el contrario, los genes que dan rasgos de "hígado" estarán activos en el hígado e inactivos en la piel.'*[6]

Esta información está organizada por unidades hereditarias y es lo que hoy se le conoce como genes. En términos más formales un gen se define como la secuencia completa de ácidos nucleicos necesarios para la síntesis de un producto génico funcional (polipéptido) o ARN funcional y a este fragmento del ADN se le denomina región codificante [34, Cap. 4.1].

1.2. Síntesis de Proteínas

Las proteínas son moléculas que cumplen diversas funciones estructurales, enzimáticas, hormonales, transportación de oxígeno, etc. Por lo que es de vital importancia conocer

qué secuencias de bases nitrogenadas codifican genes que dan lugar a distintas proteínas.

Las proteínas están compuestas a partir de aminoácidos. Hay 20 aminoácidos diferentes, y cada proteína tiene una secuencia de aminoácidos particular.[6]

1.2.1. Transcripción

La síntesis de proteínas inicia con la transcripción: los genes representados en el lenguaje de los nucleótidos se copian o transcriben a otra molécula. Para esto utiliza una hebra del ADN como molde para formar una cadena complementaria de ARN como se ve en 1.5[9], comúnmente se le llama *ARN mensajero* (ARNm). Este proceso es llevado a cabo por las enzimas *ARN polimerasas*. En el inicio de la transcripción, la ARN polimerasa con ayuda de factores proteínicos, denominados factores de transcripción generales, reconoce y se une a un sitio específico, denominado promotor para así iniciar la transcripción, al finalizar el ARNm, se libera de la ARN polimerasa [34, Cap. 4.2]

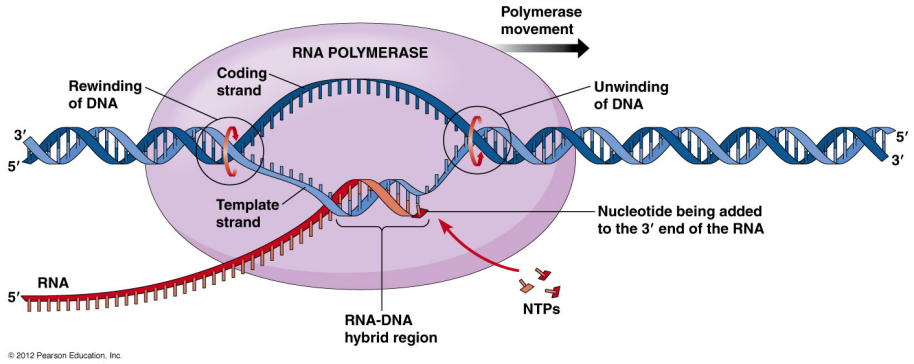


Figura 1.5: Transcripción.

1.2.2. Traducción

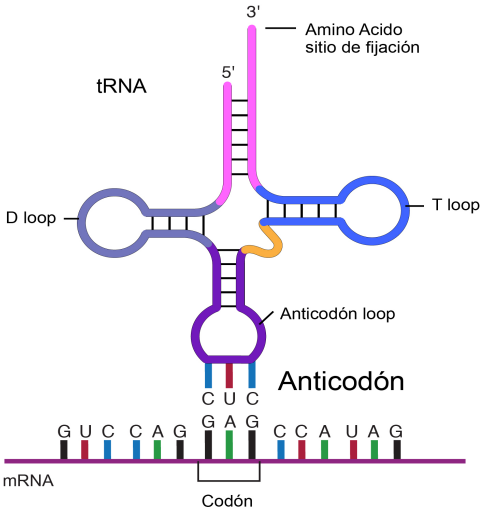
Luego de la transcripción, el ARNm se traduce al idioma de los aminoácidos para hacer proteínas. Consiste en utilizar el ARNm para ordenar y unir los aminoácidos en una cadena polipeptídica. La lectura correcta de los aminoácidos es muy importante para la producción de proteínas funcionales y lograr así el correcto funcionamiento de células y organismos. [34, Cap. 4.4]

La traducción se lleva a cabo en el ribosoma. El ARNm trasporta la información genética y se lee la secuencia de nucleótidos del ARN mensajero por tripletes o tríos de nucleótidos, denominados codones. A medida que el ribosoma lee la secuencia de codones se va formando una proteína a partir de la unión de aminoácidos. [34, Cap. 4.4]

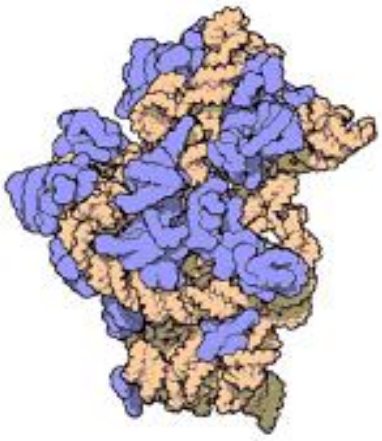
[9]Figura tomada de [9].

En este proceso participan principalmente tres moléculas: ARN mensajero (ARNm), ARN transferencia (ARNt) y ARN ribosómico (ARNr).

- ARNm: Transporta la información del ADN en forma de una serie de secuencias de tres nucleótidos, denominados codones, cada una especifica un aminoácido.
- ARNt: Cada tipo de aminoácido tiene su propio subgrupo de ARNt, que contiene una secuencia de tres nucleótidos en un extremo de esta molécula, que es llamada anticodón y es complementaria de las bases del codón. Transfiere el aminoácido adecuado a la creciente cadena polipeptídica en el procesos de síntesis. En la Figura 1.6a[10] se señala la estructura de esta molécula.
- ARNr: Se asocia con un conjunto de proteínas para formar ribosomas. Su principal función es catalizar el ensamblaje de aminoácidos para formar cadenas polipeptídicas; además, unen el ARNt con las proteínas. En la Figura 1.6b [11] se muestra su estructura. [34, Cap. 4.4]



(a) ARNt.



(b) ARNr.

Figura 1.6: Tipos de ARN.

Si se toman todas las combinaciones de cuatro bases tomadas de a tres, existe un total de 64 codones posibles. Cada codón determina qué aminoácido se colocará en la proteína que se está fabricando. De los 64 codones, 61 corresponden a aminoácidos y 3 son codones

[10]Figura tomada de [37].
[11]Figura tomada de [44].

de terminación (stop), responsables de la finalización de la síntesis proteica. Sólo existen 20 aminoácidos en la naturaleza, así que, varios codones pueden codificar para el mismo aminoácido. Al código se le denomina degenerado ya que más de un codón puede especificar el mismo aminoácido. En la Figura 1.7[12] se puede ver qué codones corresponden a cada aminoácido.[26].

Ala	GCU, GCC, GCA, GCG	**Leu**	UUA, UUG, CUU, CUC, CUA, CUG
Arg	CGU, CGC, CGA, CGG, AGA, AGG	**Lys**	AAA, AAG
Asn	AAU, AAC	**Met**	AUG
Asp	GAU, GAC	**Phe**	UUU, UUC
Cys	UGU, UGC	**Pro**	CCU, CCC, CCA, CCG
Gln	CAA, CAG	**Ser**	UCU, UCC, UCA, UCG, AGU, AGC
Glu	GAA, GAG	**Thr**	ACU, ACC, ACA, ACG
Gly	GGU, GGC, GGA, GGG	**Trp**	UGG
His	CAU, CAC	**Tyr**	UAU, UAC
Ile	AUU, AUC, AUA	**Val**	GUU, GUC, GUA, GUG
START	AUG	**STOP**	UAG, UGA, UAA

©BIOINNOVA
innovabiologia.com

Figura 1.7: Aminoácidos asociados a cada codón.

En la mayoría de los ARNm, el codón de inicio (AUG) específica la metionina aminoterminal. Sin embargo, en algunos ARNm bacterianos, GUG se usa como el codón iniciador, y en ocasiones CUG se usa como codón iniciador para la metionina en los eucariontes. En general, el significado de cada codón es el mismo en la mayoría de los organismos conocidos; pero se han encontrado organismos [13] en los que el significado de algunos codones difiere. [34, Cap. 4.3]

A la secuencia de codones que se encuentra desde un codón de inicio hasta un codón de terminación se le denomina marco de lectura abierto; a este fragmento también se le conoce como ORF por sus siglas en inglés *Open Reading Frame*. Un ARNm podría ser traducido en tres marcos de lectura diferentes, cada uno de ellos comienza en un nucleótido diferente de un mismo triplete. Sin embargo, la mayoría de los ARNm pueden ser leídos solamente en un marco de lectura, ya que los codones de terminación se encargan de detener la traducción antes de que se produzca la proteína funcional. Otra disposición de codificación no habitual se debe al corrimiento del marco de lectura *(frameshifting)*, donde cuatro nucleótidos son leídos como un aminoácido y se retrocede una base.[56]

[12] Figura tomada de [26].
[13] Un ejemplo son los protozoos ciliados y las acetabularia.

1.3. Diferencias de Genes Procariontes y Eucariontes

Las células del tipo procariota[14] se caracterizan por carecer de un núcleo definido y tienen una estructura interna sencilla. Las bacterias, cianobacterias o algas verdeazuladas son organismos unicelulares o cadenas filamentosas de células procariontes. Las células del tipo eucariota[15] contienen un núcleo definido rodeado por una membrana y otros compartimientos internos. Los organismos que poseen este tipo de célula son protozoarios, hongos, plantas y animales.

La codificación y la densidad de los genes cambia entre células eucariontes y procariotas, por ejemplo, se estima que en eucariontes el ADN codificante constituye aproximadamente el 2 % del genoma, mientras que en procariotas el 98 % del ADN es codificante. En eucariontes existen regiones no codificantes dentro de los genes llamados intrones, los cuales son importantes para tener diversas proteínas. [56]

La regulación génica corresponde a la forma en que una célula controla los genes que se expresan; también puede regular qué proteína produce y con qué rapidez. Así, se pueden tener células con diferentes funciones y que contienen la misma información genética. La forma de controlar la expresión de genes también difiere si la célula es procarionte o eucarionte. [34]

1.3.1. Procarionte

Los genes codificadores de proteínas en células procariontes que están dedicados a un único objetivo metabólico y se suelen encontrar en posiciones contiguas en el ADN. A este grupo funcional se denomina operón pues actúa como una unidad a partir de un único promotor y produce una hebra continua de ARNm, que luego es traducido a proteína. Los operones son inducibles o reprimibles, de acuerdo al mecanismo de control. En la Figura 1.8 [16] se ilustra esta estructura. Además, los genes están estrechamente empaquetados, con muy pocos intervalos no codificadores. [34, Cap. 4.3]

[14] Esta palabra se compone etimológicamente del prefijo *pro* que significa *antes de* y *karyo* que se refiere a *núcleo*. Se considera a la célula procariota anterior a la célula que tiene un núcleo celular.

[15] Esta palabra deriva del griego *eukayron*, compuesta por *eu* = *verdadero*, y *karyon*= *núcleo*. Así refiere a núcleo verdadero.

[16] Figura tomada de [31].

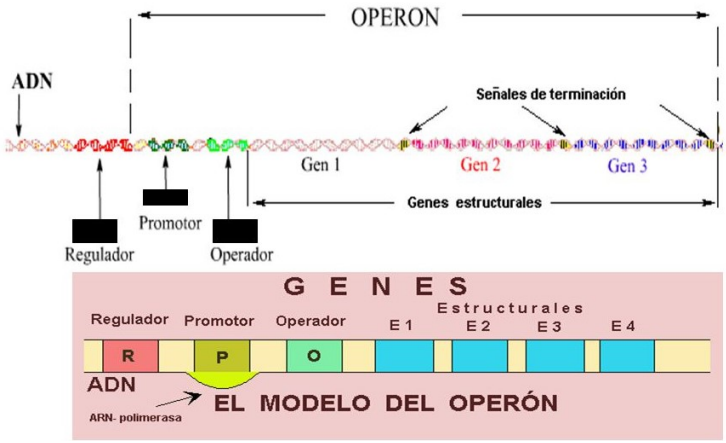


Figura 1.8: Estructura del operón.

Un operón se conforma de:

- Operador: controla el acceso de la ARN polimerasa al promotor.
- Promotor: donde la ARN polimerasa reconoce el sitio de inicio de la transcripción.
- Gen regulador: Controla el tiempo y velocidad de transcripción de otros genes.
- Gen estructural: codifica las enzimas relacionadas o las proteínas estructurales.

Otra característica de los procariontes es que pueden llevar a cabo la transcripción y la traducción al mismo tiempo.

La transcripción de un gen es reprimida cuando el ARNm correspondiente es sintetizado a baja velocidad y la trascripción de un gen es activada cuando el ARNm es codificado a mayor velocidad para así obtener proteínas más rápido. La expresión génica en bacterias y otros organismos unicelulares es mayormente regulada por enzimas y componentes estructurales que reaccionan a cambios en el medio nutricional y físico. De modo que, estos organismos solo sintetizan las proteínas que son requeridas para sobrevivir en condiciones particulares.[34, Cap. 4.3]

Un operón está regulado conjuntamente, es decir, todos los genes en el operón son activados o son reprimidos. La transcripción de operones es controlada por una interacción entre la ARN polimerasa y proteínas represoras y activadoras específicas. El proceso general, involucra un represor, el cual es una proteína específica que se une a la región operadora de un gen evitando el inicio de la transcripción. Algunos genes son regulados exclusivamente por activadores cuyos sitios de unión están localizados comúnmente $80-160$ pares de bases antes del sitio de inicio y son denominados amplificadores.[34, Cap. 4.3]

1.3.2. Eucariontes

Los genes en células eucariotas dedicados a una única vía están a menudo separados físicamente en el ADN, es decir, se pueden encontrar en diferentes cromosomas. Cada gen se transcribe a partir de su propio promotor, produciendo un ARNm. Además, el gen eucarionte existe en porciones de secuencias codificantes que se les conoce como exones, separados por segmentos no codificadores de proteínas, a los cuales se les denomina intrones.

Los intrones se transcriben junto con todo el gen, pero deben ser removidos del ARNm antes de que se complete el producto maduro de ARNm. El paso final es el corte y empalme (splicing) del ARN, la división interna del transcrito para eliminar los intrones, seguida por la ligación de los exones codificadores. En la Figura 1.9[17] se aprecia este proceso.[34]

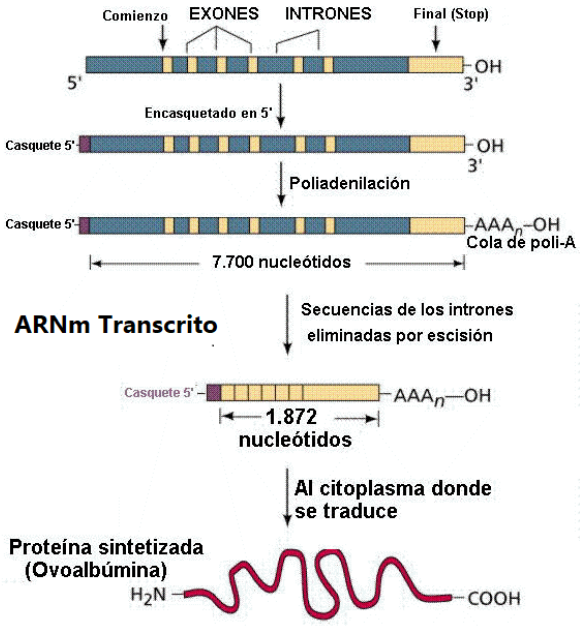


Figura 1.9: Proceso de transcripción de genes en células eucariontes.

La presencia de intrones en eucariontes permite la expresión de proteínas múltiples y emparentadas a partir de un gen único por el corte y empalme alternativo, también es un mecanismo para la producción de diferentes formas de una proteína que se les llama isomorfas. Se estima que el 60 % de los genes humanos se expresan como producto del corte y empalme alternativo.

La expresión génica en organismo multicelulares tiene como objetivo que sea expresado el gen correcto en la célula correcta en el momento adecuado y así contar con diferentes tejidos.

[17] Figura tomada de [31].

1.3.3. ADN no Codificador

El ADN no codificador que se encuentra dentro de los genes es el ADN no codificador intragenético, también denominado intrón. Estos deben ser removidos del ARN antes de que se complete el producto maduro de ARN. Aún no se sabe con exactitud cuál es el papel de los intrones en el ADN, pero se sabe que algunos regulan la expresión de los genes adyacentes a ellas; sin embargo de otros no se tiene ninguna función conocida. [13]

El ADN no codificador en los organismos multicelulares contiene muchas regiones que son similares, pero no idénticas; además, no se hallan en posiciones constantes dentro del ADN de individuos de las mismas especies. También estas regiones varían entre individuos originando así una huella digital de ADN.

En eucariontes superiores los genes se encuentran en medio de grandes extensiones de ADN no codificador; además, dentro de los genes se encuentran los intrones que es ADN aparentemente no funcional que suele ser más grande que el ADN codificante, los exones. Por ejemplo, en seres humanos alrededor del 30 % de su genoma está formado por intrones. Sólo alrededor del 3 % consiste de ADN codificador y el resto del genoma consiste de ADN no codificador, secuencias repetitivas y regiones regulatorias.[13]

Los intrones son poco frecuentes en eucariontes unicelulares y muy raros en bacterias. Se sabe que algunas porciones de intrones regulan la expresión de los genes adyacentes a ellos; sin embargo, entre intrones existe variación en las secuencias, incluso puede ser un cambio total, por esta razón se sugiere que tienen poca significancia funcional.[34, Cap. 10.1]

1.4. Predicción de Genes

Es indispensable usar enfoques computacionales para la predicción de genes. Este proceso involucra la predicción de la secuencias de nucleótidos que hay entre un codón de inicio y otro de terminación en la secuencia de ARN de traducción. A esta secuencia se le denomina marcos de lecturas abiertos (ORF). Además, en caso de células eucariontes se tiene que hacer la predicción de los delimitadores correspondientes a intrones. [56]

La predicción de genes representa un gran problema en el reconocimiento de patrones, ya que las regiones codificantes no tienden a conservar 'motivos' [18], por lo que es difícil reconocer un patrón con características pequeñas.[56]

[18] Un motivo en bioinformática es una secuencia de nucleótidos que es muy extendida con un significado biológico.

1.4.1. Programas de Predicción Genética

Estos métodos pueden ser clasificados en dos categorías, los enfoques *initio based* y *homology based*.

Initio–based se fundamenta en la predicción de genes dada una sola secuencia basándose principalmente en dos características. La primera es el reconocimiento de señales como los codones de inicio y parada, señales de corte y empalme de intrones, sitios de unión a factores de transcripción, sitios de unión a ribosomas y sitios de poliadenilación (poli-A). Además, la estructura del triplete del codón limita la longitud del marco de codificación a múltiplos de tres, lo que puede usarse como condición para la predicción de genes. La segunda característica es el contenido genético a partir de una descripción estadística de las regiones de codificación. Se puede hacer esta descripción o caracterización de regiones codificantes empleando modelos probabilísticos como las Cadenas de Markov o los Modelos Ocultos de Markov.[56]

Homology-based se fundamenta en hacer predicciones usando coincidencias de la secuencia que se analiza con secuencias de genes conocidas. Si mediante una búsqueda en una base de datos, se obtiene que la secuencia a predecir es similar a una proteína conocida o una familia de proteínas puede ser una fuerte evidencia de que la región codifica una proteína. [56]

Algunos algoritmos usan las dos estrategias anteriores para obtener mejores resultados. De igual manera, hay programas que combinan los resultados de diferentes algoritmos para obtener un consenso. Por último, también se pueden aplicar los métodos estadísticos que se utilizan en procariontes para la identificación de genes o técnicas de Machine Learning como una Red Neuronal (Neuronal Network - NN) o Análisis de Discriminante Lineal (Linear discriminant analysis - LDA). [56]

1.4.2. Predicción Genética en Procariontes

Los procariontes suelen tener un genoma pequeño con un rango de 0.5 Mbp a $10Mbp$, tomando en cuenta que $1\,Mbp = 10^6\ mp$ (par de bases). La densidad del genoma es alta. Alrededor del 90 % de este contiene secuencias codificantes. Como se mencionó antes, los genes en eucariontes están en bloques u operones. Los sitios codificantes suelen iniciar con el codón ATG y ocasionalmente GTG o TTG. La presencia de estos codones no necesariamente indican un sitio de inicio de la traducción. En cambio, para identificar el codón de iniciación de la trascripción, se utilizan otras características como el sitio de unión ribosomal también llamado 'Shine-Delgarno' que es un tramo de secuencia rica en purina. Se encuentra después del sitio de inicio de la transcripción y ligeramente antes del codón

de inicio de la traducción. En muchas bacterias, tiene un motivo de AGGAGGT. El codón de parada se caracteriza por una señal de terminación de la transcripción llamada terminador independiente. El *ρ-independient terminator* se distingue por tener una estructura secundaria de tallo-bucle seguida de una cadena de T's. Por otro lado, si se encuentra un codón de terminación en aproximadamente veinte codones, los más probable es que se esté en una región no codificante. Un marco de más de treinta codones sin interrupción sugiere una región de codificación de genes, aunque generalmente son más largos de cincuenta o sesenta codones. Los genes típicos están en el rango de 100 a 500 aminoácidos y los genes atípicos son más cortos o más largos; estos genes tienden a escapar a la detección de genes utilizando el modelo genético típico. [56]

El ADN procarionte se puede traducir en los seis cuadros posibles: tres cuadros hacia adelante y tres cuadros hacia atrás. En la práctica se suelen calcular los patrones estadísticos para todos los marcos posibles. Existe el método TESTCODE que usa el hecho de que los nucleótidos del tercer codón en una región codificante tienden a repetirse. Al trazar los patrones repetidos de los nucleótidos en esta posición se pueden diferenciar las regiones codificantes y no codificantes. [56]

1.4.3. Predicción Genética en Eucariontes

El genoma en los eucariontes es muy largo, con un tamaño que va desde los 10 Mbp a los 670 Gbp (1 $Gbp = 10^9$ bp), sin embargo la densidad de genes es muy baja. Por ejemplo, en los humanos solo el 3 % del genoma es codificante. En promedio hay un gen por cada 100 kbp. La transcripción de un gen en eucariontes se modifica de tres formas. La primera consiste en el corte de los extremos, metilación en el residuo inicial del ARN. El segundo es el empalme, el cual consiste en eliminar los intrones y unir los exones, aún no se sabe con exactitud cómo ocurre este proceso. También hay un fenómeno llamado empalme alternativo, este consiste en que algunos genes pueden tener sus transcripciones empalmadas y unidas de diferentes formas para generar más de una transcripción por gen. La tercera modificación es la poliadenilación, que es la adición de un tramo de A's (más o menos 250) en el extremo 3' del ARN. [56]

Hay características de los genes en eucariontes que hacen menos difícil su predicción. Por ejemplo, las uniones de empalme de intrones y exones siguen la regla GT-AG en la que un intrón en la unión de la posición 5' de empalme tiene un motivo de GTAAGT; y, del lado 3' la unión de empalme es un motivo de consenso de $(Py)_{12}$NCAG. Además, la mayoría de estos genes tienen una alta densidad de dinucleótidos CG cerca del sitio de inicio de la transcripción, a esta región se le conoce como isla CpG. [56]

1.4.4. Bases de Datos

Para contestar la pregunta *¿El patrón que se observa en el genoma varía con respecto al de los genes?* Primero se necesita extraer los genes, sin embargo, como ya se expuso en la sección anterior el reconocimiento de genes no es una tarea sencilla y es un problema que no se va tratar en este estudio, pero sí es necesario conocerlos. Para solucionar este problema se hará uso de la base de datos de secuencias genéticas de la *National Institutes of Health* (NIH) que es llamada '*GenBank*'. Esta es una colección anotada de todas las secuencias de ADN disponibles públicamente y que está conformada por el Banco de Datos de ADN de Japón (DDBJ), el Archivo Europeo de Nucleótidos (ENA) y el GenBank del NCBI.

GenBank está diseñada para proporcionar el acceso a la información de secuencias de ADN más actualizada y completa, para saber más sobre esta base y su uso, visitar la página oficial a través del siguiente url https://www.ncbi.nlm.nih.gov/genbank/.

Para encontrar los genes se utilizaron identificadores de secuencia y anotaciones con *Entrez Nucleotide* y la librería de Python *Bio*. BioPython es un conjunto de herramientas para la computación biológica. En particular se utilizará Bio.SeqIO, el cual es un módulo que se utiliza para trabajar con archivos de GenBank. Para analizar archivos GenBank, hay que iterar sobre un identificador que contiene varios registros de GenBank, estos son objetos de registro específicos.

Cada secuencia de ADN contiene información como: el nombre, id, descripción, referencias a otras bases de datos y un atributo muy útil para el presente trabajo, que es features.

Features contiene información de regiones codificantes para diferentes etapas de la síntesis de proteínas.

- **CDS** - Secuencias codificantes (CDS, *coding sequence*). Estas incluyen los codones de comienzo y fin.
- **mRNA** - Es la versión del ARN del gen, el cual es utilizado para fabricar las proteínas ARNm.
- **gene** - Región de interés biológico identificada como gen.

En particular se van tomar los CDS, porque el objetivo del trabajo es describir la estructura de genes. Para más información sobre la estructura de los datos que ofrece GenBank consultar repositorios universitarios, UNAM[17].

Los genomas y las regiones codificantes actuales que se encuentran en la base de datos de las diferentes especies se están actualizando constantemente, el último genoma que

se obtuvo fue el día 19/02/2021. Además, hay que tener presente la existencia de genes o regiones codificantes que aún no están identificados y esto afectaría levemente a las representaciones gráficas de los genomas como de genes.

Por otro lado, sería interesante observar el patrón que forman las proteínas funcionales asociadas a un gen, considerando que esto solo se podría hacer con células eucariontes, pero esto no es tema de la presente investigación.

Capítulo 2

Fractales y Sistemas Dinámicos Discretos

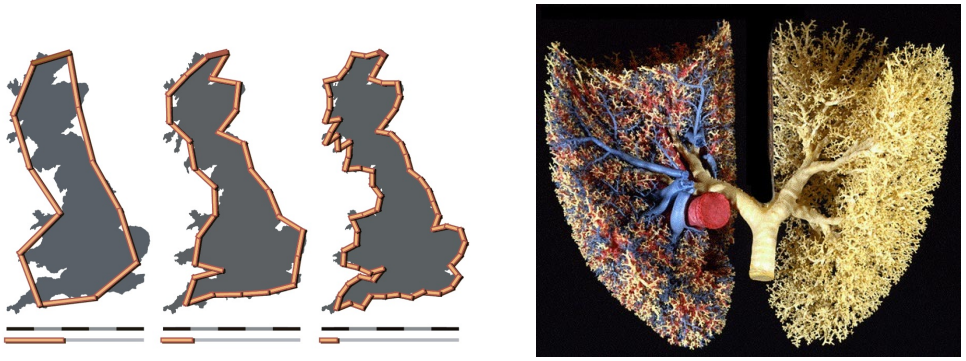
Los sistemas dinámicos tienen como objetivo estudiar la dinámica de una función a través del tiempo. Usualmente, para hacer este análisis se utiliza la composición de una función consigo misma, a este proceso se le llama iterar sobre una función. Suelen ser de interés los puntos fijos, que son aquellos tales que $f(x) = x$. Los sistemas aleatorios de funciones iteradas son un conjunto de funciones que se iteran de tal forma que, en cada repetición se escoge aleatoriamente la función a aplicar del conjunto de funciones. En esta sección se expone cómo se puede definir una noción de 'punto fijo' para los sistemas de funciones iteradas. Estos se pueden encontrar por un método determinista o aleatorio y algunos tienen una estructura fractal muy interesante.

El término fractal, que viene del latín *'fractus'* y significa fragmentado o roto, fue dado por el matemático Benoit B. Mandelbrot. Un fractal es un objeto matemático cuya estructura se repite a diferente 'escalas', el cual tiene generalmente la propiedad de auto similitud.

Las características de los fractales hicieron que se les considerara como *monstruosidades* geométricas sin ninguna aplicación. Mandelbrot en 1967 publicó el artículo *How Long Is the Coast of Britain?* [36] donde la problemática central es dar la longitud exacta de la costa de Gran Bretaña; en él explica que la medida depende de cuánto detalle se tenga al medir. La costa no es estrictamente similar a sí misma: varía al azar. En la Figura 2.1a[1] se observan tres aproximaciones a la longitud de la costa con diferentes escalas, este fue el primer acercamiento a curvas autosimilares. Posteriormente en 1977 Mandelbrot en su libro *'The Fractal Geometry of Nature'*[35] muestra su aplicación en fenómenos como la distribución de las estrellas del Universo, la ramificación alveolar en los pulmones (ver

[1]Figura tomada de [48].

Figura 2.1b [2]), la frontera difusa de una nube, las fluctuaciones de precios en un mercado y en la frecuencia de repetición de las palabras de un texto.



(a) Midiendo la costa de Gran Bretaña.

(b) Pulmones humanos (con sistema circulatorio a la izquierda).

Figura 2.1: Ejemplos de fractales en la naturaleza.

Hoy en día se les utiliza, entre muchas cosas, para almacenar o transmitir señales visuales y para simular paisajes a través de funciones iteradas. Se utilizan también en la mecánica estadística, en particular cuando se trata de sistemas físicos que tienen características que parecen aleatorias. Por ejemplo, Benoit B. Mandelbrot en su libro utiliza fractales para analizar la turbulencia de fluidos, singularidades en ecuaciones diferenciales, el movimiento browniano, entre otras aplicaciones [35]. La geometría fractal también ha contribuido a los gráficos por computadora (ver Sección 2.5).

2.1. Ecuación Logística

Un sistema dinámico tiene como objetivo analizar un fenómeno natural a través del tiempo. Uno de los sistemas más sencillos a estudiar, es la dinámica de poblaciones, que tiene como objetivo relacionar la reproducción de la población actual con la generación precedente involucrando distintos factores. En 1976, *Robert May* formula la ecuación $x_{n+1} = rx_n(1 - x_n)$ para el estudio de crecimiento de poblaciones. Este es un sistema no lineal que toma en cuenta los efectos de saturación del ecosistema, donde $r \geq 0$ es la tasa de crecimiento, $n = 0, 1, \ldots$ corresponde a la generación, y $x_n \geq 0$ representa la relación entre la población existente y la población máxima posible. Para saber más de planteamiento de este sistema dinámico ver [32, pág. 4].

Para entender su comportamiento a largo plazo, se verá el mapeo logístico en términos más generales; como un ejemplo de un mapeo a tiempo discreto unidimensional, tal y

[2] Figura tomada de [12]

como se muestra en la siguiente ecuación.

$$x_{n+1} = f(x_n, r), \quad \textit{donde} \quad f(x, r) = rx(1 - x) \tag{2.1}$$

La dinámica de f solo tiene sentido en $[0, 1]$ que es donde existe una población; es decir, x_n representa una densidad no negativa. Además, se puede observar que, fijando r, el máximo de f se encuentra en $x = \frac{1}{2}$ donde alcanza el valor de $\frac{r}{4}$. Más aún, se puede demostrar que para cada r, la función $f(x, r)$ representa una parábola hacia abajo que pasa por los puntos $(0, 0)$ y $(1, 0)$ y, por lo tanto, su máximo lo alcanza en $\frac{1}{2}$. En la Figura 2.2 se presenta la función para diferentes valores de r.

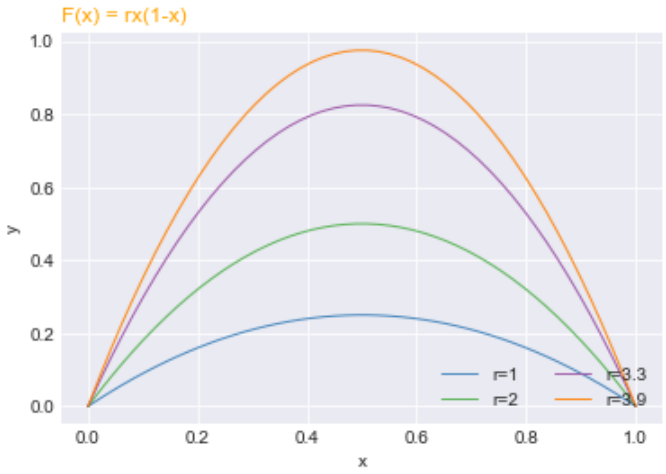


Figura 2.2: Gráfica de la ecuación logística para los valores de $r = 1,\ 2,\ 3.3,\ 3.9$.

El objetivo es analizar el comportamiento a largo plazo, ya que de esta manera se puede predecir el crecimiento de la población. Así, se introducirán los siguientes conceptos para formalizar la idea de iterar una función y además este concepto será de utilidad en capítulos posteriores. La mayoría de las definiciones fueron tomadas del libro *Sistemas dinámicos discretos*, escrito por *Jefferson King y Héctor Méndez* [32].

Un sistema dinámico discreto se compone de dos elementos: X un espacio métrico y $f : X \to X$ una función continua. En esta sección se va a tomar a X como el conjunto de los números reales a menos que se diga lo contrario.

Definición 2.1.1. La *órbita* de x bajo f se define como:

$$o(x, f) = \left\{x, f(x), f^2(x), f^3(x) \dots\right\}.$$

Definición 2.1.2. Se dice que $x_0 \in X$ es *punto fijo* de f si $f(x_0) = x_0$; equivalentemente si $o(x_0, f) = \{x_0\}$.

Definición 2.1.3. Sea $f : X \to X$ y $x_0 \in X$. Se dice que x_0 es un *punto periódico de* f si existe $n \in \mathbb{N}$ tal que $f^n(x_0) = x_0$. Al conjunto de todos los puntos periódicos de f se denota por $Per(f)$. Si $x \in Per(f)$, se dice que $o(x, f)$ es una *órbita periódica.*

Sea $x_0 \in Per(f)$. Se dice que x_0 tiene período k si

$$k = min\left\{n \in \mathbb{N} : f^n(x_0) = x_0\right\}.$$

Definición 2.1.4. Sea x_0 un punto fijo de f. Se dice que x_0 es un *punto fijo atractor* si existe un intervalo I abierto que contenga a x_0 que satisface la condición:

si $x \in I$ entonces $f^n(x) \in I$ para toda n, y además, $f^n(x) \to x_0$ cuando $n \to \infty$.

Definición 2.1.5. Sea x_0 un punto fijo de f. Se dice que x_0 es un *punto fijo repulsor* si existe un intervalo I abierto que contenga a x_0 que satisface la condición:

si $x \in I$ y $x \neq x_0$ entonces hay un entero $n > 0$ tal que $f^n(x) \notin I$.

Definición 2.1.6. Sea $f : X \to X$ una función continua en X. Se dice que un punto x_0 de X tiene *órbita (Lyapunov) estable*, si para toda $\epsilon > 0$, existe $\delta > 0$, tal que para toda $x \in B(x_0, \delta)$ y para toda $n \geq 0$ se tiene que

$$d(f^n(x), f^n(x_0)) < \epsilon.$$

Definición 2.1.7. La órbita de $x_0 \in X$ *no es estable* si sucede lo siguiente: existe $\epsilon_0 > 0$ tal que para todo $\delta > 0$, se puede encontrar un punto $y \in B(x_0, \delta)$ y un número natural n, que depende de y, tales que

$$d(f^n(x_0), f^n(y)) \geq \epsilon_0.$$

El siguiente teorema (tomado de [32, Pág. 128]) proporciona condiciones suficientes para caracterizar el tipo de punto fijo.

Teorema 2.1.1. Sea $f : A \to A$ un función continua en un intervalo $A \subset \mathbb{R}$. Sea $x_0 \in A$. Supóngase que f es derivable en x_0, y que $f(x_O) = x_0$.

1. Si $|f'(x_0)| < 1$ entonces x_0 es un punto fijo estable.
2. Si $|f'(x_0)| > 1$ entonces x_0 es un punto fijo que no estable.

Definición 2.1.8. Una función f es contractiva o una contracción si existe una constante $k < 1$, tal que para $x, y \in X$ se cumple

$$d(f(x), f(y)) < kd(x, y).$$

Una forma para poder visualizar la órbita de un punto es usando el *Diagrama de telaraña*. Para construirlo primero se tiene que graficar la función $y = f(x)$ del mapeo discreto y la función identidad $y = x$.

Algoritmo 2.1.1 (Diagrama de telaraña.)**.** Sea $f : X \to X$ una función y x_0 en el dominio de la función [32, pág. 17].

1. Se comienza en el punto $(x_0, 0)$.
2. Ubicar en la curva de la función f la imagen de x_0, cuyas coordenadas son $(x_0, f(x_0))$ y trazar una línea vertical que una el punto $(x_0, 0)$ con el punto $(x_0, f(x_0))$.
3. Se traza una línea horizontal desde un punto anterior hasta la identidad, es decir, un segmento que una el punto $(x_0, f(x_0))$ con el punto $(f(x_0), f(x_0))$.
4. Unir verticalmente desde el punto de la diagonal hasta la curva de función f. Este segmento une los puntos $(f(x_0), f(x_0))$ y $(f(x_0), f(f(x_0)))$.
5. Volver al paso 3, hasta que sea necesario.

Para más detalles sobre la implementación del algoritmo 2.1.1 en el apartado D.1 se adjunta el código escrito en el lenguaje Python3 . En la Figura 2.3a, se hace el diagrama de telaraña con el mapeo logístico que corresponde a la ecuación (2.1) para $r = 1/3$. Se observa que la órbita de $x_0 = 0.8$ converge a $x = 0$ que es punto fijo estable (ver Ejemplo 2.1.1), es decir, la tasa de crecimiento no es lo suficientemente grande para evitar que la población se extinga. De hecho se puede decir más, si una función es contracción entonces el punto fijo es atractor global, en otras palabras, para toda condición inicial su órbita converge al punto fijo; este resultado se obtiene directamente del Teorema de Punto Fijo de Banach [38].

Teorema 2.1.2 (Teorema del punto fijo de Banach)**.** Sea X un espacio métrico completo, no vacío, y sea $\phi : X \to X$ una contracción. Entonces se cumple lo siguiente:

- ϕ tiene un único punto fijo x^*.
- Para cualquier $x_0 \in X$, la sucesión $(\phi^k(x_0))$ converge a x^* en X, y se cumple que:

$$d(x^*, \phi^n(x)) \leq \frac{k^n}{1-k} d(\phi(x_0), x_0), \quad \forall n \geq 1. \tag{2.2}$$

Ejemplo 2.1.1. La función logística con $r = 1/3$ es una contracción y por lo tanto $x_0 = 0$ es su único punto fijo. Para demostrar que f es una contracción se calcula:

$$d(f(x), f(y)) = |\frac{1}{3}x(1-x) - \frac{1}{3}y(1-y)|$$

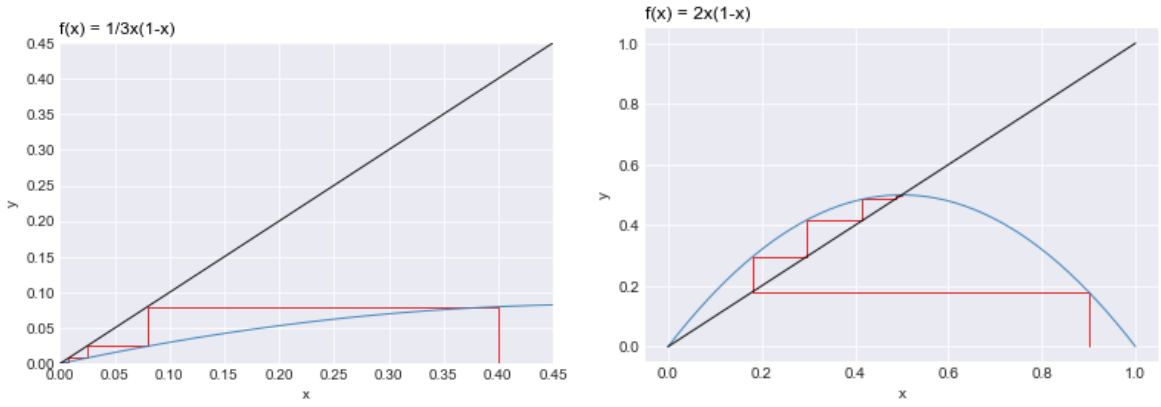
Usando el Teorema de Valor medio el cual dice: Sean $a, b \in \mathbb{R}$ con $a < b$ y $f : [a, b] \to \mathbb{R}$ una función continua en $[a, b]$ y derivable en $[a, b]$. Entonces existe $c \in [a, b]$ tal que $f(b) - f(a) = f'(c)(b - a)$. Se tiene que:

$$\frac{1}{3}x(1-x) - \frac{1}{3}y(1-y) = f'(c)(x-y)$$

Como $x \in [0, 1]$ entonces $f'(c) = \frac{1}{3}(1 - 2x) \in [\frac{-1}{3}, \frac{1}{3}]$ por lo que $f'(c) \in [\frac{-1}{3}, \frac{1}{3}]$. Tomando valor absoluto de ambos lados

$$\left|\frac{1}{3}x(1-x) - \frac{1}{3}y(1-y)\right| = |f'(c)|\,|(x-y)| \leq |x-y|$$

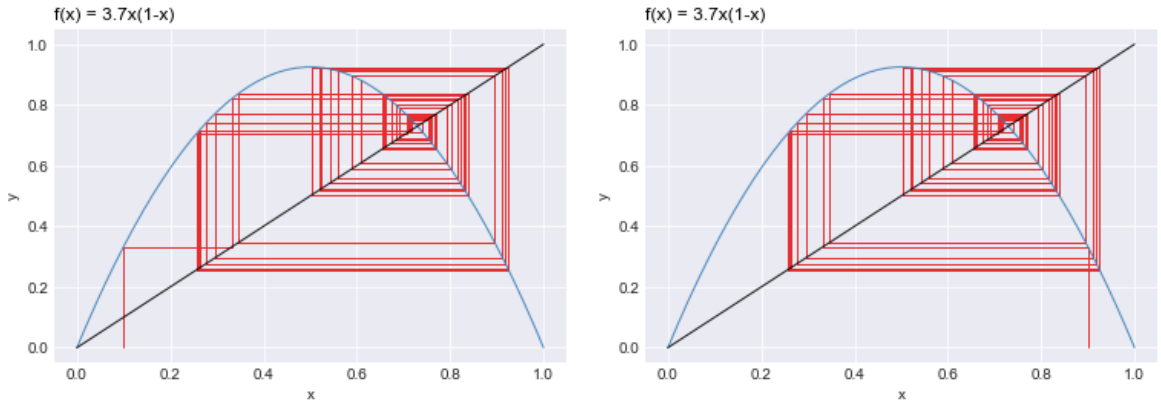
$\therefore f$ es contracción.



(a) $f(x) = \frac{1}{3}x(1-x)$, $x_0 = 0.8$ (b) $f(x) = 2x(1-x)$, $x_0 = 0.9$

Figura 2.3: Diagrama de Telaraña para el mapeo logístico con $r = \frac{1}{3}, 2$.

En la Figura 2.3 se muestran a los diagramas de telaraña para la ecuación logística con $r = \frac{1}{3}$ y $r = 2$ en 2.3b. Para el segundo caso, su órbita converge a un punto fijo que es distinto de cero, es decir, la población no se extingue; se estabiliza. Por último, se hace el diagrama para $r = 3.7$ en la Figura 2.4 para diferentes puntos de inicio $x_0 = 0.1$ y $x_0 = 0.9$, donde no se puede apreciar una convergencia, más bien parece tener un ciclo.



(a) $f(x) = 3.7x(1-x)$, $x_0 = 0.1$ (b) $f(x) = 3.7x(1-x)$, , $x_0 = 0.9$

Figura 2.4: Diagrama de Telaraña para el mapeo logístico con $r = 3.7$.

La característica a destacar de los puntos fijos, es que su estabilidad da información de cómo se comporta el sistema a largo plazo. Si se tiene un punto atractor, todos los puntos en una vecindad tienden al punto fijo atractor cuando se itera la función, por otro lado, si el punto fijo es repulsor, entonces todos los puntos cercanos al ser iterados se alejan de este.

La figura 2.5 ilustra este fenómeno para diferentes valores iniciales x_0, con valores fijos de r. Si se itera x_n para una tasa menor que 1 la población se extingue; hay que resaltar que solo hay un punto fijo (el origen) y es atractor, por lo tanto, la población tiende a extinguirse, ya que $x_n \to 0$ cuando $n \to \infty$.

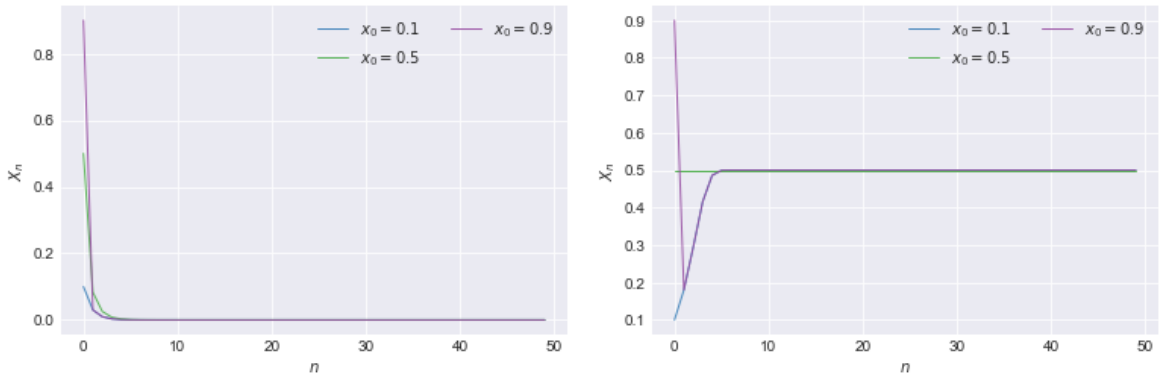
Por otro lado, para valores de $1 < r < 3$ habrá dos puntos fijos $x^* = 0$ y $x^* = 1 - \frac{1}{r}$, y puede mostrarse que la población tiende a estabilizarse en el valor distinto de cero. Para verificar que esto es cierto, se puede hacer uso del Teorema 2.1.1.

Ejemplo 2.1.2. Los puntos fijos para la ecuación $f(x) = rx(1-x)$ son $x^* = 0$ y $x^* = 1-\frac{1}{r}$. Obsérvese que $x = rx(1-x)$ entonces $(1-r)x + rx^2 = 0$; las soluciones están dadas por $x_1^* = 0$ y $x_2^* = 1 - \frac{1}{r}$, esta última solución solo tiene sentido para el modelo si $x_2^* \in (0, 1)$; es decir, $r > 1$.

Como se menciona en el Teorema 2.1.1 se puede saber la estabilidad de los puntos fijos evaluándolos en la derivada. Se sigue que $f'(x) = r(1-2x)$ y evaluando los puntos se obtiene $f'(x_1^*) = f'(0) = r$ y $f'(x_2^*) = f'(1-\frac{1}{r}) = r(1-2(1-\frac{1}{r})) = 2-r$.

En el caso $r < 1$ se tiene un solo punto fijo que $|f'(x_1)| = |f'(0)| = r < 1$ y así $x_1^* = 0$ es el atractor del sistema. Si $1 < r < 3$ se producen dos punto fijos $x_1^* = 0$ es un punto fijo repulsor, mientras que el punto $x_2^* = 1 - \frac{1}{r}$ es atractor. Finalmente, si $r > 3$ ambos puntos fijos x_1^* y x_2^* son repulsores.

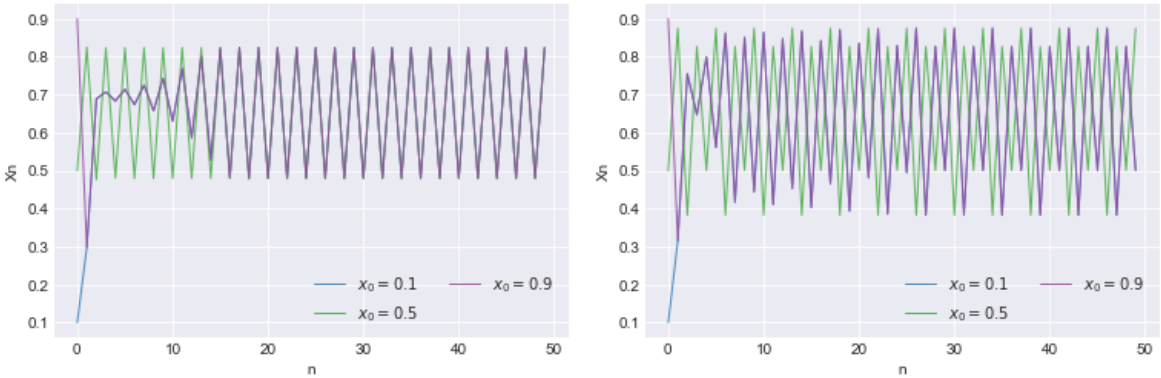
Esto se puede ver gráficamente en la Figura 2.5 se ve como x_n converge al punto atractor independientemente del punto x_0. En el eje x se muestra el número n que corresponde a la iteración y en el eje y se muestra su valor en la función.



(a) Mapeo logístico para $r = \frac{1}{3}$ (b) Mapeo logístico para $r = 2$

Figura 2.5: Gráfica de las órbitas para los puntos $x_0 = 0.1,\ 0.5,\ 0.9$. Representa la secuencia de valores al iterar un punto x_0 en la función a través del tiempo (número de iteraciones).

Para valores de $r > 3$ no se puede llegar a un estado donde la población se estabilice. Por ejemplo, para $r = 3.3$ la órbita de x_0 bajo f, que se obtiene para diferentes valores iniciales tienden a oscilar entre dos valores que alternan entre una población grande en una generación y una población más pequeña en la siguiente, es decir, surgen órbitas de período dos que son atractoras. Para $r = 3.5$ se duplica este período, obteniendo órbitas de período 4. La Figura 2.6a y 2.6b muestran estas oscilaciones.



(a) Mapeo logístico para $r = 3.3$ (b) Mapeo logístico para $r = 3.5$

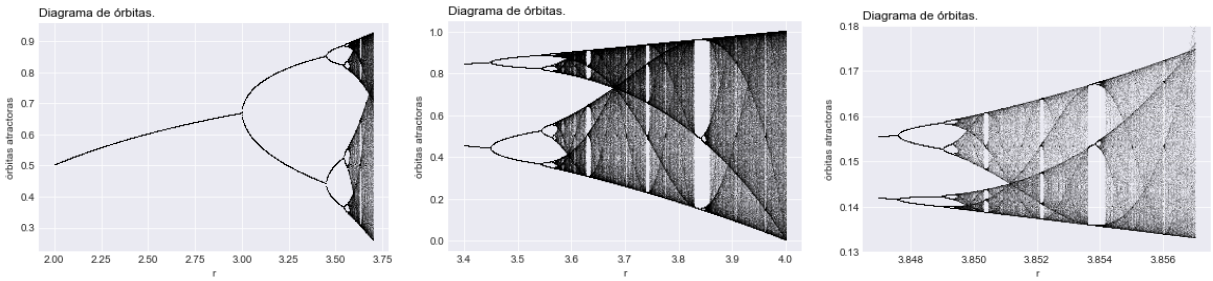
Figura 2.6: Gráficas de las órbitas para los puntos $x_0 = 0.1,\ 0.5,\ 0.9$. Representa la secuencia de valores al iterar un punto x_0 en la función a través del tiempo (número de iteraciones).

2.1.1. Atractor de la Función Logística

Como se vio en la sección anterior, variando el parámetro r en la ecuación logística se puede describir la estructura del flujo del sistema. Se logra observar la aparición o desaparición de puntos fijos u órbitas periódicas, así también, cómo cambia su estabilidad; a estos cambios cualitativos en la dinámica del sistema se les llama Bifurcaciones. [41] Además, a los valores del parámetro donde ocurren estos cambios se les llama puntos de bifurcación. El Diagrama de Bifurcaciones es una herramienta para visualizar las bifurcaciones de un sistema dinámico y con esta información poder especular el comportamiento del sistema.[32]

El Diagrama de Bifurcaciones consiste en graficar para cada valor fijo de r el ciclo atractor al que tiende el sistema. Por ejemplo, en la Figura 2.6a se ve que su ciclo es de período 2 y sus valores son $x_1 \approx 0.81\ldots$ y $x_2 \approx 0.49\ldots$ por lo que, en la Figura 2.7, en $r = 3.3$ se gráfica x_1 y x_2.

Se suele construir este diagrama utilizando un ordenador, pues para ciertos valores del parámetro r la órbita atractora tiene muchos valores, y sería muy difícil calcularlos de forma exacta. Se empieza tomando un punto arbitrario x_0 perteneciente al dominio de la función, en este caso $x_0 \in [0, 1]$, posteriormente se itera el punto x_0 una cantidad de veces N donde N es un número grande para asegurar la convergencia a la órbita atractora y finalmente se grafican los últimos valores que resultarán ser los valores de la órbita atractora, m donde $m < N$. Por ejemplo, si se itera x_0 1000 veces entonces se grafican los últimos 300 valores.[32]



(a) $2 \leq r \leq 4$ (b) $3.4 \leq r \leq 4$ (c) $3.847 \leq r \leq 3.857$

Figura 2.7: Diagrama de órbitas.

La Figura 2.7c es una aproximación a la región $(r, x) \in [3.847, 3.857] \times [0.13, 0.18]$. Inesperadamente se forma una copia del diagrama en miniatura y dentro de esta copia va a existir otra minicopia de ella misma y esto va a suceder una cantidad infinita de veces. Este fenómeno es llamado autosemejanza. Esta es una característica que posee un conjunto de ser igual a un número fijo de copias reescaladas de sí mismos y se observa

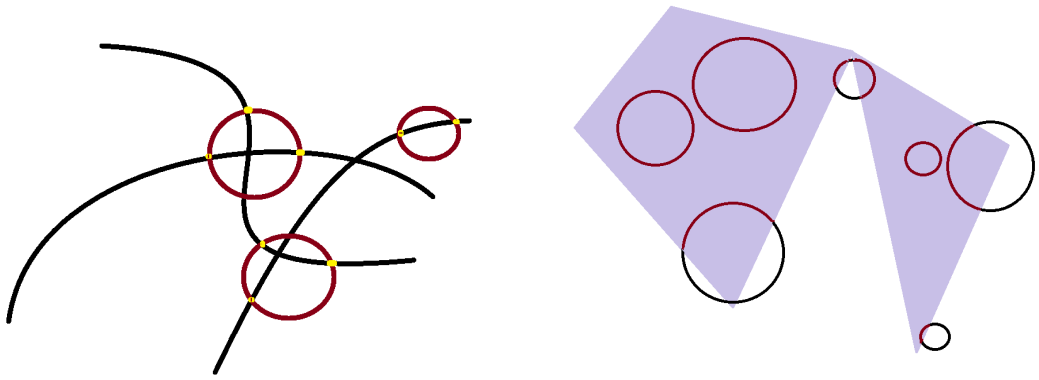
en los fractales que se verán en la Sección 2.2. El diagrama de órbita revela una mezcla inesperada de orden y caos, con ventanas periódicas intercaladas entre nubes caóticas de puntos, alcanzando su límite en $r_\infty = 3.5699\ldots$ con una órbita infinita [53, pág. 356].

2.2. Fractales

Antes de dar la definición formal de fractal, es importante partir de algunos conceptos previos.

Definición 2.2.1. Un conjunto $S \subset \mathbb{R}^n$ tiene dimensión topológica k, donde k es el entero no negativo más pequeño para el que se tiene: cada punto en S tiene vecindades arbitrariamente pequeñas cuyas fronteras interceptan a S en un conjunto de dimensión $k-1$. Si S es una colección numerable de puntos de $\mathbb{R}^n$ se dice que S tiene dimensión cero [16, pág. 186].

Para ilustrar este concepto en la Figura 2.8 se muestran dos ejemplos. El primero son curvas en el plano que corresponde a la Figura 2.8a, el conjunto que corresponde a la intersección de la curva y de la frontera para cualquier vecindad arbitrariamente pequeña; se puede observar forzosamente que coinciden en uno o dos puntos. Por lo tanto, un segmento tiene dimensión topológica igual a 1 [16, pág. 186].



(a) La frontera cualquier vecindad se intersecta con S en un conjunto de dimensión 0.

(b) La frontera de cualquier vecindad se intersecta con S en un conjunto de dimensión 1.

Figura 2.8: Ejemplos de dimensión topológica.

De manera análoga en el segundo ejemplo, para A una región en el plano, la intersección de la frontera con A es un conjunto de dimensión 1 como se muestra en la Figura 2.8b, por lo tanto A tiene dimensión 2. Las siguientes definiciones y ejemplos fueron tomados

del libro *A First Course in Chaotic Dynamical Systems* escrito por *Robert L. Devaney* [16].

Definición 2.2.2. Un conjunto S es llamado autosimilar afín, si S puede ser subdividido en k subconjuntos congruentes. Es decir, cada uno de los subconjuntos puede ser expandido por la magnitud de un factor M para reconstruir el conjunto S. [16, pág. 187]

Definición 2.2.3. Sea S un conjunto autosimilar afín que puede ser subdividido en k piezas congruentes cada una, las cuales pueden ser expandidas por la magnitud de un factor M para formar el conjunto S. Entonces la dimensión fractal D de S es:

$$D = \frac{log(k)}{log(M)} = \frac{log(nro.\ de\ piezas)}{log(magnitud\ del\ factor)} \tag{2.3}$$

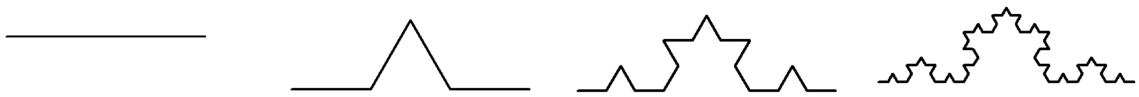
Definición 2.2.4. Un fractal es un subconjunto de $\mathbb{R}^n$ que es autosimilar afín y cuya dimensión fractal excede su dimensión topológica.[16, pág. 178]

A continuación se presentan algunos ejemplos de los fractales más famosos.

Ejemplo 2.2.1. La Curva de Koch

Para construir la Curva de Koch se parte del segmento unidad $[0, 1]$ el cual es k_0, posteriormente se elimina el intervalo abierto central de longitud $\frac{1}{3}$ y sustituyéndolo por dos intervalos de longitud $\frac{1}{3}$ cada uno de ellos y que forman en el intervalo eliminado un triángulo equilátero. Con esto se habrá obtenido una curva K_1 formada por cuatro segmentos de longitud $\frac{1}{3}$.

Si se repite este proceso sobre cada uno de los segmentos de K_1 se obtendrá otra curva K_2 formada por 4^2 segmentos de longitud 3^{-2} cada uno de ellos, y así sucesivamente. En el paso n se tendrá una curva K_n formada por 4^n segmentos de longitud 3^{-n}. Esta sucesión de curvas cuando n tiende a infinito, converge a una curva K llamada Curva de Koch. En la Figura 2.9 se muestra las primeras 3 iteraciones de la curva.



(a) Inicio. (b) 1a. iteración. (c) 2a. iteración. (d) 3a. iteración.

Figura 2.9: Construcción de la curva de Koch.

Una de las características de la Curva de Koch es que no tiene tangente en ningún punto, es decir, no es diferenciable. La longitud de toda la curva es infinita, pero está contenida

en un área finita, esto es fácil de ver, ya que en la iteración n se tienen 4^ns segmentos de longitud 3^{-n} por lo que la longitud de la curva en el paso n, K_n, es $\frac{4^n}{3^n}$ y si $n \to \infty$ entonces la longitud de la curva $\lim_{n\to\infty} \frac{4^n}{3^n} = \infty$ tiende a infinito. Además la longitud de la curva entre dos puntos cualesquiera de la curva también es infinita, ya que hay una copia de la curva de Koch entre dos puntos cualesquiera; lo cual es asombroso, ya que hay una distancia infinita entre puntos que se ven 'cercanos'. De aquí se sigue que la distancia entre dos puntos de su perímetro es infinita.

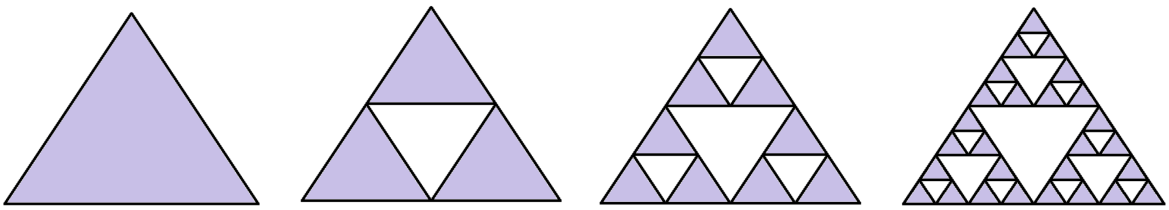
Utilizando la ecuación (2.3), se obtiene que la dimensión fractal de la Curva de Koch es $D = \frac{log(4)}{log(3)} \approx 1.261$, mientras que su dimensión topológica es 1.

Ejemplo 2.2.2. El Triángulo Sierpinski

Para construir el triángulo de Sierpinski, se parte de un triángulo equilátero de longitud de lado 1, llámese T_0. En el primer paso se toma los tres triángulos equiláteros cerrados T_1^1, T_2^1, T_3^1 contenidos en T_0 de lado $\frac{1}{2}$ y que contiene sus vértices como se puede apreciar en la Figura 2.10.

En el siguiente paso se repite el proceso anterior a escala $\frac{1}{2}$ sobre cada uno de los triángulos obtenidos, el número total de triángulos será 3^2 de lado $\frac{1}{4}$ y así sucesivamente, en el paso n se tendrán 3^n triángulos cerrados de lado $\frac{1}{2}^n$.

Este conjunto tiene como características tener perímetro infinito, ser compacto, área nula y donde sus puntos de ramificación forman un subconjunto denso. [46].



(a) Inicio. (b) 1ra iteración. (c) 2da iteración. (d) 3ra iteración.

Figura 2.10: Construcción del Triángulo de Sierpinski.

Finalmente la dimensión fractal del triángulo de Sierpinski es $D = \frac{log(3)}{log(2)} \approx 1.584$ y su dimensión topológica es 1.

Los fractales son objetos matemáticos, los cuales no se pueden encontrar en la naturaleza por su muy estricta construcción infinita; sin embargo, son una excelente herramienta para modelar e intentar comprender la naturaleza, como los problemas que se expusieron en la introducción de este capítulo, un ejemplo de ello es el problema de la costa de Gran Bretaña. Lo que se pretende hacer es encontrar una forma de utilizarlos para modelar un problema; con la finalidad de mejorar estas aproximaciones se crea otro objeto matemático

que no es un fractal estrictamente, pero localmente sí, el cual sería como una aproximación a un fractal.

2.2.1. Fractales Deterministas y Aleatorios

En el momento de analizar fractales es atractivo poder describirlos de una forma detallada, por lo que una noción natural que surge es modelarlos de forma aleatoria para obtener una representación más fiel de la naturaleza y así surgen los siguientes tipos de fractales.

Para los fractales deterministas, como los que se mostraron anteriormente, no importa cuántas veces se vuelvan a construir, siempre se obtendrá el mismo resultado, pues aunque no se pueda ver en su totalidad el fractal, su estructura ya está determinada.

A continuación se da otro ejemplo de un fractal determinista, el cual también es muy famoso en diversas ramas de las matemáticas por tener características muy importantes y peculiares.

Ejemplo 2.2.3 (Conjunto de Cantor)**.** Es un subconjunto de puntos del intervalo $[0,1]$ que se construye a partir de un proceso infinito. Se parte del intervalo unidad $I_0 = [0,1]$ a este se le quita el intervalo abierto central de longitud $\frac{1}{3}$, quedándose con los intervalos $I_1^1 = [0, \frac{1}{3}]$ y $I_2^1 = [\frac{2}{3}, 1]$, es decir $I_1 = I_1^1 \cup I_2^1$.

A cada uno de estos nuevos intervalos se le quita el intervalo abierto central que ahora tendrá longitud $\frac{1}{9}$ obteniendo cuatro intervalos $I_1^2, I_2^2, I_3^2, I_4^2$ de longitud $\frac{1}{9}$ por lo que $I_2 = I_1^2 \cup I_2^2 \cup I_3^2 \cup I_4^2$. Así, sucesivamente en el paso enésimo se tendrá 2^n intervalos $I_1^n, I_2^n, \ldots, I_{2^n}^n$ de longitud $\frac{1}{3}^n$, en general, $I_n = I_1^n \cup I_2^n \cdots \cup I_n^{2^n}$. En Figura 2.11 se muestran las primeras 6 iteraciones para la construcción del conjunto.

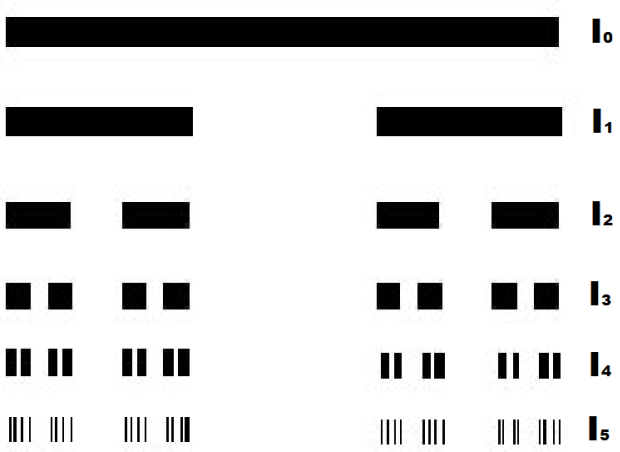


Figura 2.11: Construcción del Conjunto de Cantor.

Entre algunas propiedades, destaca el hecho de que el conjunto tiene medida 0, tiene una cantidad no numerable de puntos y se puede dar una biyección del conjunto con el intervalo $[0, 1]$ y es compacto. Además, en sistemas dinámicos es el conjunto atractor de la función iterada 'tienda' [32, Cap. 10].

Para obtener una representación alternativa del conjunto de Cantor se debe de considerar la expresión ternaria de todos los números $x \in [0, 1]$, entonces por definición x es de la siguiente forma:

$$x = \sum_{i=1}^{\infty} \frac{b_i}{3^i} \quad \text{con} \quad b_i \in \{0, 1, 2\} \text{ para } i \geq 1$$

o escrito de otra forma $x = 0.b_1b_2b_3\ldots$. Por otro lado, obsérvese que para ciertos valores de x, este puede tener una doble expansión ternaria. Por ejemplo, $x = \frac{1}{3}$ y su expansión $x = 0.1$

$$x = \frac{1}{3^1} = \frac{1}{3}$$

Por otro lado, la expansión $x = 0.0222\cdots = 0.0\overline{2}$ también es una representación de $x = \frac{1}{3}$

$$x = \sum_{i=2}^{\infty} \frac{2}{3^i} = \frac{2}{9}\sum_{i=0}^{\infty}\left(\frac{1}{3}\right)^i = \frac{2}{9}\cdot\frac{3}{2} = \frac{1}{3}$$

Hay que notar que los números para los cuales ocurre esta ambigüedad son los números con una expansión finita $x = 0.b_1b_2\ldots b_k$ para alguna k. Por ejemplo, si $b_k = 1$ entonces se puede reemplazar este 1 por un 0 seguida de una cola infinita de 2, si $x = 0.01$ su segunda representación sería $x = 0.00\overline{2}$.

Un método sencillo para obtener la representación ternaria es el siguiente. Si x tiene expansión ternaria $x = 0.b_1b_2b_3\ldots$ entonces b_1 determina a cual de los 3 intervalos del $[0, 1]$ pertenece. Si $b_1 = 0$ entonces $x \in [0, \frac{1}{3}]$, esto se debe a que la cola de b_i no es más grande que $x = 0.00\overline{2}$

$$\sum_{i=2}^{\infty} \frac{b_i}{3^i} \leq \sum_{I=2}^{\infty} \frac{2}{3^i} = \frac{1}{3}$$

Siguiendo el mismo razonamiento $b_1 = 1$ entonces $x \in [\frac{1}{2}, \frac{2}{3}]$ y si $b_1 = 2$ entonces $x \in [\frac{2}{3}, 1]$. Argumentando exactamente de la misma forma b_2 indica en qué tercera parte (izquierda, media o derecha) de estos subintervalos x pertenece. La representación ternaria tiene una relación directa con el conjunto de Cantor, pues si $x = 0.b_1b_2b_3\ldots$ tiene una expansión

ternaria tal que $b_k = 1$ para alguna k, esto implica que x debe estar en alguno de subintervalos medios que son removidos durante la construcción del conjunto de Cantor. La única excepción es si x es un extremo del intervalo, entonces x tiene una expansión ternaria alternativa, que no contiene algún 1 [16].

Por lo tanto, se puede decir que el conjunto de Cantor es el conjunto de números reales pertenecientes al intervalo $[0, 1]$, para los cuales existe una expansión ternaria que no contiene algún 1. Ahora, si dada una x perteneciente al conjunto de Cantor se cambia cada 2 por 1, la secuencia resultante es la expansión binaria de los números pertenecientes al intervalo $[0, 1]$. Con esto se puede concluir que la cardinalidad del conjunto de Cantor es igual a la del intervalo $[0, 1]$ y su representación alternativa está dada por

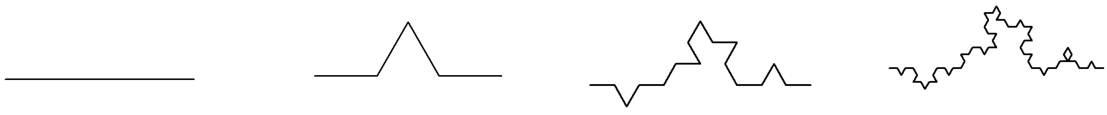
$$C = \left\{ x \in [0,1] : x = \sum_{i=1}^{\infty} \frac{b_i}{3^i} \text{ donde } b_i = 0, 2 \; \forall n \in \mathbb{N} \right\}$$

Por último, tomando la definición de dimensión fractal, se obtiene que $D = \frac{log(2)}{log(3)} \approx 0.631$ y su dimensión topológica es 0.

Los fractales aleatorios son aquellos en que, para su generación actúa el azar. Tienen formas irregulares y por esta razón son los que mejor representan a la naturaleza. Este tipo de fractales tiene la característica de que nunca se formará el mismo fractal aunque la probabilidad p se fije; cada vez que se construya un fractal aleatorio será diferente.

Ejemplo 2.2.4. Curva de Koch aleatoria.

Su construcción es muy similar a lo curva de Koch solo que esta vez se pondrán las dos líneas arriba o abajo con probabilidad p con $p \in (0, 1)$. En la Figura 2.12 se muestra una ejecución de este algoritmo con $p = 1/2$.



(a) Inicio. (b) 1a. iteración. (c) 2a. iteración. (d) 3a. iteración.

Figura 2.12: Construcción del Triángulo de Sierpinski con $p = 1/2$.

Su dimensión fractal se mantiene igual a la de la curva de Koch $D = \frac{log(4)}{log(3)} \approx 1.261$. Este conjunto es utilizado para simular costas [19, págs 244-257].

La deducción de la dimensión en los fractales aleatorios es un poco más elaborada que la de los fractales deterministas, ya que ahora se estará hablando de una familia de fractales

que son todos los posibles fractales que pueden resultar de un algoritmo aleatorio. Estos fractales poseen una dimensión fractal y otra topológica, las cuales suelen no variar entre ellos. Sin embargo, existe un grupo de fractales para los cuales la dimensión fractal toma todo tipo de valores, aunque la probabilidad de este conjunto es nula. Al valor característico de la dimensión fractal de las demás muestras se le denomina valor casi seguro. No se profundizará en el método para la obtención de la dimensión fractal, pues no resulta relevante para el desarrollo del presente trabajo; para más detalles se puede consultar los capítulos IV, VII y VIII del libro *La geometría Fractal de la naturaleza* [35].

2.3. Sistemas de Funciones Iteradas y el Espacio de los Fractales

Los sistemas de funciones iteradas son un conjunto de funciones contractivas que tienen la particularidad de poseer conjuntos atractores. Si las funciones son escogidas adecuadamente, estos atractores pueden ser fractales. Estos sistemas sirven como base de algunos métodos para comprimir datos, en especial imágenes. Por ejemplo, en la Figura 2.17 se observar una hoja de Maple creada a partir de un sistema de funciones iteradas [14, pág. 89].

Definición 2.3.1. Un Sistema (hiperbólico) de funciones iteradas consiste en un espacio métrico completo (X, d) aunado a un conjunto finito de mapeos o funciones que sean contracciones $w_i : X \to X$ con su respectivo factor de contracción s_i para $i = 1, 2, \ldots, n$.

Se suele usar $\{X, w_i, i = 1, \ldots, n\}$ como notación para un sistemas de funciones iteradas, se dice que su factor contractivo es $s = max\{s_n : n = 1, \ldots, N\}$. Usualmente se omite el término hiperbólico y es llamado simplemente sistema de funciones iteradas [8].

Hay que recordar que la norma Euclidiana en $\mathbb{R}^n$ se denota de la siguiente manera

$$\|\overline{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Ejemplo 2.3.1. Sea $0 < \beta < 1$ y $(\mathbb{R}^2, \|\cdot\|_2)$. Sea $z_1, z_2, \ldots, z_n$ puntos en el plano. Defínase $\mathcal{A}(z) = \beta(z - z_i) + z_i$ para cada $i \in \{1, \ldots, n\}$. La colección de funciones $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ es un sistemas de funciones iteradas, ya que cada $\mathcal{A}_i$ es una transformación afín con punto fijo z_i, que además es contracción.

En notación vectorial los sistemas de funciones iteradas en $\mathbb{R}^2$ que utilizan transformaciones afines, se escriben como en la ecuación (2.4), donde $z_i = (a_i, b_i)$ es el punto fijo de $\mathcal{A}_i$

$$\mathcal{A}_i \begin{pmatrix} x \\ y \end{pmatrix} = \beta \begin{pmatrix} x - a_i \\ y - b_i \end{pmatrix} + \begin{pmatrix} a_i \\ b_i \end{pmatrix} \quad i \in \{1, \ldots, n\} \tag{2.4}$$

Estos sistemas dinámicos poseen atractores que resultan interesantes de observar, puesto que son conjuntos fractales. Originalmente a estos conjuntos se les llama atractores extraños, sin embargo, hoy en día se denomina así a un atractor que exhibe sensibilidad a condiciones iniciales [53]. En gran parte de este trabajo se hará uso mayormente de SFI cuyas funciones son transformaciones afines.

Para producir un fractal, se escoge arbitrariamente un punto inicial en el plano y se itera su órbita bajo la función $\mathcal{A}_i$ que es escogida aleatoriamente. Se puede demostrar que con probabilidad 1 la órbita converge a un subconjunto en el plano, el cual es el atractor [16, Pág. 199]. Los siguientes ejemplos fueron tomados del libro *A First Course in Chaotic Dynamical Systems: Theory and Experiment* [16]

Ejemplo 2.3.2. Supóngase que se tiene el siguiente sistema de funciones iteradas

$$\mathcal{A}_1 \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{3} \begin{pmatrix} a \\ b \end{pmatrix} \quad \mathcal{A}_2 \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{3} \begin{pmatrix} a - 1 \\ b \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.5}$$

Los puntos fijos atractores corresponden al $(0, 0)$ y $(1, 0)$ y el factor de contracción es $\frac{1}{3}$.

Hay que notar que, independiente de cualquier punto con el que se comience, este tiende al eje x. Sea $x_n = (a_n, b_n)$ el enésimo punto, hay observar que $b_{k+1} = \frac{b_k}{3}$ sin importar la función que se escoge, esto quiere decir, que b_n tiende a cero con una tasa geométrica. Es decir,

$$x_n = \begin{pmatrix} a_n \\ b_n \end{pmatrix} \quad \text{donde} \quad b_n = \frac{b_0}{3^n} \quad \forall\, n \in \mathbb{N}$$

Para analizar el comportamiento en la primera entrada, defínase una sucesión $s = (s_1, s_2, \ldots)$, donde cada s_j es 0 o 1, si se escogió a la función $\mathcal{A}_1$ o $\mathcal{A}_2$ respectivamente.

Por ejemplo, si tiene esta secuencia $s = (0\ 1\ 0\ 1\ 0\ 1 \ldots)$ entonces las coordenadas de las iteraciones tienen la siguiente forma:

$$a_1 = \frac{a_0}{3}, \quad a_2 = \frac{a_0}{3^2} + \frac{2}{3}, \quad a_3 = \frac{a_0}{3^3} + \frac{2}{3^2}, \quad a_4 = \frac{a_0}{3^4} + \frac{2}{3^3} + \frac{2}{3} \quad \ldots$$

Se pueden pensar las sumas anteriores como:

$$a_n = \begin{cases} \frac{a_0}{3^n} + \frac{2}{3^{n-1}} + \frac{2}{3^{n-3}} + \frac{2}{3^{n-5}} + \cdots + \frac{2}{3} & \text{si } n \text{ es par} \\ \frac{a_0}{3^n} + \frac{2}{3^{n-1}} + \frac{2}{3^{n-3}} + \frac{2}{3^{n-5}} + \cdots + \frac{2}{3^2} & \text{si } n \text{ es impar} \end{cases}$$

En general, para cualquier iteración f^n con una sucesión $s = (s_1, s_2, \dots)$ asociada, se puede describir de la siguiente forma:

$$a_n = \frac{a_0}{3^n} + \left(\frac{2s_1}{3^n} + \frac{2s_2}{3^{n-1}} + \frac{2s_3}{3^{n-2}} \cdots + \frac{2s_n}{3^1} \right)$$

donde $s_1 = s_3 = s_5 = \cdots = 0$ y $s_2 = s_4 = s_6 = \cdots = 1$. Si se toma el límite cuando x_n tiende a infinito, entonces

$$\lim_{n\to\infty} a_n = \frac{a_0}{3^n} + \left(\frac{2s_1}{3^n} + \frac{2s_2}{3^{n-1}} + \frac{2s_3}{3^{n-2}} \cdots + \frac{2s_n}{3^1} + \dots \right) = \sum_{i=1}^{\infty} \frac{t_i}{3^n}; \quad t_i \in \{0, 2\} \tag{2.6}$$

El primer término tiende a cero cuando n tiende a infinito, esto quiere decir que, en el límite la órbita es independiente de a_0, el punto inicial. El término t_i toma valores en $\{0, 2\}$ dependiendo de la s_i, es importante percatarse que la serie de la ecuación (2.6) no converge a un solo punto ya que depende de la sucesión; más bien, es la representación de los puntos pertenecientes al conjunto de Cantor como se vio en el Ejemplo 2.2.3.

Por ejemplo, considérese la sucesión $s = (1, 1, 1, \dots)$ entonces

$$\lim_{n\to\infty} a_n = \sum_{i=1}^{\infty} \frac{2}{3^n} = 2\left(-1 + \sum_{n=0}^{\infty} \frac{1}{3^n} \right) = -2 + \frac{2}{1 - \frac{1}{3}} = 1$$

El cual es un punto dentro del conjunto de Cantor. En conclusión, cuando $n \to \infty$ el conjunto de puntos $\{x_N, x_{N+1}, \dots\}$ para una $N > n$ tal que $n, N \in \mathbb{N}$, converge el conjunto de Cantor. Los cuales son el resultado de iterar un punto aleatorio x_0 en el sistema de funciones iteradas (2.5).

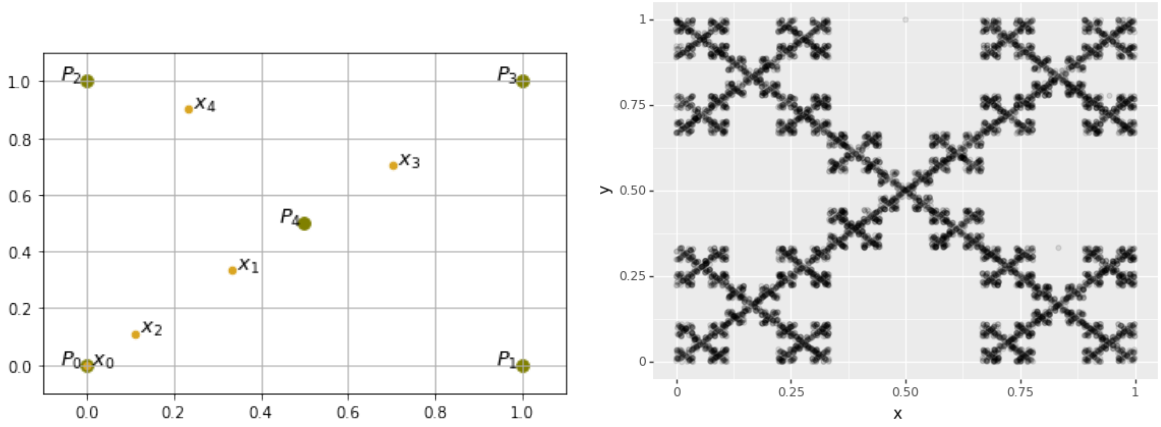
Con los Sistemas de funciones iteradas (SFI) se pueden construir una infinidad de fractales. A continuación se muestra un ejemplo gráfico.

Considérese un factor de contracción $\beta = 1/3$, los siguientes puntos fijos y funciones contractivas

$$x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad x_4 = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix},$$

$$\mathcal{A}_i(x) = \frac{1}{3}(x - x_i) + x_i. \tag{2.7}$$

La Figura 2.13 es el conjunto atractor, el cual es resultado de iterar un punto aleatorio $x_0 = (\frac{1}{2}, 1)$ en el sistema (2.7), para más información ver el algoritmo del Apéndice D.2,



(a) Primeras 5 iteraciones.

(b) Atractor resultante.

Figura 2.13: Conjunto atractor del sistema de funciones iteradas (2.7).

Estos sistemas pueden tener variaciones. Por ejemplo, en cada iteración contraer con un factor β, aplicar una matriz de rotación con p_0 punto atractor

$$A\begin{pmatrix} x \\ y \end{pmatrix} = \beta \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \cdot \begin{pmatrix} x - a_0 \\ y - b_0 \end{pmatrix} + \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \tag{2.8}$$

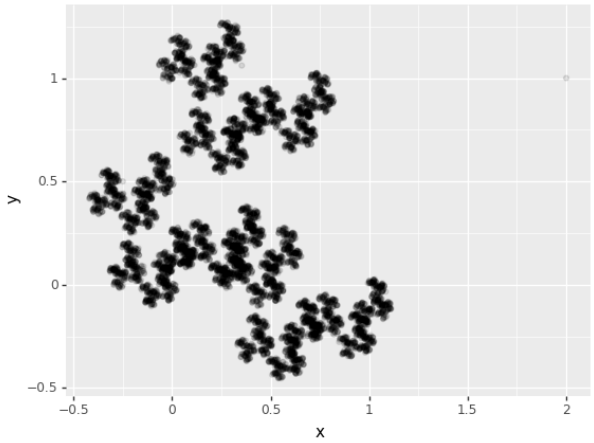


Figura 2.14: Sistema con rotación con $\beta = 1/2$, $\theta = \pi/4$ y tres puntos atractores $(0, 0)$, $(1, 0)$ y $(0, 1)$.

La Figura 2.14 es conjunto atractor, el cual es resultado de iterar un punto aleatorio $x_0 = (2, 1)$ en el sistema (2.8), para más información ver el algoritmo del Apéndice D.3,

2.4. Visualización de Sistemas de Funciones Iteradas

Normalmente se utilizan dos algoritmos para programar un sistema de funciones iteradas, el determinista y el aleatorio. El determinista consiste en aplicar a un conjunto cualquiera cada una de las N diferentes funciones, obteniendo N conjuntos y a la unión de dichos conjuntos se le vuelve a aplicar las N funciones.

Por otro lado, el algoritmo aleatorio está fundamentado en la teoría ergódica. En dicho algoritmo se aplican las funciones a un único punto, a cada transformación se le asigna una probabilidad. La elección de dichas probabilidades determina la forma y la densidad del atractor (ver Figuras 2.17 y 2.18). Esta subsección esta basada principalmente en los capítulos III,IV y IX del libro *Fractals Everywhere* [8].

2.4.1. Algoritmo Determinista

Para sustentar el procedimiento determinista se presentan los siguientes conceptos.

Definición 2.4.1. Sea X un conjunto. Una métrica (o distancia) en X es una función $d : X \times X \to \mathbb{R}$ que tiene las siguientes tres propiedades:

1. $d(x, y) = 0$ si y solo si $x = y$.
2. $d(x, y) = d(y, x)$ para cualesquiera $x, y \in X$.
3. $d(x, z) \leq d(x, y) + d(y, z)$ para cualesquiera $x, y, z \in X$. A esta desigualdad se le llama desigualdad de triángulo.

Un espacio métrico es un conjunto de X provisto de una métrica d. Se le denota por (X, d). [38]

Definición 2.4.2. Sea (X, d) un espacio métrico y dos subconjuntos no vacíos A y B de X. La métrica de Hausdorff se define:

$$h(A, B) = max\left\{sup\left\{d(a, B) : a \in A\right\}, sup\left\{d(b, A) : b \in B\right\}\right\}$$

donde

$$d(a, B) = inf\left\{d(a, b) : b \in B\right\}$$

Se denota a $\mathcal{H}(X)$ como el espacio correspondiente de subconjuntos compactos no vacíos con métrica de Hausdorff $h(d)$ de un espacio métrico (X, d). Se puede probar que la métrica de Hausdorff es en efecto una métrica, para más detalles ver el capítulo 5 de *Lectures on Hausdorff and Gromov–Hausdorff Distance Geometry* [1].

Esta métrica mide qué tan lejos están dos subconjuntos de un espacio métrico entre sí. Es la mayor de todas las distancias desde un punto en un conjunto al punto más cercano en el otro conjunto. Fue introducida por Ponpeiu y Hausdorff; en su honor se le suele llamar la métrica de Hausdorff o métrica de Ponpeiu-Hausdorff. La demostración del siguiente teorema se puede encontrar en [8, pág. 81].

Teorema 2.4.1. Sea $\{X, w_n, n = 1, 2, \ldots, N\}$ un sistema de funciones iteradas hiperbólico con un factor contractivo s. Entonces la transformación $W : \mathcal{H}(X) \to \mathcal{H}(X)$ definida por

$$W(B) = \bigcup_{n=1}^{N} w_n(B)$$

para todo $B \in \mathcal{H}(X)$, es un mapeo contractivo en el espacio métrico completo $(\mathcal{H}(X), h(d))$ con un factor contractivo s. Y satisface

$$h(W(B), W(C)) \leq s \cdot h(B, C)$$

para todo $B, C \in \mathcal{H}(X)$. Su único punto fijo, $A \in \mathcal{H}(X)$ satisface

$$A = W(A) = \bigcup_{n=1}^{N} w_n(A)$$

y está dado por $A = \lim_{k\to\infty} W^k(B)$ para toda $B \in \mathcal{H}(X)$.

La secuencia de iteraciones $w^k(A)$ converge con la distancia Hausdorff al atractor A.

Definición 2.4.3. El punto fijo $A \in \mathcal{H}(X)$ descrito en el teorema anterior es llamado el atractor del sistema de funciones iteradas $\{X, w_n, n = 1, \ldots, N\}$.

El atractor puede considerarse como una generalización de puntos fijos de transformaciones contractivas [30].

De modo que, el siguiente algoritmo genera una aproximación al atractor de un sistema de funciones iteradas.

Algoritmo 2.4.1. Sea $\{X, w_1, \ldots, w_n\}$ un sistema de funciones iteradas hiperbólico. Elíjase un subconjunto compacto $A_0 \subset \mathbb{R}^2$, entonces, se calcula sucesivamente $W^{\circ k}(A)$ de

acuerdo a

$$A_{k+1} = \bigcup_{j=1}^{n} w_j(A_k) \quad para\ k \in \{1, \ldots, M\}$$

Por el Teorema 2.4.1, la secuencia $\{A_k : k = 0, 1, 2, 3 \ldots M\} \subset \mathcal{H}(X)$ converge al atractor del sistema de funciones iteradas con la métrica de Hausdorff, si M es suficientemente grande [8, Pág. 85],

Es importante notar que, no necesariamente se tiene que utilizar un conjunto de transformaciones afines que sean contracciones, como se usó en los ejemplos anteriores, se puede hacer uso de transformaciones no lineales. En este ejemplo, se emplean transformaciones de Möbius, estas funciones son de la forma:

$$f(z) = \frac{az + b}{cz + d}$$

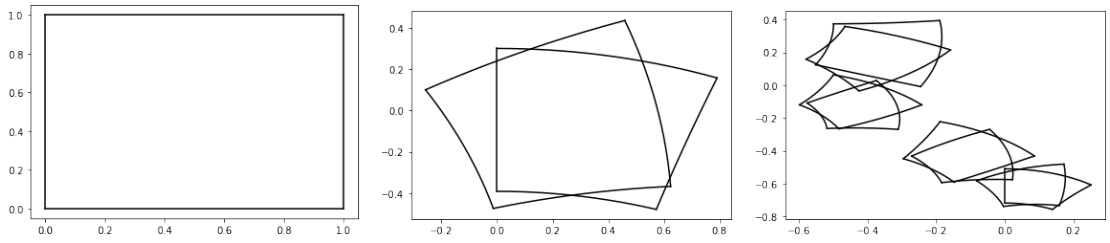
donde z, a, b, c, d son números complejos que cumplen que $ad - bc \neq 0$.

Ejemplo 2.4.1. Sea $X = \mathbb{C}$ y

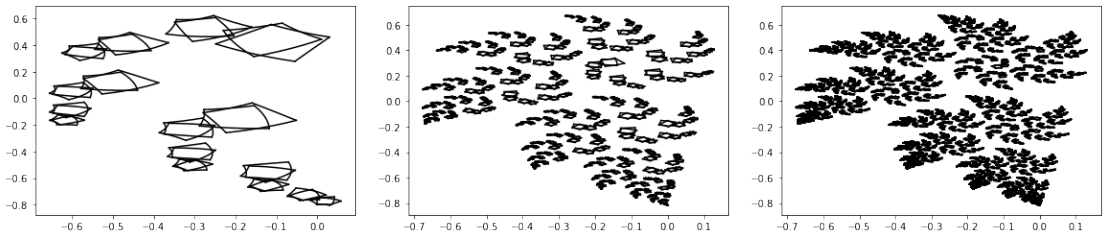
$$w_1(z) = \frac{(0.6966 - 0.4607i) \cdot z - 0.3307 - 0.0497i}{0.2i \cdot z + 1.0523 + 0.601i}$$
$$w_2(z) = \frac{-0.8438 \cdot z + 0.5119i}{-0.2i \cdot z + -1.3064}$$

Se elige como conjunto a iterar un cuadrado como el que se ve en la Figura 2.15a. La elección de este conjunto es por conveniencia, ya que su parametrización es sencilla; sin embargo, se puede iterar cualquier otro conjunto. En el Apéndice D.4 se adjunta el código escrito en Python3 utilizado para la ejecución del algoritmo.

En las Figuras de 2.15 se distingue la transformación del conjunto original, formando un fractal.



(a) Conjunto inicial (b) 1a. iteración (c) 3a. iteración



(d) 5a. iteración (e) 10a. iteración (f) 13a. iteración

Figura 2.15: Algoritmo Determinista con funciones no lineales.

Obsérvese que es sencillo deducir cuántas iteraciones se tienen que hacer si se quiere que la iteración A_k tengo un error de a lo más ϵ,

$$h(A_{k+1}, A_k) = h(W^k(A_1), W^k(A_0)) \leq s^k \cdot h(A_1, A_0)$$

Además para cualesquiera $B, C \in X$, se cumple que:

$$\begin{aligned} h(B, C) &\leq h(B, W(B)) + h(W(B), W(C)) + h(W(C), C) \\ &\leq h(B, W(B)) + S \cdot h(B, C) + h(W(C), C) \end{aligned}$$

es decir,

$$(1 - s) \cdot h(B, C) \leq h(B, W(B)) + h(W(C), C)$$

Tomando $B = A_k$ y $C = A_j$

$$h(A_k, A_j) \leq \frac{h(A_k, W(A_k)) + h(W(A_j), A_j)}{1 - s} \leq \frac{s^k + s^j}{1 - s} h(x_1, x_0)$$

Sea $\epsilon > 0$. Como $s \in (0, 1)$ existe $k_0 \in \mathbb{N}$ tal que:

$$\frac{s^k}{1 - s} h(A_1, A_0) \leq \frac{\epsilon}{2} \quad \forall k \geq k_0 \tag{2.9}$$

Como consecuencia, si se quiere una aproximación con en error de ϵ solo se tiene que encontrar la k que satisfaga la ecuación (2.9) y k será el número de iteraciones que se tiene que hacer.

2.4.2. Algoritmo Aleatorio

Para definir el algoritmo aleatorio se tiene que asignar una probabilidad a cada una de las funciones en el sistema.

Definición 2.4.4. Un sistema de funciones iteradas con probabilidades consiste en un sistema de funciones iteradas

$$\{X; w_1, w_2, \ldots, w_N\}$$

junto con un conjunto ordenado de números $\{p_1, p_2, \ldots, p_N\}$ tal que

$p_1 + p_2 + \cdots + p_N = 1$ y $p_i > 0$ para $i = 1, 2, \ldots, N$.

Algoritmo 2.4.2. Sea $\{X, w_1, \ldots, w_N\}$ un sistema de funciones iteradas hiperbólico con probabilidades $\{p_1, p_2, \ldots, p_N\}$. Se escoge $x_0 \in X$ y se escoge otro recursivamente de forma independiente

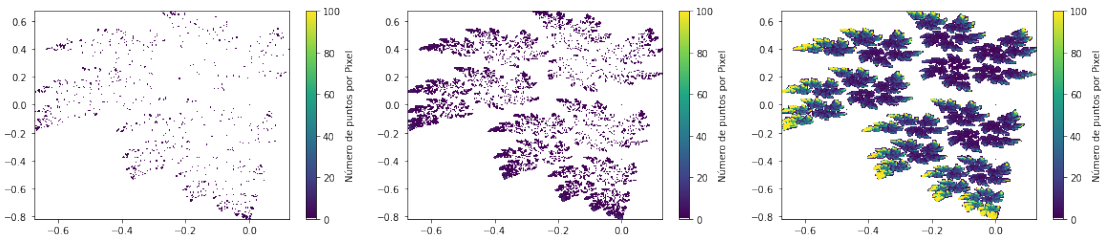
$$x_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), \ldots, w_N(x_{n-1})\} \quad para\ n = 1, 2, 3 \ldots$$

donde la probabilidad del evento $\{x_n = w_i(x_{n-1})\}$ es p_i. Así se construye la secuencia $\{x_n : n = 0, 1, 2, 3 \ldots\} \subset X$.

Ejemplo 2.4.2. Para ilustrar el Algoritmo 2.4.2, se toma el mismo sistema de funciones iteradas del Ejemplo 2.4.1, un punto al azar en el plano y ambas funciones serán equiprobables de elegir.

Sea $X = \mathbb{C}$ y

$$w_1(z) = \frac{(0.6966 - 0.4607i) \cdot z - 0.3307 - 0.0497i}{0.2i \cdot z + 1.0523 + 0.601i} \qquad p_1 = \frac{1}{2}$$
$$w_2(z) = \frac{-0.8438 \cdot z + 0.5119i}{-0.2i \cdot z + -1.3064} \qquad p_2 = \frac{1}{2}$$



(a) $N = 1000$ (b) $N = 10000$ (c) $N = 500000$

Figura 2.16: Algoritmo Aleatorio con funciones no lineales.

En el Apéndice D.5 se agrega el código en Python3 para implementar el Algoritmo 2.4.2. La Figura 2.16 muestra la evolución del algoritmo mediante gráficas de densidad.

Lo más natural sería preguntarse por la convergencia de este algoritmo para asegurarse que la órbita de x_0 que se obtiene como resultado de este algoritmo converja al mismo atractor que en el algoritmo determinista. Debido a la alta complejidad de la demostración de esta afirmación solo se dará un esbozo de esta, pero si se quiere ver una explicación detallada consultar el capítulo IX del libro *Fractals Everywhere*[8].

El algoritmo aleatorio proporciona la idea de masa, ya que la densidad de puntos por píxel en la figura varía de acuerdo a las probabilidades asignadas, esto se aprecia mejor en las Figuras 2.17 y 2.18. Se necesita un nuevo concepto matemático para describir esta densidad, tal concepto es el de 'medida'. Las medidas se pueden utilizar para describir distribuciones de 'masa' en espacios métricos, por esta razón se tiene que explorar y formalizar el concepto de medida en un espacio métrico. Para ver las nociones básicas de Teoría de la medida consultar el Apéndice C.1.

Se podría cuestionar acerca de esta medida, ¿cómo se sabe que esta medida siempre es la misma?, la respuesta no es trivial, ya que se tienen que introducir nuevos conceptos como la métrica Hutchinson, el operador de Markov asociado a un sistema de funciones iteradas junto con varios teoremas que se pueden ver en el Apéndice C.4, todas estas fórmulas están motivadas para darle una justificación matemática a la fórmula (2.10).

Sea A el atractor del SFI, entonces existe una medida invariante del SFI, el cual se denotará por μ, tal que μ asigna 'masa' a varios subconjuntos de X. Por otro lado, los subconjuntos de X tales que tienen 'masa' son llamados subconjuntos de Borel de X y se denotan por $\mathcal{B}(X)$.

Sea B la bola cerrada en X. Para calcular la 'masa' de la bola $\mu(B)$, se ejecuta el algoritmo aleatorio al SFI con probabilidades para producir la secuencia $\{z_i\}_{i=0}^{n}$ entonces

$$\mathcal{N}(B) = \text{número de puntos en } \{z_0, z_1, \ldots, z_n\} \cap B$$

entonces casi seguramente

$$\mu(B) = \lim_{n\to\infty} \frac{\mathcal{N}(B,n)}{n+1} \tag{2.10}$$

Al final se llega a la conclusión de que esta medida es invariante, así se asegura la convergencia de los sistemas de funciones iteradas con probabilidades.

2.5. Aplicación al Diseño de Imágenes

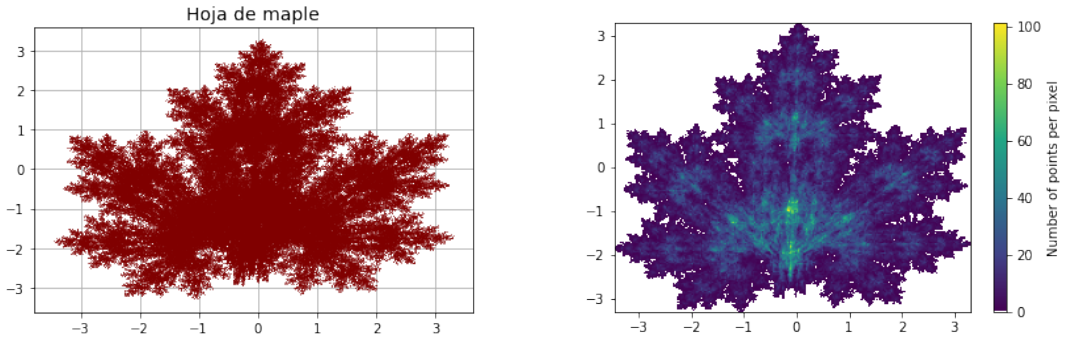
Para simplificar la notación se puede ver cada transformación afín $\mathcal{A}_i : \mathbb{R}^2 \to \mathbb{R}^2$ con una probabilidad p_i, de la siguiente manera:

$$\mathcal{A}_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

En el libro *fractals everywhere* [8], Barnsley ilustra cómo se pueden utilizar los sistemas de funciones para la creación de imágenes realistas a través de múltiples ejemplos que son muy famosos como *the spleenwort fern* cuyo atractor tiene forma de helecho. Uno de los sistemas que enuncia es el siguiente:

	a	**b**	**c**	**d**	**e**	**f**	**p**
A_1	0.14	0.01	0.00	0.51	-0.08	-1.31	0.10
A_2	0.43	0.52	-0.45	0.50	1.49	-0.75	0.35
A_3	0.45	-0.49	0.47	0.47	-1.62	-0.74	0.35
A_4	0.49	0.00	0.00	0.51	0.02	1.62	0.20

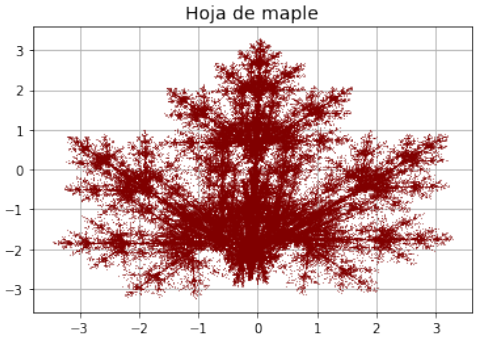
Cuadro 2.1: Sistema de funciones iteradas para la hoja de Maple.



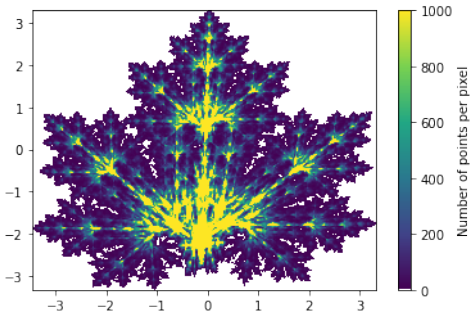
(a) Hoja de Mapple.

(b) Hoja de Mapple iluminada de acuerdo a su densidad.

Figura 2.17: Representación gráfica del sistema de funciones iteradas del Cuadro 2.1, con probabilidades $[0.25, 0.25, 0.25, 0.25]$.



(a) Hoja de Mapple con cambio probabilidades.



(b) Hoja de Mapple con cambio probabilidades iluminada de acuerdo a su densidad.

Figura 2.18: Representación gráfica del sistema de funciones iteradas del Cuadro 2.1 con probabilidades $[0.3, 0.25, 0.25, 0.20]$.

Es preciso señalar que la cantidad de iteraciones es la misma en las Figuras 2.17b y 2.18b, sin embargo, la distribución de puntos es diferente. En la Figura 2.17 se puede apreciar una distribución uniforme o equitativa de las iteraciones dentro de atractor cuya forma se parece a una hoja de maple. En la Figura 2.18 se ve una distribución más concentrada en ciertas áreas dentro del atractor como si se tratase del tallo de la hoja de maple, esto se debe al cambio de probabilidades asignadas a cada función del sistema. Se puede concluir que la forma es la misma pero la '*densidad*' del atractor final se ve afectado por el conjunto de probabilidades que se escojan.

Si se quisiera considerar en el problema inverso, donde dada una imagen se quiera encontrar el sistema de funciones iteradas con probabilidades que dé como resultado una textura que se asimile mucho a la imagen original, esto se puede lograr. Esto tiene su sustento en un teorema llamado *The Collage Theorem for Measures*. No obstante, este problema no es relevante para el presente trabajo, para más información véase el Capítulo IX del libro *Fractals Everywhere*[8].

2.5.1. Relación Entre los Algoritmos

Al comparar el resultado de ambos algoritmos lo que se puede concluir es que el resultado es el mismo, ambos convergen al mismo atractor. Sin embargo, la diferencia de estos algoritmos se encuentra en el costo computacional, el cual es de vital importancia para este estudio.

El algoritmo aleatorio converge rápidamente y es muy fácil de programar, como se ve en el apartado D.5. Tiene un orden de complejidad lineal $\mathcal{O}(n)$, pues solo se tiene que evaluar el punto anterior en la función que se escoja al azar.

Por otro lado, el algoritmo determinista tiene un orden exponencial $\mathcal{O}(N^M)$, donde M es el número de iteraciones en la aproximación y N es número de funciones en nuestro sistema. En el Ejemplo 2.4.1 se tiene $N = 2$ y un conjunto de cuatro lados, a estos se les tienen que aplicar ambas funciones por lo que se realizan $4 \cdot 2 = 8$ operaciones. En el siguiente paso a esos 8 conjuntos se les aplican ambas funciones, en total se tiene que hacer $4 \cdot 2^2 = 16$ operaciones. Siguiendo el mismo razonamiento en el paso M el número de operaciones a ejecutar es $4 \cdot 2^M$ y así se concluye que su orden de complejidad es exponencial; es importante mencionar computacionalmente no es un algoritmo óptimo ni eficiente.

En conclusión, el algoritmo determinista está asociado directamente con el atractor del sistema y es más intuitivo, pues se ven las transformaciones que le están ocurriendo a nuestro conjunto en cada iteración; eso da una mejor noción del por qué se va formando el fractal. En cambio, el algoritmo aleatorio es el que se usa para generar imágenes por su versatilidad computacional. Su implementación resulta ser más sencilla y práctica, aunque no es muy intuitivo porque se forman imágenes fractales.

Capítulo 3

Representación del ADN con el Juego del Caos

El Juego del Caos es un sencillo algoritmo propuesto por Michael Barnsley en 1980. Este solo requiere la presencia de un triángulo cualquiera en el plano con vértices A, B y C.

La regla del juego es la siguiente: Dado un punto cualquiera en el plano llámese P_0, escoger un vértice del triángulo al azar. A partir de este vértice se determina un nuevo punto al cual se le denotará P_1, este estará posicionado en el punto medio entre el vértice elegido y el punto inicial P_0. Posteriormente se repite este proceso, pero ahora con P_1, se elige un vértice del triángulo de forma aleatoria y se obtiene P_2 como producto de tomar el punto medio entre el vértice y P_1. Se aplica esta regla descrita una gran cantidad de veces y el resultado es una sucesión de puntos.

La pregunta es ¿cuál es el resultado de graficar los puntos de esta sucesión en el plano? La respuesta no es obvia, ya que se podría pensar que el resultado son puntos dispersos dentro del triángulo; sin embargo, el producto que se obtiene no es más que el triángulo Sierpinski; fractal visto en el Ejemplo 2.2.2. Este es un resultado sorprendente, ya que se empezó con un punto arbitrario en el plano, se construyó la secuencia de puntos a partir de elecciones aleatorias y se obtuvo un conjunto que claramente no es aleatorio.

El Juego del Caos es un caso particular de un sistema de funciones iteradas aleatorio cuyos puntos fijos son los vértices A, B y C de triángulo y la probabilidad de elección es $\frac{1}{3}$ para todas las funciones, como se describe en las ecuaciones 3.1 y 3.2. Como se mencionó en la sección anterior, estos sistemas poseen un conjunto atractor que en este caso es el triángulo Sierpinski. En la Sección 3.1.1 se indagará más acerca de este peculiar algoritmo y su atractor.

$$w_1\begin{pmatrix}x\\y\end{pmatrix}=\frac{1}{2}\begin{pmatrix}x-A_x\\y-A_y\end{pmatrix}+\begin{pmatrix}A_x\\A_y\end{pmatrix}\quad w_2\begin{pmatrix}x\\y\end{pmatrix}=\frac{1}{2}\begin{pmatrix}x-B_x\\y-B_y\end{pmatrix}+\begin{pmatrix}B_x\\B_y\end{pmatrix}\tag{3.1}$$

$$w_3\begin{pmatrix}x\\y\end{pmatrix}=\frac{1}{2}\begin{pmatrix}x-C_x\\y-C_y\end{pmatrix}+\begin{pmatrix}C_x\\C_y\end{pmatrix}\quad \text{con}\quad p_1,p_2,p_3=\frac{1}{3}\tag{3.2}$$

Este algoritmo se puede generalizar para polígonos de k lados, lo que se tiene que hacer es ampliar el sistema de funciones iteradas a uno de tamaño k y los puntos fijos deben coincidir con cada vértice del polígono. Casi siempre el resultado de este algoritmo aplicado a diferentes polígonos resulta ser una imagen fractal. No obstante, hay una notable diferencia en el caso de $k = 4$ la cual se verá con detalle en la Sección 3.1.2, pues su atractor es todo el polígono, por lo cual el interior se llena de puntos distribuidos uniformemente, y esta característica será de gran utilidad para el desarrollo de esta investigación.

En 1990, Jeffrey Joel propuso ciertas modificaciones al Juego del Caos para el caso de un polígono regular de 4 lados, donde a cada vértice se le asigna un nucleótido. La fuente de elecciones de los 4 vértices es la secuencia genética que ofrece cualquier genoma, este nuevo método es útil para estudiar e interpretar la estructura del genoma de cualquier especie.

3.1. Juego del Caos

Definición 3.1.1 (Juego del Caos Generalizado)**.** Para $k \geq 3$. Considérese k puntos $P_1, \ldots, P_k$ en una posición convexa y $P_i = (P_i^x, P_i^y)$. Defínase las funciones $\mathcal{A}_1, \ldots, \mathcal{A}_k$ donde

$$\mathcal{A}_i\begin{pmatrix}x\\y\end{pmatrix}=\frac{1}{2}\begin{pmatrix}x-P_i^x\\y-P_i^y\end{pmatrix}+\begin{pmatrix}P_i^x\\P_i^y\end{pmatrix}=\frac{1}{2}\begin{pmatrix}x\\y\end{pmatrix}+\frac{1}{2}\begin{pmatrix}P_i^x\\P_i^y\end{pmatrix}\tag{3.3}$$

El juego del Caos es la implementación de este sistema de funciones iteradas como se definió en el Algoritmo 2.4.2. Con la particularidad de que cada $\mathcal{A}_i$ es una transformación lineal cuyo punto fijo es P_i, que además, cada una es una contracción con factor $\beta = \frac{1}{2}$.

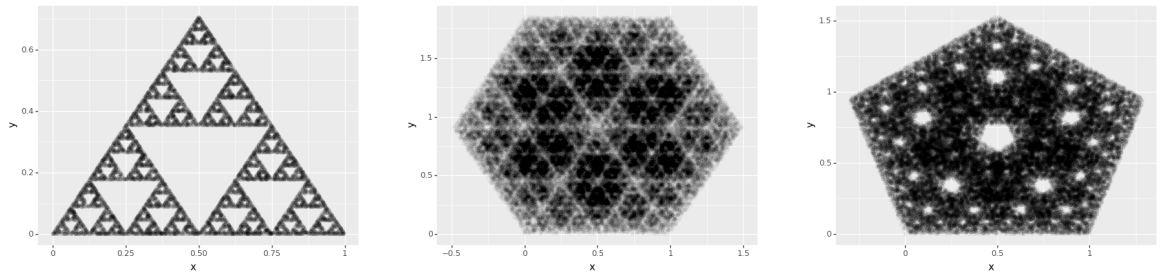
Algoritmo 3.1.1. [El Juego del Caos Generalizado] Sea $P_1, P_2, \ldots, P_k$ los vértices de un polígono con $k \geq 3$, $i \in \{1, \ldots, k\}$ y $n = 0, \ldots, N$.

1. Sin pérdida de generalidad se comienza tomando un punto z_0 al azar dentro del polígono. Este punto corresponde al punto inicial. Se hace $n = 0$.

2. Elegir de manera equiprobable uno de los vértices $P_1, \ldots, P_k$, nómbrese $(P_i)_n$ al i-ésimo vértice seleccionado en la enésima iteración.

3. Se hace z_{n+1} al punto medio entre P_{i_n} y z_n, se actualiza $n = n + 1$ y se vuelve al paso 2 hasta que $n = N$.

El juego del Caos se puede realizar con cualquier polígono, por ejemplo: triángulos, hexágonos, pentágonos, etc. En casi todos los casos el conjunto atractor del sistema de funciones iteradas resultante es un fractal, sin importar que no sea un polígono regular. A continuación se muestran simulaciones del Juego del Caos para polígonos de 3, 5 y 6 lados. En la Figura 3.1 se muestran las simulaciones realizadas con el software Python3 de este algoritmo, el código se puede consultar en la apéndice E.1.



(a) Triángulo. (b) Hexágono. (c) Pentágono.

Figura 3.1: Juego del Caos con distintos polígonos.

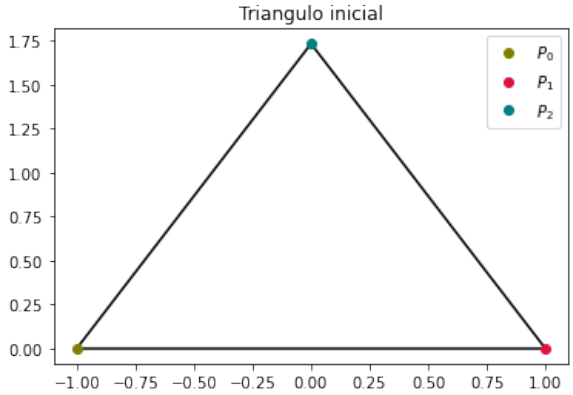
Cabe destacar que todas las funciones son contractivas, de modo que, si el Algoritmo 2.4.2 comenzara en un punto arbitrario en el plano, entonces eventualmente una de las iteraciones de x_0 caería dentro del polígono $P_1, \ldots, P_k$ y la órbita de x_0 convergería al atractor del sistema. Además, en la mayoría de casos, las probabilidades de elección que son distintas de cero, es decir, ningún vértice tiene asignada la probabilidad igual a cero, la forma de fractal es la misma, sin embargo, la distribución de los puntos en el fractal es la que varía de acuerdo a las probabilidades asignadas, como se vio en las Figuras 2.17 y 2.18.

3.1.1. Análisis del Juego del Caos

Primero se demostrará que el atractor del Juego del Caos en el caso $k = 3$ es homotético al Triángulo Sierpinski. Más aún, si se tiene que el punto x_n está en cierta región del triángulo $P_1P_2P_3$ entonces se puede inferir cuáles fueron las últimas elecciones (aleatorias) de algoritmo. Está subsección esta basada en la unidad 2 del libro *Fractals for the Classroom: Strategic Activities Volume One* [43]

Tómese un triángulo cualquiera en el plano cuyos vértices sean P_0, P_1 y P_2, en la Figura 3.2a se muestra el triángulo que se utilizará. Sin perdida de generalidad se supondrá que el punto inicial está dentro del triángulo.

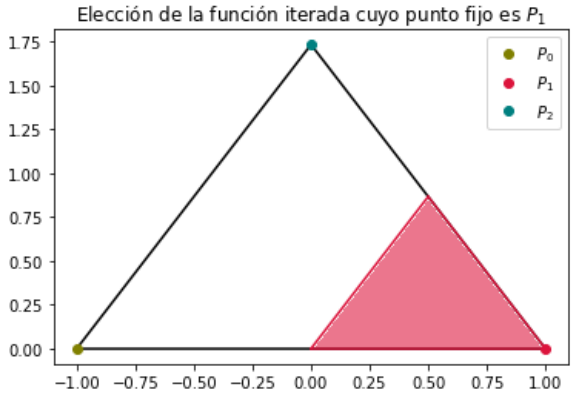
Siguiendo el algoritmo del Juego del Caos, el siguiente paso es elegir un vértice al azar. Supóngase que se eligió el vértice P_0, cualquier punto que se encuentre en el triángulo al aplicarle la transformación lineal, será desplazado a la mitad de la distancia que tiene con respecto al punto P_0. En la Figura 3.2b se ve en verde la zona en la que caerá el punto en el siguiente paso.



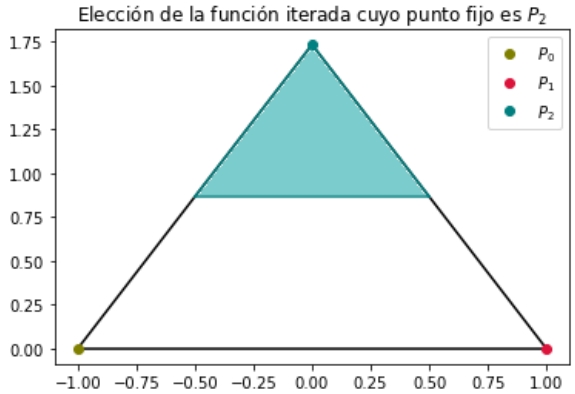
(a) Triángulo inicial.

(b) En color verde se nuestra la transformación del triángulo original al elegir el vértice P_0.

Figura 3.2: Análisis Juego del Caos.



(a) En color rojo se nuestra la transformación del triángulo original al elegir vértice P_1.



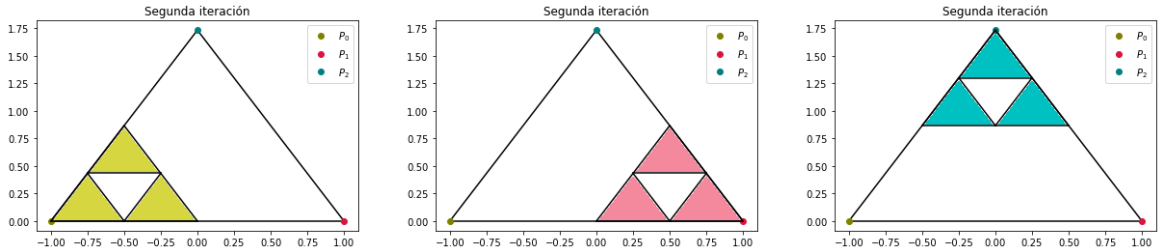
(b) En color jade se nuestra la transformación del triángulo original al elegir vértice P_2.

Figura 3.3: Análisis del Juego del Caos, 1^a iteración.

Lo mismo pasa si elige alguno de los otros dos vértices, en cada elección se aplica una transformación lineal que contrae al triángulo hacia el vértice que se eligió con un factor

de contracción $\beta = \frac{1}{2}$, en la Figura 3.3 se muestran las otras dos alternativas. Por lo tanto, en la primera iteración solo hay tres zonas a las que se podría acceder, la cual es resultado de la unión de las zonas formadas al aplicarle las distintas transformaciones al triángulo como se ve en el Figura 3.5a.

Siguiendo el mismo razonamiento, se puede analizar lo que pasará en la siguiente iteración. Cualquier punto en la zona accesible, después de la primera iteración, será desplazado a la mitad de la distancia que tiene con respecto al punto P_i para $i = 0, 1, 2$, en la Figura 3.4 se observa las distintas zonas a las que se puede acceder en la segunda iteración. El conjunto al que se puede acceder en la segunda iteración es el resultado de la unión de los conjuntos que se obtienen al aplicar cada distinta transformación a la región accesible que se obtuvo en la iteración anterior. Este nuevo conjunto es el que se muestra en la Figura 3.5b.



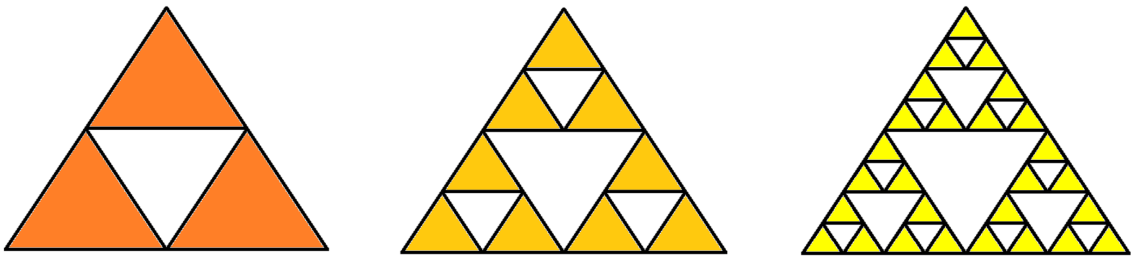
(a) Transformación de la zona accesible si la elección fue el vértice P_0.

(b) [Transformación de la zona accesible si la elección fue el vértice P_1.

(c) [Transformación de la zona accesible si la elección fue el vértice P_2.

Figura 3.4: Análisis del Juego del Caos, 2^a iteración.

Con el anterior análisis se enfatiza que hay regiones a las que no se puede acceder en cada iteración y estas son independientes de las elecciones del algoritmo. Este conjunto es la primera iteración de la construcción del Triángulo de Sierpinski que se mostró en el Ejemplo 2.2.2.



(a) Regiones accesibles en la 1a. iteración del Juego del Caos.

(b) Regiones accesibles en la 2a. iteración del Juego del Caos.

(c) Regiones accesibles en la 3a. iteración del Juego del Caos.

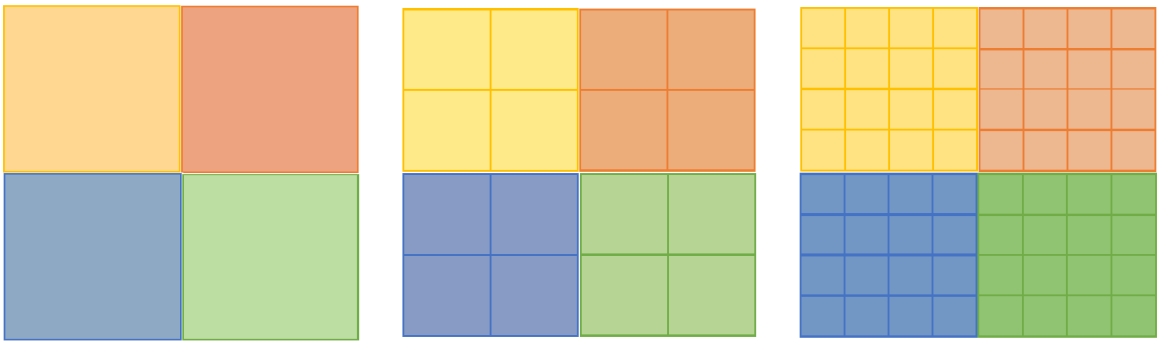
Figura 3.5: Regiones accesibles del Juego del Caos.

En la segunda iteración las regiones accesibles se muestran en la Figura 3.5b, la cual corresponde a la segunda iteración de la construcción del Triángulo de Sierpinski. Con este patrón ya se puede deducir que la enésima iteración corresponde al enésimo paso de la construcción del Triángulo de Sierpinski. Por esta razón, cualquier conjunto $\{x_1, \ldots, x_N\}$ obtenido con el Juego del Caos tiende al atractor del sistema: el Triángulo de Sierpinski. En otros palabras, la órbita de x_o llena 'densamente' al atractor, es decir, dado cualquier punto del triángulo de Sierpinski, habrá puntos de la órbita tan cerca como se quiera de este punto.

Hay que mencionar que existen algunas series de elecciones de vértices que no tienden al atractor. Por ejemplo, supóngase el caso donde siempre se elige el mismo vértice entonces lo que obtendría es una secuencia de puntos que se acerca tanto como se quiere al vértice que se elige. Sin embargo, la probabilidad de que esto pase es 0.

3.1.2. Variantes del Juego del Caos

Como ya se mencionó anteriormente, se puede ejecutar el Juego de Caos con distintos polígonos y se obtendrán conjuntos atractores diferentes, que también son fractales. Sin embargo, existen excepciones, una de ellas es el caso de un polígono de 4 lados. El conjunto de puntos $\{x_0, \ldots, x_N\}$ obtenido de la trayectoria de cualquier punto en el plano bajo el algoritmo, se distribuye equitativamente en el interior del cuadrado como se ve en la Figura. 3.8a.



(a) Regiones accesibles en la 1a. iteración.

(b) Regiones accesibles en la 2a. iteración.

(c) Regiones accesibles en la 3a. iteración.

Figura 3.6: Regiones accesibles del JC para $k = 4$.

Se puede justificar la forma de este peculiar atractor haciendo un análisis similar al que se hizo en la sección anterior. Se comienza visualizando el resultado de aplicar cada una de las transformaciones lineales posibles y se marcan las regiones accesibles en la primera

iteración. Recuérdese que geométricamente cada transformación contrae al cuadrado hacia cada vértice diferente como se muestra en la Figura 3.6a.

A diferencia del juego del Caos en un triángulo, aquí no hay hueco, es decir, no existe esa región a la que no se pueda acceder independientemente de la función seleccionada; por lo que, la unión de todas las posibles regiones accesibles es el cuadrado original y este comportamiento se repite para las siguientes iteraciones como se ve en las Figuras 3.6b y 3.6c. Por esta razón, la órbita de cualquier punto llena uniformemente el cuadrado pues todo cuadrado es el atractor de este sistema de funciones iteradas. No obstante, si se cambia la probabilidad de elección el cuadro ya no se llenaría de manera equitativa pues habría zonas con más probabilidad de acceso.

A pesar de ello, se puede modificar el Juego del Caos para $k = 4$ para obtener figuras fractales. Si se imponen restricciones en la elección de los vértices se pueden obtener fractales como se ve en la Figura 3.8. Si se numeran los vértices del cuadrado comenzando en la parte superior izquierda, comenzando desde cero y se sigue en sentido horario como se muestra en la Figura 3.7, se pueden imponer las siguientes restricciones:

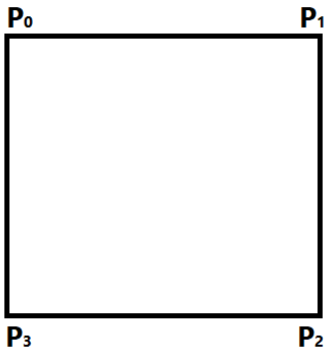
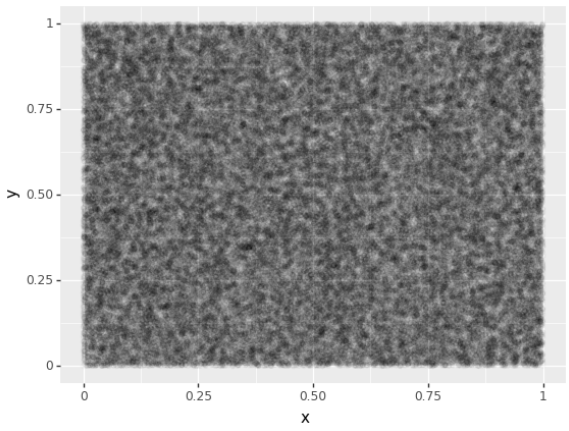
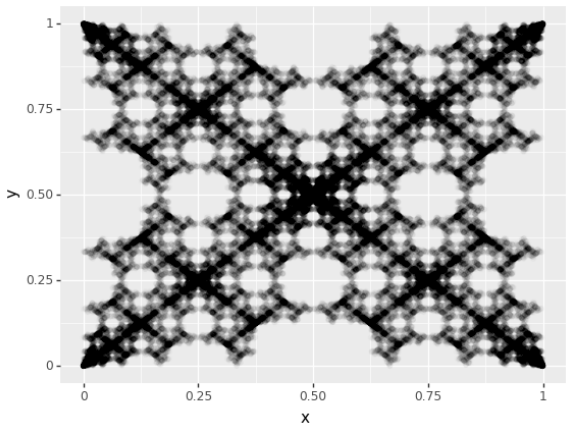


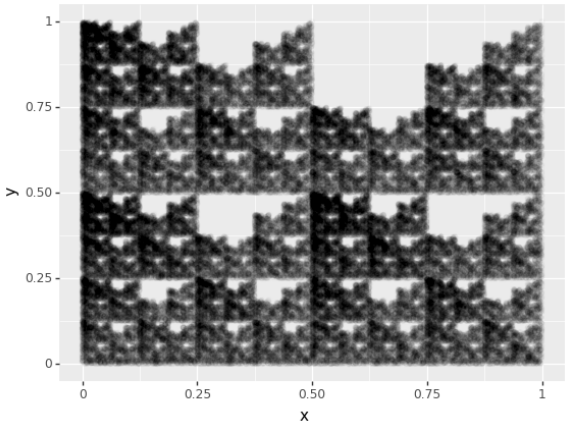
Figura 3.7: Enumeración del cuadrado.



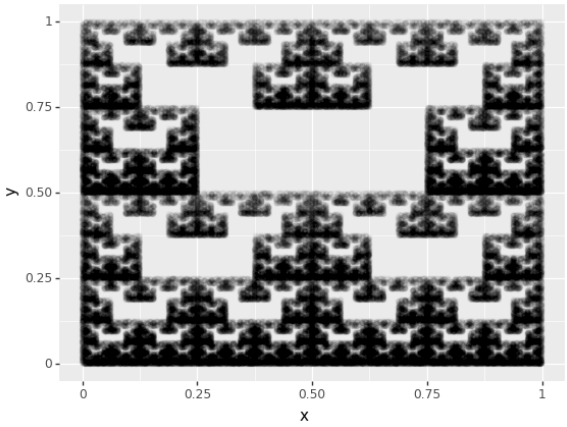
(a) Juego del Caos original.



(b) Restricción, Caso 1



(c) Restricción, Caso 2



(d) Restricción, Caso 3

Figura 3.8: Variantes.

- **Caso 1**. El vértice elegido no puede ser vecino del vértice elegido anteriormente. Si los dos vértices elegidos previamente son los mismos; en este caso se elige de forma equiprobable uno de los 2 vértices restantes. La Figura 3.8b es el resultado de esta restricción.

- **Caso 2**. El vértice elegido no puede ser P_1 si el vértice elegido previamente fue P_0 y se escoge de manera uniforme uno de los 3 vértices restantes. En la Figura 3.8c se muestra el Juego del Caos habitual con esta restricción.

- **Caso 3**. Finalmente, el vértice elegido aleatoriamente no puede ser el P_0 si el anterior vértice fue el P_2 y se escoge de forma uniforme uno de los 3 vértices restantes. Además, tampoco puede ser el vértice P_1 si el vértice previo fue el vértice P_3 y se elige de manera equiprobable entre los vértices restantes. La Figura 3.8d muestra la ejecución del algoritmo.

En los apartados E.2, E.3, y E.4 se puede ver el código para las anteriores restricciones respectivamente, escrito en el lenguaje de Python3.

Los ejemplos anteriores no son como los sistemas de funciones iteradas como con los que se había trabajado, ahora las elecciones no son independientes de las elecciones anteriores; en el Algoritmo 3.1.1 cada P_{i_n} es escogida de forma uniforme e independiente, es decir, para $(i_n, n \geq 1)$ i_n es una sucesión de variables aleatorias independientes e idénticamente distribuidas con una distribución uniforme discreta en $\{1, 2, 3, 4\}$; es decir,

$$\mathbb{P}[i_n = k] = \frac{1}{4} \quad \text{para} \quad k = 1, 2, 3, 4.$$

En cambio, en las variantes existe una dependencia con los estados previos, esto es, la distribución del estado i_n depende del estado i_{n-1} e incluso i_{n-2}. Las modificaciones en los casos anteriores se pueden ver como procesos estocásticos a tiempo discreto; en particular, el proceso de los Casos 2 y 3 forma una cadena de Markov.

Sea $E = \{0, 1, 2, 3\}$ el espacio de estados accesibles del proceso estocástico i_n con $n \geq 0$. Se tomarán los índices módulo 4 para simplificar la notación.

- **Caso 1.**

  Si $l \neq m$ entonces:

$$\begin{cases} \mathbb{P}[i_n = k | i_{n-1} = l, i_{n-2} = m] = \frac{1}{4} & \text{para } k = 0, 1, 2, 3 \\ \mathbb{P}[i_n = k | i_{n-1} = l, i_{n-2} = l] = 0 & \text{para } |k - l| \cong 1 \mod 4 \\ \mathbb{P}[i_n = k | i_{n-1} = l, i_{n-2} = l] = \frac{1}{2} & \text{para } |k - l| \ncong 1 \mod 4 \end{cases}$$

- **Caso 2.**

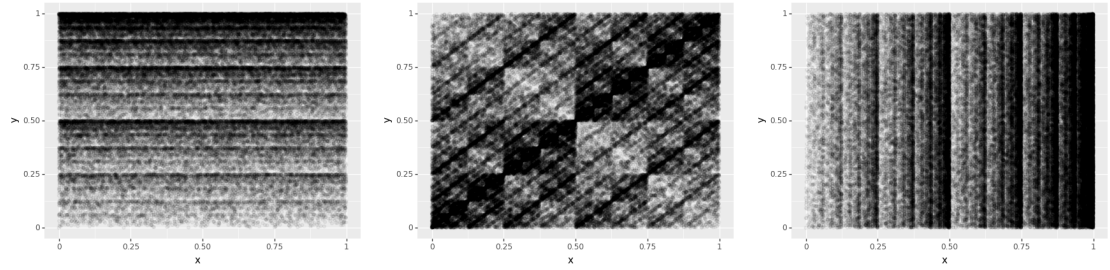
$$\mathbb{P}[i_n = k | i_{n-1} = l] = \begin{cases} \mathbb{P}[i_n = k | i_{n-1} = l] = \frac{1}{4} & \text{para } k = 0, 1, 2, 3 \quad l = 1, 2, 3 \\ \mathbb{P}[i_n = k | i_{n-1} = 0] = \frac{1}{3} & \text{para } k = 0, 2, 3 \\ \mathbb{P}[i_n = 1 | P_{i_{n-1}} = 0] = 0 \end{cases}$$

- **Caso 3.**

$$\mathbb{P}[i_n = k | i_{n-1} = l] = \begin{cases} \mathbb{P}[i_n = k | i_{n-1} = 1] = \frac{1}{4} & \text{para } k = 0, 1, 2, 3 \\ \mathbb{P}[i_n = k | i_{n-1} = 2] = \frac{1}{4} & \text{para } k = 0, 1, 2, 3 \\ \mathbb{P}[i_n = k | i_{n-1} = 2] = \frac{1}{3} & \text{para } k = 1, 2, 3 \\ \mathbb{P}[i_n = k | i_{n-1} = 3] = \frac{1}{3} & \text{para } k = 0, 2, 3 \\ \mathbb{P}[i_n = 0 | i_{n-1} = 2] = 0 \\ \mathbb{P}[i_n = 1 | i_{n-1} = 3] = 0 \end{cases}$$

Las formas fractales que se observan en las variantes del Juego del Caos son un reflejo de las correlaciones de la secuencias $(i_n)_{n \geq 1}$.

Cabe destacar que la distribución de probabilidades en este sistema de funciones iteradas es un factor decisivo para la formación del fractal. Si se toma una distribución distinta de la uniforme, pero sin que ninguno de los vértices tenga asignado la probabilidad nula, se obtendrá como resultado la formación de un fractal distinto entre cada distribución. Por ejemplo, tómese la numeración de los vértices del cuadrado como anteriormente se describió 3.7 y $p = (p_0, p_1, p_2, p_3)$ el vector de probabilidades correspondiente a cada vértice. En la Figura 3.9 se ilustra lo dicho anteriormente, se ejecuta el juego del Caos en cuadrado con distintos vectores de probabilidades.

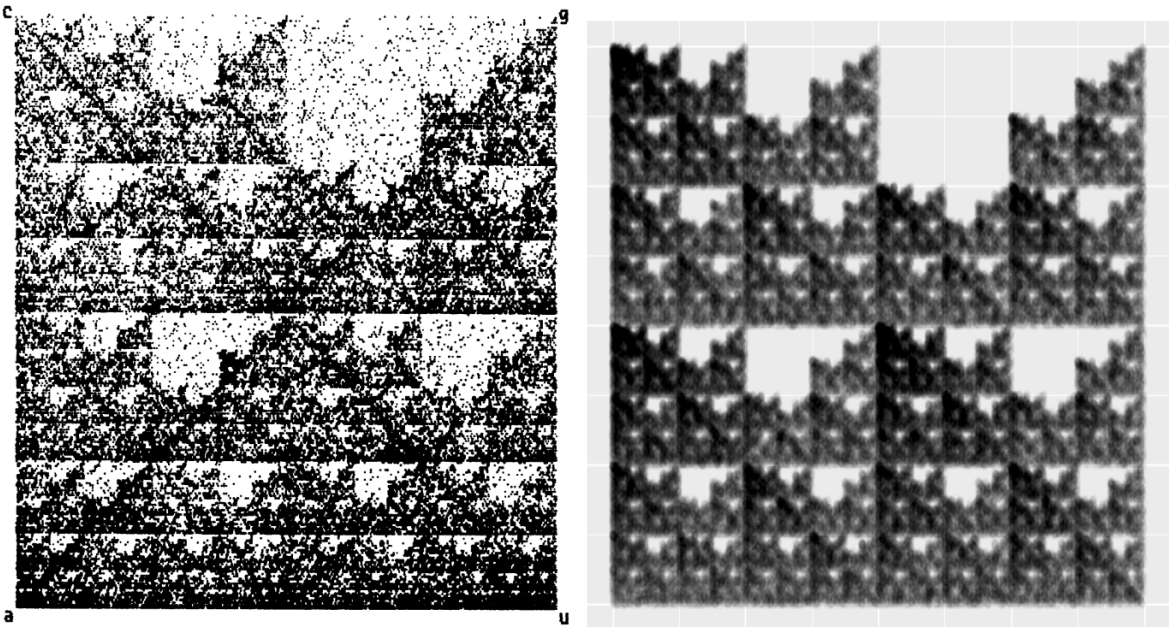


(a) $p = (1/3, 1/3, 1/6, 1/6)$ (b) $p = (1/6, 1/3, 1/6, 1/3)$ (c) $p = (1/6, 1/3, 1/3, 1/6)$

Figura 3.9: Juego del Caos con $k = 4$ y diferentes probabilidades.

3.2. Juego del Caos con ADN

En 1989 H. Joel Jeffrey publicó su artículo *'Chaos game representation of gene structure'* [27] en el que se muestra un nuevo método para representar patrones en el ADN usando sistemas de funciones iteradas; ya que, el genoma se puede ver como una cadena de letras A, C, T, G. Si se le asocia a cada vértice de un cuadrado una letra, entonces se puede ejecutar el Juego de Caos tomando como la sucesión aleatoria de vértices elegidos la secuencia del ADN. Esta subsección esta principalmente fundamentada en los artículos [27] y [28].



(a) Representación del gen HBB de [27] (b) Simulación del Juego del Caos, variante 2 (ver subsección 3.1.2).

Figura 3.10: Comparación del Juego del Caos ejecutado con datos provenientes del ADN y datos sintéticos.

En el artículo, Jeffrey aplicó el Juego del Caos a la región de la Beta Globina Humana en el cromosoma 11, el gen HBB con 73 357 bases; el resultado que obtuvo es lo que se muestra en la Figura 3.10a. Lo más destacable de esta representación, es el área casi vacía que se forma en el cuadrante superior derecho (el cuadrante G), esta extraña forma Jeffrey la bautizo como '*double-scoop*'. La forma de '*double-scoop*' se moldea debido a la escasez de guanina después de citosina, es decir, la subsecuencia CG es escasa en la secuencia del gen. Jeffrey también experimentó con otros genes: HUMALBGC *(human serum albumin gene, complete)* y HUMADAG *(human adenosine deaminase gene, complete)* y observó la misma estructura.

Otra aspecto a recalcar es que, esta '*double-scoop*' se parece mucho al resultado del Caso 2 de la subsección anterior en el se obtuvo la Figura 3.10b. De acuerdo a la correspondencia entre nucleótidos y vértices del cuadrado de la Figura 3.10b se construye la restricción de que no se puede elegir G si anteriormente el vértice elegido fue una C, lo cual concuerda con lo anteriormente expuesto. Al comparar las Figuras 3.10a y 3.10b se puede inferir que las densidades de estas figuras no son las mismas; en la simulación de la variante del Juego del Caos todo parece ser más definido y se observan zonas donde claramente no hay puntos. En la variante del Juego de Caos estrictamente se cumple que no se puede poner un 1 después de un 0 o equivalentemente usando nucleótidos no se puede poner una G después de una C, en cambio en la secuencia de ADN utilizada esto es solamente una tendencia. Si se observa dentro de la '*double-scoop*' se pueden encontrar puntos, esto quiere decir que la subsecuencia CG no es frecuente en esa región del genoma. Este hecho tiene sentido biológicamente pues en los vertebrados el dinucleótido CG tiende a desaparecer por la metilación[1] y suelen estar en gran cantidad en las zonas precedentes a las regiones codificantes que son las llamadas islas CpG; para más información véase [45].

Jeffrey no solo experimentó con secuencias del ADN provenientes de genoma humano, en su artículo se menciona que ejecutó el Juego del Caos con fragmentos de ADN provenientes de diversos animales vertebrados y en la mayoría obtuvo el mismo patrón de la '*double-scoop*'. Sin embargo, al realizar la representación con diferentes tipos de plantas no obtuvo ningún patrón característico; también experimentó con genes provenientes de bacterias, virus y fagos[2] obteniendo distintos patrones.

Al concluir el artículo, Jeffrey deja varias preguntas sin responder:

- ¿El patrón que se observa varía con el grupo de genes?

[1] La metilación es la adición de un grupo metilo ($-CH_3$) a una molécula. La metilación de los ácidos nucleicos puede afectar la forma en que estos actúan en el cuerpo.[10]

[2] Más conocidos como Bacteriófagos.

- ¿Podemos encontrar la representación matemática que represente los grupos de genes?
- ¿Podemos encontrar la representación matemática que represente las regiones no codificantes?
- ¿Se podría decir que el patrón observado en la Figura 3.10a es característica de los vertebrados?

Inspirado en las preguntas anteriores, este trabajo se enfoca a contestar la pregunta *¿El patrón que se observa en el genoma varía con respecto al de los genes?* .

Al tomar la secuencia de ADN como una serie de instrucción para la elección de los vértices del cuadrado, es decir, las primeras k-elecciones en el Juego de Caos corresponden a los primeros k nucleótidos; de esta manera se obtiene una forma de interpretar la estructura del ADN haciendo uso del algoritmo del Juego del Caos.

Por el análisis de las variantes del Juego del Caos se puede esperar lo siguiente:

- Si se logra distinguir algún patrón fractal, se puede descartar la idea de que el ADN es una secuencia de variables aleatorias independientes e uniformemente distribuidas.
- Los patrones que se obtengan con el Juego del Caos revelan información acerca de las correlaciones de los nucleótidos en la secuencia del ADN. Cada región de la figura revela una fina estructura de la secuencia utilizada.
- El patrón del Juego del Caos visible proveniente del algoritmo representa un patrón global y local. Al hacer la representación del genoma completo se tendrá una visualización completa del genoma, pero también estará ahí la visualización parcial de subsecuencias como la de los genes, que correspondería a tener imágenes encimadas.

Siendo más formales se precisa cómo estas caracterizaciones de la representación del Juego del Caos ocurren como consecuencia de las subsecuencias provenientes de fragmentos de distintos genomas [27].

Teorema 3.2.1. Existe un mapeo uno a uno entre la secuencia y el interior del cuadrado en la $k-$ésima iteración con la representación del Juego del Caos con una secuencia, que corresponde a la primera subsecuencia inicial de longitud k de la secuencia, y a ninguna otra subsecuencia (hasta la resolución de la pantalla). Por lo tanto, existe una correspondencia uno a uno entre las subsecuencias (ancladas al inicio) de una secuencia y los puntos de la representación del Juego del Caos [28, Pág. 27].

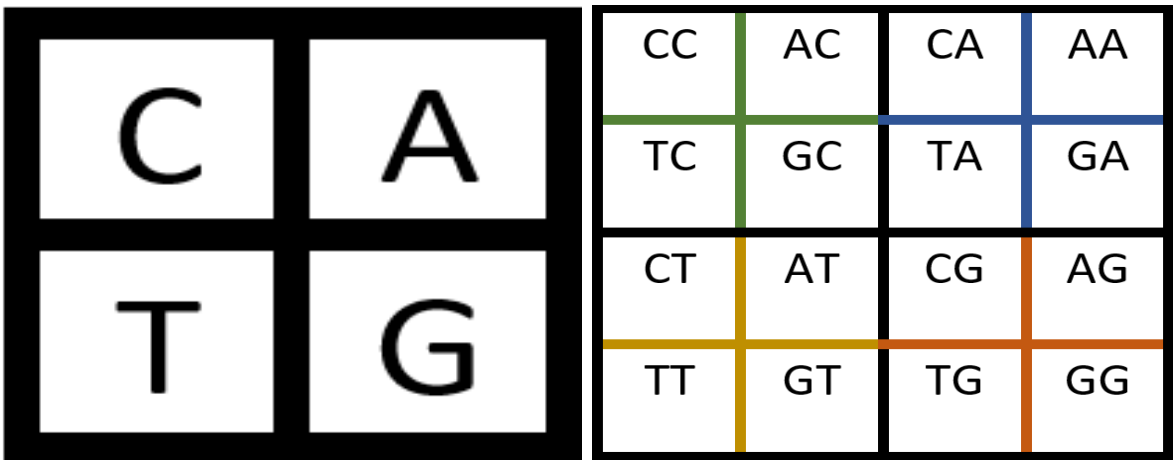
Dado un genoma de longitud N donde $N \in \mathbb{N}$, se va a tener una secuencia de puntos $\{x_0, \ldots, x_N\}$ que son el resultado de aplicar el Juego del Caos a un punto arbitrario x_0 del plano, siguiendo las elecciones según la secuencia del genoma. El punto x_l se asocia con la secuencia de los primeros l nucleótidos del genoma.

Para cada secuencia de puntos $x_0 \ldots x_N$ se puede asociar una secuencia de ADN de longitud N, se le llamará sufijo de longitud k a los últimos k nucleótidos de la secuencia de ADN, que corresponden a las últimas k instrucciones que se tomaron en el Juego del Caos.

Teorema 3.2.2. En la representación del Juego de Caos con un cuadrado de lado igual a 1, dos secuencias con sufijo de longitud k están contenidas dentro del cuadrado con lado de longitud 2^{-k}. [28]

En palabras sencillas, el Teorema 3.2.1 dice que dado un punto en el cuadrado se pueden deducir las anteriores k elecciones que se hicieron, en nuestro caso corresponden a los k nucleótidos finales de la secuencia del ADN.

Para ilustrar el Teorema 3.2.2 supóngase que el cuadro del SFI tiene longitud 1 y usando la misma correspondencia de vértices y nucleótidos que utilizó Jeffrey, se puede cuadricular este cuadro en cuadros de tamaño $\frac{1}{2} \times \frac{1}{2}$ como se ve en la Figura 3.11a, estos cuadrantes brindan información de cuál fue la última secuencia elegida, esto es fácil de deducir si se recuerda el análisis del Juego del Caos para $k = 3$ de la Sección 3.1.1.



(a) $k = 1$ (b) $k = 2$

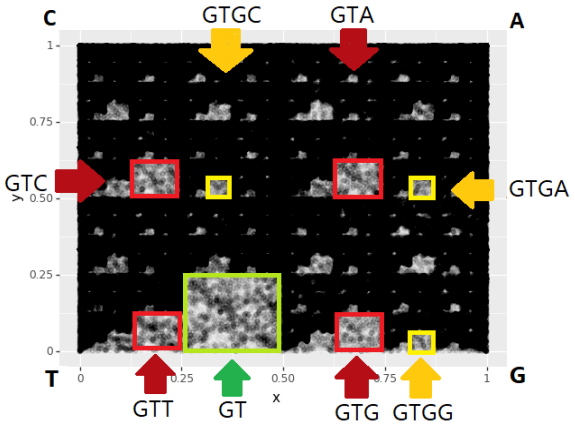
Figura 3.11: Cuadrículas del cuadrado.

Se puede obtener más información dividiendo en cuatro secciones cada cuadrante como se ve en la Figura 3.11b, así se obtiene una malla donde cada cuadro mide $\frac{1}{4} \times \frac{1}{4}$ y se obtiene información de la últimos dos nucleótidos. Si se quisiera una mayor precisión

en las últimas k-subsecuencias elegidas, se tendría que cuadricular el cuadro original en cuadros de tamaño 2^{-k}; para terminar de ilustrar esta idea en la Figura 3.12a se muestra la cuadrícula para $k = 3$ y las subsecuencias correspondientes a cada cuadro.

CCC	ACC	CAC	AAC	CCA	ACA	CAA	AAA
TCC	GCC	TAC	GAC	TCA	GCA	TAA	GAA
CTC	ATC	CGC	AGC	CTA	ATA	CGA	AGA
TTC	GTC	TGC	GGC	TTA	GTA	TGA	GGA
CCT	ACT	CAT	AAT	CCG	ACG	CAG	AAG
TCT	GCT	TAT	GAT	TCG	CG	TAG	GAG
CTT	ATT	CGT	AGT	CTG	ATG	CGG	AGG
TTT	GTT	TGT	GGT	TTG	GTG	TGG	GGG

(a) $k = 3$



(b) Obtención de subsecuencias.

Figura 3.12: Ejemplos de deducción de las bases anteriores.

En la Figura 3.12b se ilustra cómo se puede obtener la estructura que tienen las subsecuencias del ADN que caen en esa región, a partir de un cuadro de tamaño 2^{-k} para $k = 2, 3, 4$ que se señalan con los colores verde rojo y amarillo respectivamente. Es importante notar que si dos puntos se encuentran en el mismo cuadrante, entonces esos puntos tuvieron como última base los nucleótidos asociados al cuadrante; además, las bases nitrogenadas subsecuentes no necesariamente son graficadas en puntos cercanos en el cuadrado. Para ejemplificar esto, tómese la subsecuencia CC, el punto asociado a esta subsecuencia se va a encontrar en el cuadrante superior izquierdo y si la siguiente base es T entonces el punto subsecuente se va encontrar en la parte superior izquierda del cuadrante T, el cual es lejano al punto previo.

Por otro lado, dos puntos 'cercanos' pueden tener asociados o no secuencias parecidas, todo dependerá de la distancia que los separa, su posición en el cuadrado y qué tan fina sea la cuadrícula que se utilice. Por ejemplo, en la Figura 3.12a se puede apreciar que los cuadros asociados a las secuencias AAT y CCG son cercanos, pero las secuencias son totalmente distintas o también, si se toman los cuadros asociados a los secuencias TTT y GTT son cercanos y las secuencias sí son parecidas.

Otro aspecto razonable, sería considerar en la cuadrícula $J = 3$ y asignar un aminoácido a cada cuadrante para ver la densidad del cuadro y con esto poder concluir algo sobre la frecuencia de cada aminoácido en el genoma. Sin embargo no es tan sencillo, ya que en los subcuadrados se encuentran todos los puntos asociados a cada aminoácido, pero

no todo lo que se encuentre en un subcuadrado particular corresponde al aminoácido asociado a ese cuadrado. Por ejemplo, si se fija el cuadrante, tómese como modelo el cuadrante correspondiente a la subsecuencia ATG, se encontrarán aquellos puntos cuyas secuencias asociadas tienen como sufijo ATG; pero esto no indica necesariamente que sea un aminoácido, si se considera la subsecuencia $AAT-GCC$ que solo está formada por seis nucleótidos asociados a dos aminoácidos y se lee hasta el cuarto lugar, habrá un punto en el cuadrante ATG y, sin embargo, esta secuencia nunca tuvo el aminoácido asociado a la subsecuencia ATG.

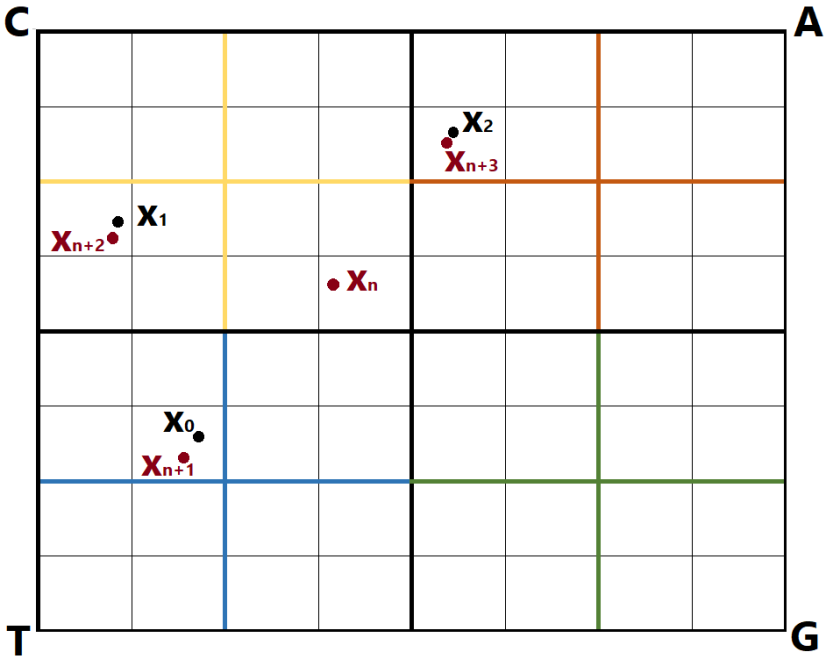


Figura 3.13: Dos puntos con distinto número de iteraciones y los sufijos.

Hay que tener en cuenta que, la afirmación contrapuesta del Teorema 3.2.2 no es cierta, es decir, si dos puntos x_n y x_m están dentro del cuadro de longitud 2^{-k}, no necesariamente tienen los mismos k sufijos. Por ejemplo, tómese la cuadrícula de $k = 3$, el cuadrante con sufijo TCA y la sucesión $\{x_i\}_{i=0}^{N}$, como en la Figura 3.13 se ilustra esta sucesión. Tómense dos puntos x_2 y x_{n+3}, al hacer el proceso de regresión se obtendría que los sufijos de ambos puntos son TCA; en caso de x_{n+3} es correcto, no obstante, en el caso x_2 no coincide porque solo habían pasado dos iteraciones; el sufijo correcto asociado a este punto es CA.

Por lo que, si se tienen dos puntos x_n y x_m, y $k > n$, entonces el sufijo de longitud k no existe. De modo que, la afirmación contrapuesta solo se cumple si $m, n \geq k$.

3.2.1. Representación del Polígono

Para comenzar el análisis del patrón que se observa en grupo de genes, se empieza haciendo lo mismo que Jeffrey. Se hará una representación visual con el Juego del Caos con fragmentos de ADN, pero con toda la secuencia de ADN perteneciente al genoma y con toda la secuencia perteneciente a las regiones codificante para así poder comparar su representación visual.

Primero se tiene que elegir cómo etiquetar el cuadrado, esto es, cómo asignar los nucleótidos a cada vértice. Realizando todos los posibles etiquetados del cuadro, se obtienen 3 etiquetados esencialmente distintos, pues no se pueden obtener como resultado de aplicarle alguna transformación lineal a alguno de ellos, es decir, no se puede obtener alguna de las tres representaciones girando o reflejando alguna de las otras dos. Hay que notar que la representación que Jeffrey utilizó es una reflexión con respecto a la identidad negativa del cuadro 3.

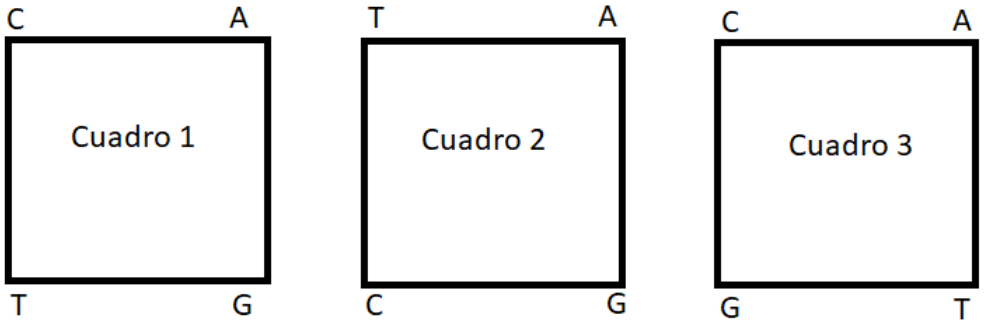


Figura 3.14: Cuadros con los que se trabaja.

3.2.2. Bases de Datos

El Centro Nacional para la Información Biotecnológica por sus siglas en inglés **NCBI** https://www.ncbi.nlm.nih.gov/, proporciona bases de datos, herramientas y otros medios para el análisis de medicina molecular; cuenta con las secuencias de ADN más actualizadas y completas. En total se tomaron para este estudio 30 genomas, los cuales son representantes de los 5 reinos en los que se dividen los seres vivos de acuerdo a la biología: Reino Monera, Reino Protista, Reino Fungi, Reino Animalia y Reino Plantae. Esta clasificación será útil al comparar los distintos cuadros obtenidos entre especies y así, si se encuentran semejanzas se podrán analizar con detalle; para más información sobre la clasificación de los seres vivos véase el Apéndice A.

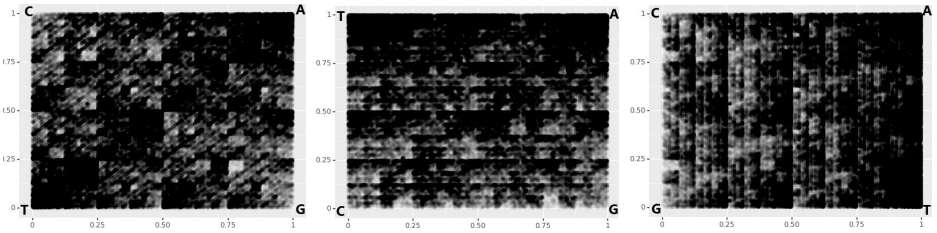
Para la representación gráfica de los genomas y las regiones codificantes, se utilizaron dos tipos de gráficas una a blanco y negro hecha con la librería Plotnine y la gráfica

de densidad a color con mpl-scatter-density. La gráfica a blanco y negro fue la primera representación, sin embargo, es muy tardado el proceso de creación de la imagen y no se logra distinguir algún patrón cuando el genoma es muy grande. Por otro lado, la gráfica a color se fija en la densidad por pixel; cuenta cuántos puntos x_n caen dentro de cada pixel y de acuerdo a este número se ilumina el pixel según la gama de colores que se le indique. El número máximo que puede tener un pixel está determinado por el número *largo de la secuencia* $\cdot$ 0.00004 siendo el valor mínimo del pixel cero; si hay pixeles donde no hay puntos, entonces los deja en blanco.

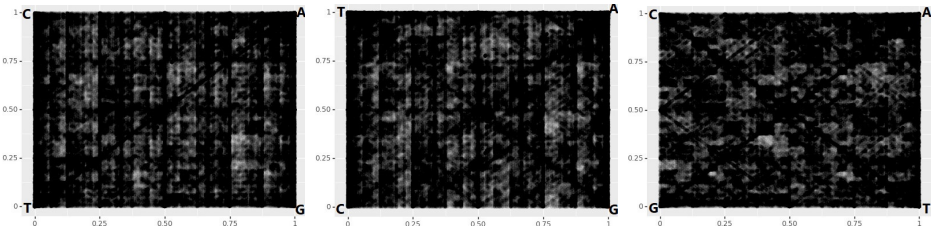
Todos los genomas, imágenes y archivos de texto con la información de cada especie, como el tamaño de genoma, porciones de regiones codificantes o los archivos de GenBank utilizados, se encuentra en la carpeta de drive https://drive.google.com/drive/folders/12USB3ZLfr-Le9aHopXYfbjz0WE_g7zPy?usp=sharing.

En la Figura 3.15 se pueden apreciar las representaciones gráficas de cinco genomas seleccionados. Los genomas parecen tener representaciones distintas, esto indicaría diferencias estructurales en los genomas, que es lo más razonable; en los apartados E.5 y E.6 se encuentra el código escrito en el lenguaje Python 3 que se utilizó para realizar las imágenes.

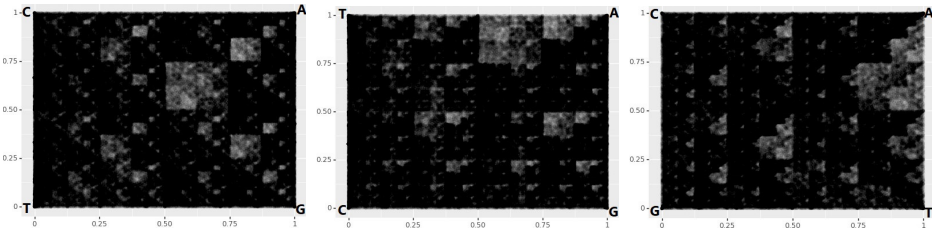
Los resultados, al ejecutar el Juego del Caos utilizando distintos genomas, son diferentes y se pueden observar claros patrones fractales. Cada imagen fractal depende del genoma que se seleccionó e indica que el genoma tiene algún patrón y cambia entre especies.



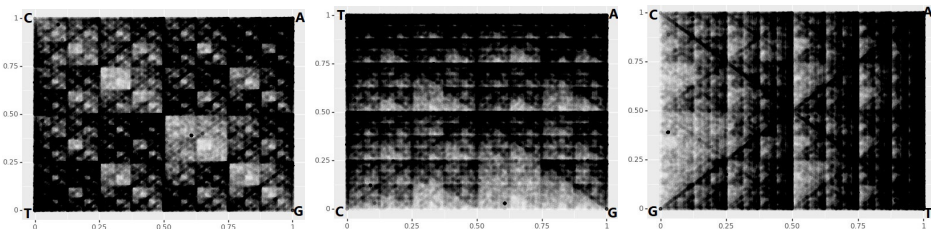
(a) Cuadro 1, bacteria (b) Cuadro 2, bacteria (c) Cuadro 3, bacteria



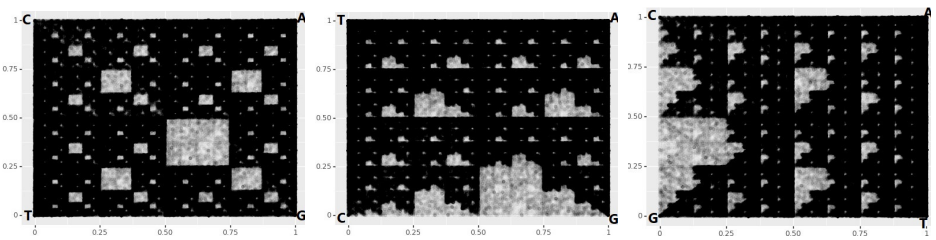
(d) Cuadro 1, protista (e) Cuadro 2, protista (f) Cuadro 3, protista



(g) Cuadro 1, hongo (h) Cuadro 2, hongo (i) Cuadro 3, hongo



(j) Cuadro 1, planta (k) Cuadro 2, planta (l) Cuadro 3, planta



(m) Cuadro 1, mamífero (n) Cuadro 2, mamífero (ñ) Cuadro 3, mamífero

Figura 3.15: Representaciones de los diferentes genomas

3.3. Patrones del Juego del Caos por Especie Representativa

Continuando con la respuesta a *¿El patrón que se observa en el genoma varía con respecto al de los genes?*, se desarrolla un análisis cualitativo detallado de los patrones obtenidos como resultado de aplicar el Juego de Caos a genes y genomas a un ejemplar representativo. Después se exponen los cuadros donde se encontrarán todas las figuras obtenidas de todos los distintos genomas y genes con los que se experimentó, junto con una tabla resumen de su información taxonómica relevante, las cuales están divididas por reinos.

Se suele medir la longitud de una molécula de ADN o ARN de acuerdo al número de pares de bases[3] (pb) para la longitud de las secuencias largas utilizando las siguientes medidas:

- kilobases (kb) equivale a 1000 pb.
- Megabase (Mb) equivale a 10^6 pb.
- Gigabase (Gb) equivale a 10^9 de pb.

3.3.1. Reino Monera

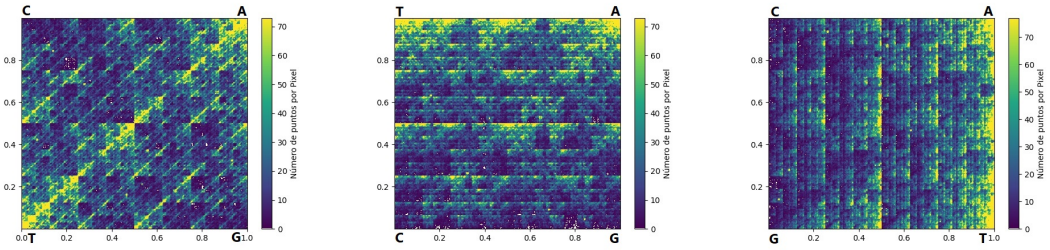
Se tomó como muestra el genoma de la bacteria *Haemophilus Influenzae*, del grupo *Haemophilus*, los cuales son parásitos que se encuentran en las membranas mucosas de los animales. El largo de su genoma es de 1.8477 mb. Las Figuras 3.16a, 3.16b, 3.16c corresponden a la representación del genoma y 3.16d, 3.16e, 3.16f a la representación de los genes de los cuadros de tipo 1, 2 y 3 respectivamente.

Las representaciones obtenidas tienen una similitud a las simulaciones de la Figura 3.9, con esto se podría deducir que hay una tendencia a la aparición de los nucleótidos A y T; en cambio, los nucleótidos C y G son menos frecuentes. Sin embargo, las Figuras 3.9 y 3.16 no son idénticas, lo que sugiere poner más detalle, es decir, observar las frecuencias de subsecuencias de longitud dos que corresponde a la malla con una partición de $\frac{1}{2^2}$. Los cuadros con mayor densidad corresponden a las subsecuencias AA, TT y en menor medida, pero aún significante, la subsecuencia AT y TA. Por otro lado, las subsecuencias CC y GG son escasas pero siguen presentes.

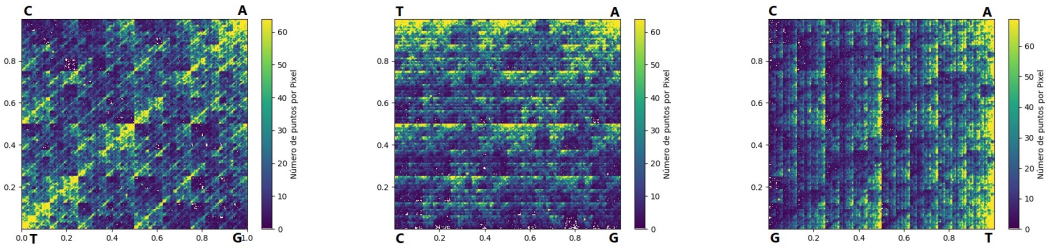
Visualmente no se aprecia una diferencia significativa con el cuadro de genomas. Hay que tomar en cuenta que, aproximadamente 88.6342 % del genoma son genes; por lo que, el 12.3658 % del genoma restante que corresponde a la región no codificante no tiene un

[3] Un par de bases mide alrededor de 3,4 Ángstrom.

patrón distintivo, pues con ella o sin ella, el patrón de los genes y del genoma se ven casi iguales, el atractor no se ve mejor definido.



(a) Cuadro 1, genoma. (b) Cuadro 2, genoma. (c) Cuadro 3, genoma.



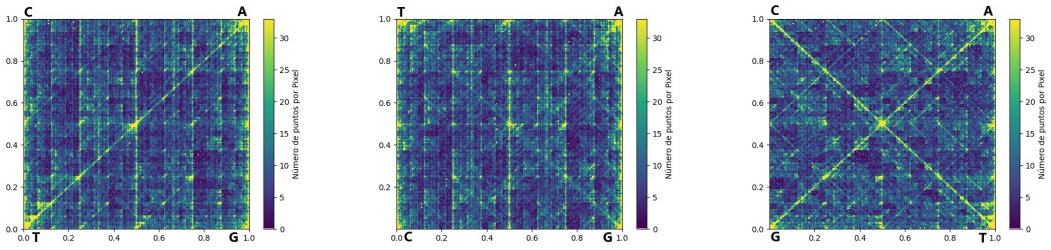
(d) Cuadro 1, genes. (e) Cuadro 2, genes. (f) Cuadro 3, genes.

Figura 3.16: Representación del genoma de *Haemophilus Influenzae*

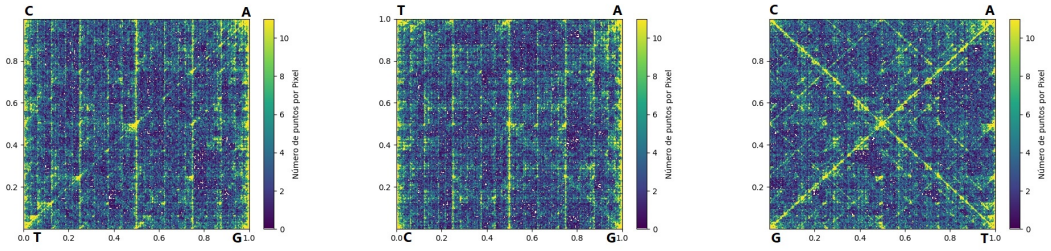
3.3.2. Reino Protista

Se tomó como ejemplar el genoma de *Plasmodium Vivax*, que es un parásito de mamíferos, como en el casa de los bonobos, gorilas o el ser humano. Se trata de la malaria más común. El largo de su genoma es de 29.1458 mb. Las Figuras 3.17a, 3.17b, 3.17c corresponden a la representación del genoma y 3.17d, 3.17e, 3.17f a la representación de los genes de los cuadros de tipo 1, 2 y 3 respectivamente.

En la representación obtenida del genoma se puede observar que la estructura del atractor no es una que se visualizara previamente. Este patrón muestra líneas verticales y transversales. No se puede divisar una tendencia en particular para la malla de $k = 1$, es decir, visualmente A, T, G, C parecen tener la misma frecuencia. Haciendo la malla más fina con $k = 2$, las subsecuencias más frecuentes corresponden a AA y TT. En el cuadro 3 3.17c, las diagonales que se marcan corresponden a una tendencia en los cuadros de las subsecuencias $CC, TC, AA, GA, AG, GG, CT, TT$.



(a) Cuadro 1, genoma (b) Cuadro 2, genoma (c) Cuadro 3, genoma



(d) Cuadro 1, genes. (e) Cuadro 2, genes. (f) Cuadro 3, genes.

Figura 3.17: Representación del genoma de *Plasmodium Vivax*.

Analizando las representaciones de los genes se puede ver un patrón más marcado en la diagonal del cuadro 3 de genes 3.17f en comparación al cuadro 3 del genoma 3.17c. Para la malla $k = 2$, las subsecuencias más frecuentes corresponden a CC, TC, AA, GA, AG, GG, CT, TT.

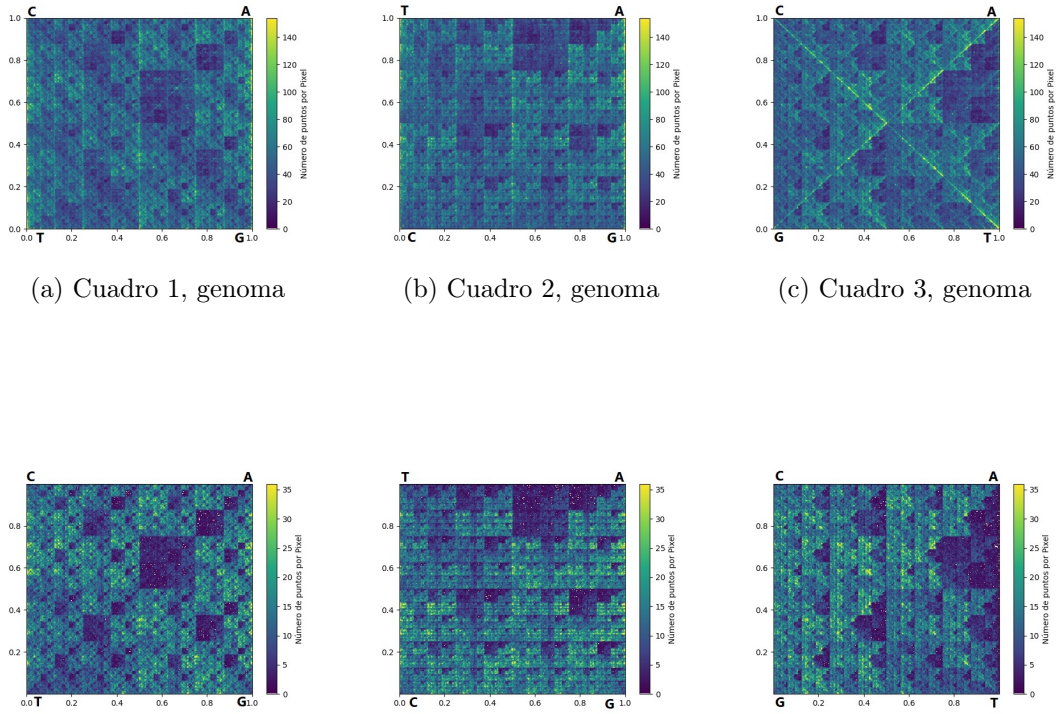
Comparando ambas representaciones, se infiere que sí se obtuvieron patrones ligeramente distintos, ya que los cuadros resultantes de las regiones codificantes se pueden apreciar más la estructura de rombos y cuadros; por lo que, se podría decir que ambos tienen tendencias distintas. La densidad de la región codificante en esta especie es del 35.2030 %. Este porcentaje es lo suficiente para caracterizar esta región.

3.3.3. Reino Fungi

Para este reino se usó el genoma del ejemplar *Penicillium Oxalicum*. Este hongo crece entre los granos y tiene un color azul-verdoso. Su genoma tiene un largo de 30.6551 Mb. La representación del genoma y genes de los cuadros 1, 2 y 3 corresponden a las imágenes de la Figura 3.18.

En las Figuras 3.18b y 3.18c, que corresponden a la representación de los cuadros 2 y 3 de genes, se puede observar la estructura de *double-scoop* pero no tan marcada, lo que indica

que la secuencia TA no es muy frecuente, pero sí suele aparecer. Otro aspecto relevante son las diagonales marcadas de la Figura 3.18c.



(a) Cuadro 1, genoma (b) Cuadro 2, genoma (c) Cuadro 3, genoma

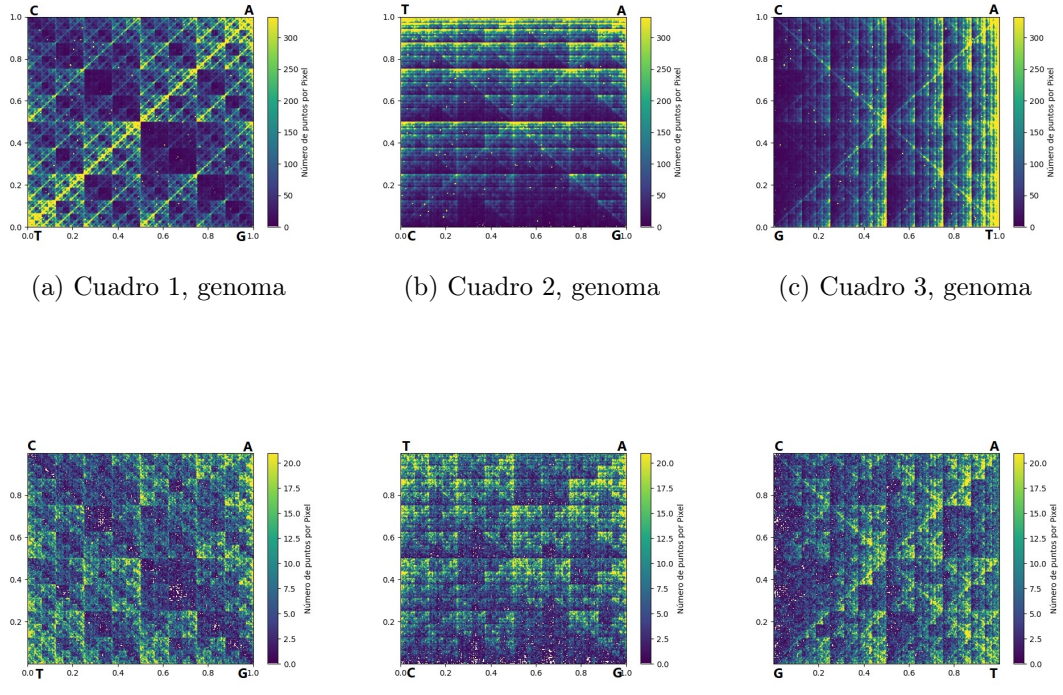
(d) Cuadro 1, genes. (e) Cuadro 2, genes. (f) Cuadro 3, genes.

Figura 3.18: Representación del genoma de *Penicillium Oxalicum*.

Por otro lado, en las imágenes provenientes de las regiones codificantes se puede inferir la forma de *double-scoop* más claramente y corresponde a las pocas apariciones de la subsecuencia TA, pero sin desaparecer. Se puede inferir también, que el resultado obtenido de las figuras de los genomas y genes, el patrón no cambia mucho; en el genoma, este patrón no es tan claro como en la secuencia correspondiente a los genes y tiene más detalles, pues se forman las diagonales en el cuadro 3 de genomas. Las regiones codificantes corresponden al 23.4975 % del genoma.

3.3.4. Reino Plantae

La especie representante elegida es *Cucumis Melo*, coloquialmente llamado melón. Esta planta es 90 % agua. El largo de este genoma es de 366.171 Mb. La representación del genoma corresponden a las Figuras 3.19a, 3.19b, 3.19c y las Figuras 3.19d, 3.19e, 3.19f corresponden la representación de los genes de los cuadros de tipo 1, 2 y 3 respectivamente.



(a) Cuadro 1, genoma (b) Cuadro 2, genoma (c) Cuadro 3, genoma

(d) Cuadro 1, genes. (e) Cuadro 2, genes. (f) Cuadro 3, genes.

Figura 3.19: Representación del genoma de *Cucumis Melo*.

La representación con el Juego del Caos del genoma vuelve a tener un ligero parecido con la Figura 3.9; por lo que, aplicando el mismo razonamiento que se hizo con las anteriores especies donde se observó este patrón, se puede deducir que, en el genoma de *Cucumis Melo* hay una carencia de nucleótidos C y G, en cambio, los nucleótido T y A son más frecuentes. No obstante, al igual que las anteriores, las figuras no son idénticas, ya que en estas representaciones se observan líneas transversales en los cuadros 3.19b y 3.19c.

Al observar las frecuencias de las subsecuencias de longitud 2, que corresponden a la malla con una partición de $\frac{1}{2^2}$, se puede discernir que los cuadros con menor densidad corresponden a las subsecuencias CG, pero no es lo suficientemente escasa para formar la *double-scoop*; las subsecuencias con mayor densidad corresponden a AA, TA, TT y AT.

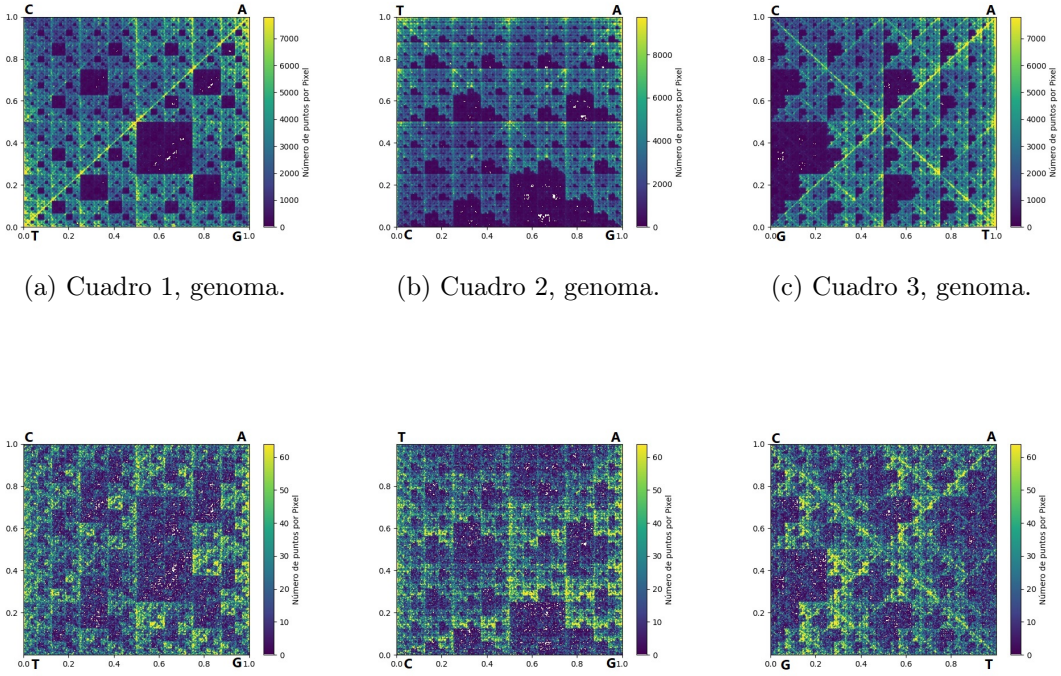
En la representación de las regiones codificantes, se puede ver que la frecuencia de los nucleótidos C y G es menor, mientras que los nucleótidos A y T son más frecuentes; utilizando una malla más delgada $k = 2$ se obtiene que las subsecuencias menos frecuentes son CG y TA; sin embargo, esta última subsecuencia suele ser más frecuente que CG.

En esta especie se puede apreciar una clara diferencia entre el atractor perteneciente a la secuencia del genoma y el de la región codificante. En este caso aproximadamente 6.5118 % del genoma pertenece a regiones codificantes, por lo que el 93.4882 % del restante genoma, que corresponde a la región no codificante, es lo suficientemente grande para tener

un patrón distinto.

3.3.5. Reino Animal

Se tomó como representante de este reino al genoma de *Mus Musculus* comúnmente conocido como ratón doméstico. Este es el mamífero más utilizado en experimentos de laboratorio. El largo de su genoma corresponde a 2.5 Gb. Las Figuras 3.20a, 3.20b, 3.20c corresponden a la representación del genoma y 3.20d, 3.20e, 3.20f a la representación de los genes de los cuadros de tipo $1, 2$ y 3 respectivamente.



(a) Cuadro 1, genoma. (b) Cuadro 2, genoma. (c) Cuadro 3, genoma.

(d) Cuadro 1, genes. (e) Cuadro 2, genes. (f) Cuadro 3, genes.

Figura 3.20: Representación del genoma de *Mus Musculus*

La representación del Juego del Caos del genoma, cuadro 2 de la Figura 3.20b, se parece mucho al caso 2 de las variantes del juego del Caos 3.8c, por lo que se puede deducir fácilmente cuál es el patrón que se sigue. En este genoma la subsecuencia CG es escasa, pero no es nula, ya que sí aparecen puntos azules dentro de la zona donde no debería haber algo. Como se mencionó anteriormente, esta característica tiene sentido por las islas CpG y por tratarse del genoma de un mamífero [45].

Con esta especie se observa una diferencia cualitativa entre los cuadros del genoma con respecto a los cuadros de las regiones codificantes. El cuadro 1 de genes 3.20d tiene un parecido con el caso 3 de la variante 3.8d del juego del Caos, por lo que las subsecuencias

que tienden a no aparecer son CG y TA, volviendo a hacer énfasis que no son nulas. El porcentaje de material genético codificante es aproximadamente 0.8227 % del genoma, este pequeño porcentaje tiene un patrón característico que es distinto al del genoma: la restricción del genoma más otra condición.

3.4. Patrones del Juego del Caos por Reinos

En esta sección se presentan de manera compacta, las distintas representaciones del Juego del Caos para 6 especies de cada reino; su taxonomía está resumida en un cuadro final de cada subsección.

De manera general, se observan las siguientes características del reino Monera. Teniendo presente que no existe el ADN intergénico en bacterias; además, que la gran mayoría del genoma es material genético codificante (alrededor del 90 %), se puede resaltar que, en la representación del genoma con el Juego del Caos de las seis especies provenientes de los géneros: *Staphylococcus*, *Haemophilus*, *Bifidobacterium*, no se logra discernir una diferencia significativa entre la representación del genoma completo y las regiones codificantes. Un factor influyente es la alta porción de material codificante en el genoma.

El resultado obtenido sugiere que la región no codificante tiene el mismo patrón que la región codificante o que sus diferencias no son significativas. Por otro lado, al comparar los tipos de cuadros entre todas las especies, se pueden destacar dos aspectos. El primero es que la representación del genoma y de las regiones codificantes son muy semejantes si se comparan cuadros provenientes de especies que tienen el mismo género. El segundo, es que los cuadros de los géneros *Staphylococcus* y *Haemophilus* se 'parecen' ligeramente pues tienen la diagonal izquierda muy marcada. En cuanto al género *Bifidobacterium* parece tener un patrón donde la diagonal izquierda es la que menos se marca.

En cuanto al reino Protista, se puede destacar lo siguiente al comparar todas las figuras obtenidas de los géneros *Giardia, Plasmodium, Trypanosoma* y *Leishmani*. Visualmente el cambio entre la representación de las regiones codificantes y genomas no es muy drástico; en los cuadros pertenecientes a las regiones codificantes parece acentuarse ligeramente más el patrón que se observa en los cuadros pertenecientes a genomas, lo que podría indicar que la región no codificante tiene un patrón distinto. En las especies seleccionadas, el porcentaje del genoma perteneciente al material genético es muy variable. El más grande tuvo el 89.5919 % y el menor 35.2030 %; sin embargo, entre más alta sea la densidad de las regiones codificantes el cambio resultante es menor y como consecuencia el cambio más notorio lo tiene *Plasmodium Vivax* donde se ve un patrón más definido parecido al de la Figura 3.9.

Analizando los resultados obtenidos entre especies se puede destacar lo siguiente:

- De las especies *Plasmodium Vivax* y *Plasmodium Ovale* los patrones obtenidos son diferentes entre ellos, en comparación al reino anterior, donde las especies con el mismo género presentaban patrones casi idénticos e indistinguibles a simple vista.
- Los patrones observados en las especies *Trypanosoma Brucei* y *Leishmania Donovani* tienen una diferencia que es significativa, aun cuando ambos pertenecen a la familia *Trypanosomatidae*.
- *Giardia Intestinalis* y *Giardia Muris* presentan patrones bastante similares, tanto en las figuras pertenecientes al genoma como a las regiones codificantes.

En tanto que, en el reino Fungi se observó que en este, las figuras obtenidas de los genomas y las regiones codificantes son ligeramente distintas. Las figuras pertenecientes a las regiones codificantes presentan el mismo patrón que el visto en su respectivo genoma, pero más marcado; lo cual sugeriría que la región no codificante tiene un patrón distinto. El porcentaje de regiones codificantes en el genoma es bajo en todas las especies, alrededor del 20 %. De los tres géneros, *Amanita, Penicillium* y *Trametes*, los tres pertenecen a clases distintas; sin embargo, únicamente los cuadros obtenidos de *Amanita* y *Penicillium* son muy parecidos, en cambio *Trametes* se le parece ligeramente, pues el patrón que se ve en sus figuras es más marcado.

Por otro lado, en el reino Plantae la diferencia en la representación del genoma y las regiones codificantes es muy notoria. Las regiones codificantes tienen un patrón completamente diferente, lo cual indica que la estructura de los genes es muy distinta a la del genoma. En estos ejemplares alrededor del 8 % del genoma pertenece a regiones codificantes.

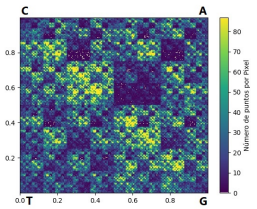
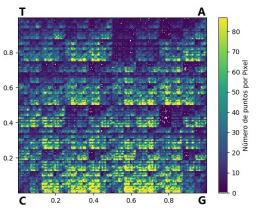
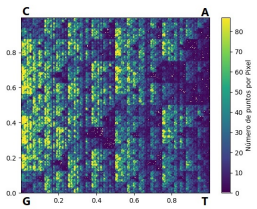
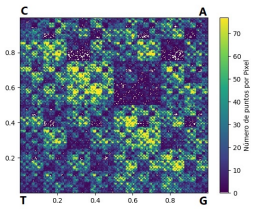
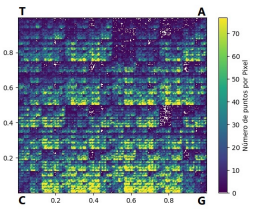
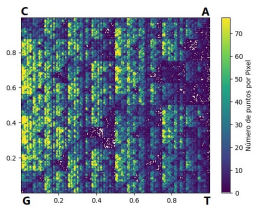
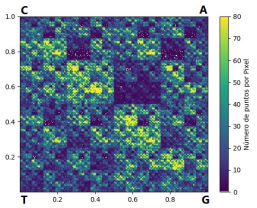
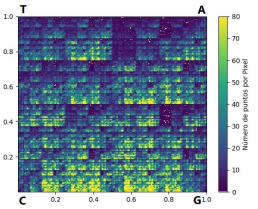
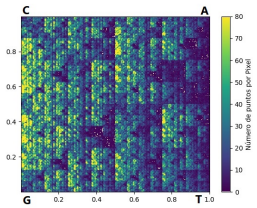
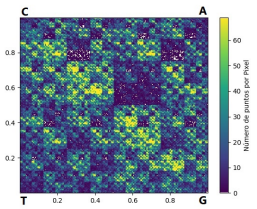
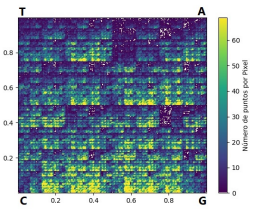
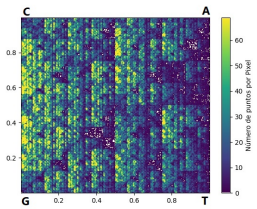
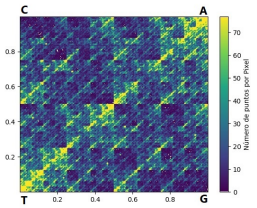
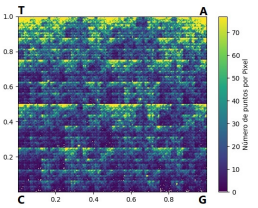
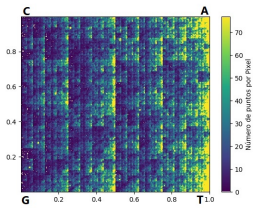
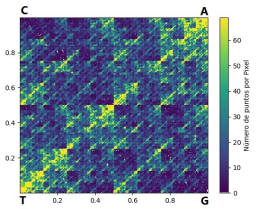
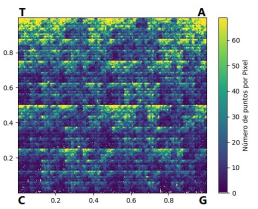
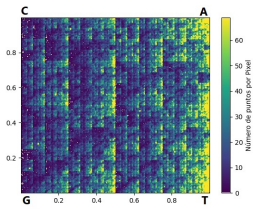
De los géneros *Oryza, Panicum, Brassica* y *Cucumis* con los que se trabajó, *Brassica* y *Cucumis* pertenecen a la misma clase; al hacer una comparación visual de los cuadros producidos de las especies pertenecientes a estos dos géneros, se puede observar cierto parecido en los cuadros del genoma; sin embargo, no son tan similares en los cuadros correspondientes a las regiones codificantes. En cambio, los géneros *Oryza* y *Panicum* pertenecen a la misma familia, la *Poaceae* y, por lo tanto, a la misma clase. Al realizar el contraste de las figuras pertenecientes a las especies de esta familia, se puede enfatizar que tiene una similitud visual, tanto en los cuadros correspondientes a las regiones codificantes como a los de los genomas.

Finalmente, el resultado obtenido en el reino Animalia, es muy similar a los últimos tres reinos, la representación del genoma y las regiones codificantes es diferente, pero es más notoria en el género *Mus*; por otro lado, en el género *Apis* no es tan notoria por la baja

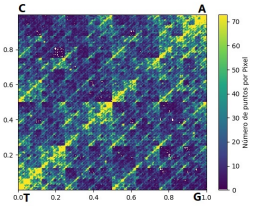
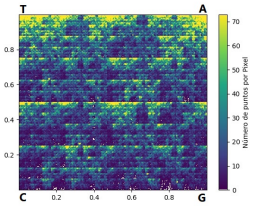
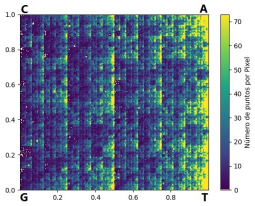
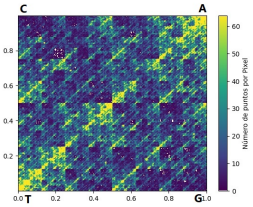
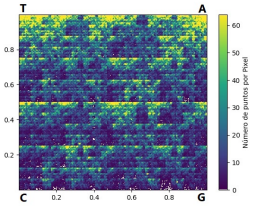
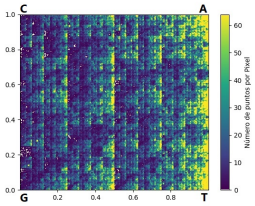
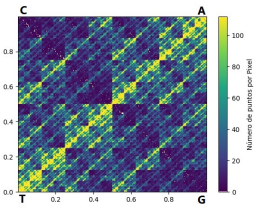
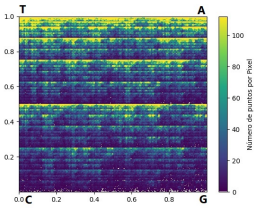
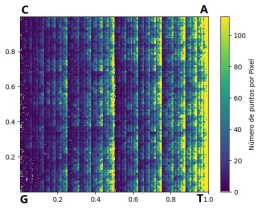
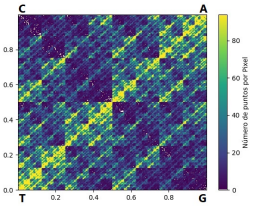
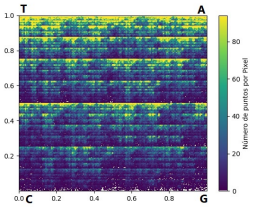
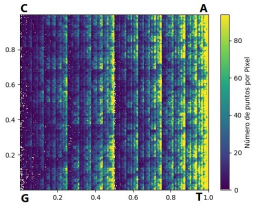
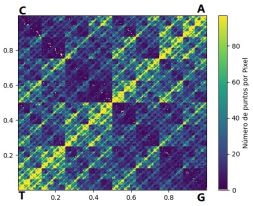
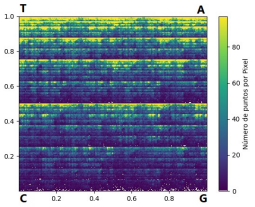
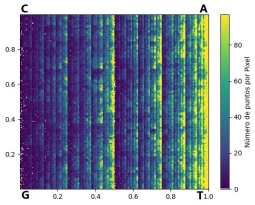
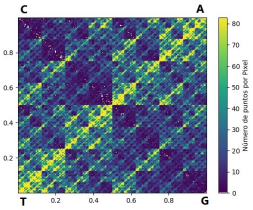
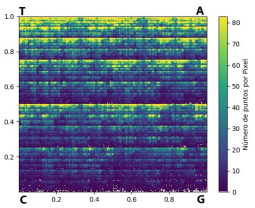
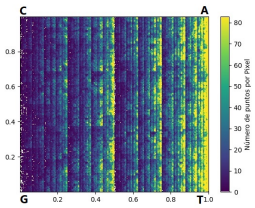
cantidad de información que había de regiones codificantes; aproximadamente el 3 % del genoma corresponde a regiones codificantes.

A contención se muestran las representaciones gráficas de todos las especies tomadas de referencia por reino.

3.4.1. Monera

Especie		Cuadro 1	Cuadro 2	Cuadro 3
Bifidobacteria Adolescentis	Genoma			
	Genes			
Bifidobacteria Angulatum	Genoma			
	Genes			
Haemophilus Haemolyticus	Genoma			
	Genes			

Cuadro 3.1: Representaciones de genomas y genes de bacterias 1.

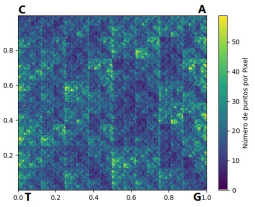
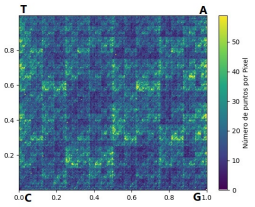
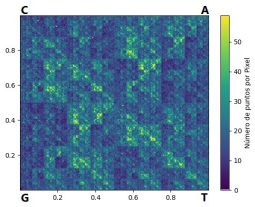
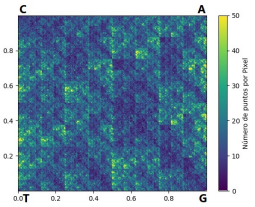
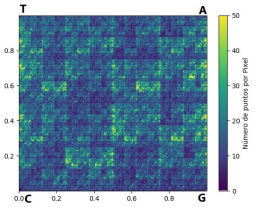
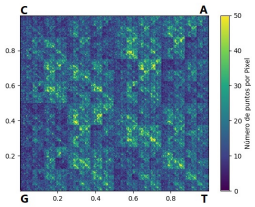
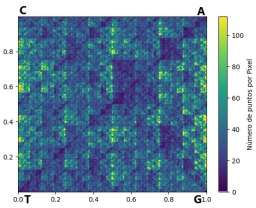
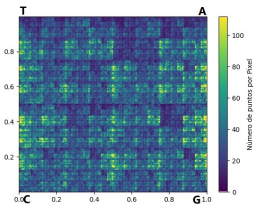
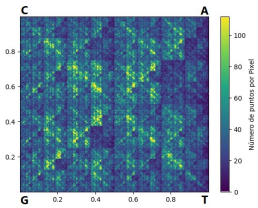
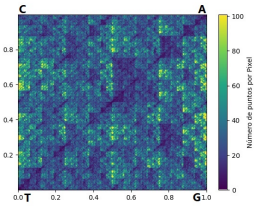
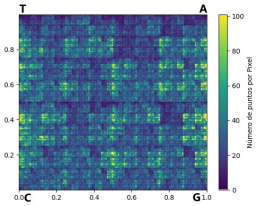
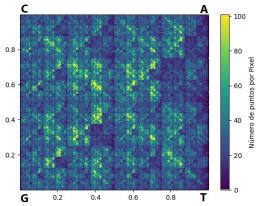
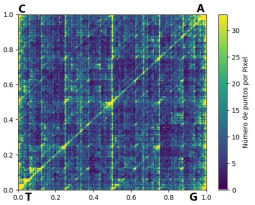
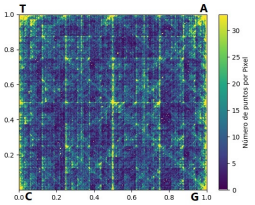
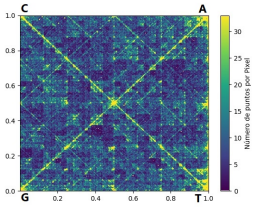
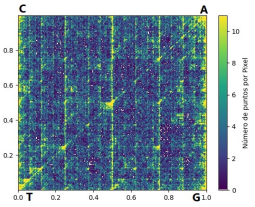
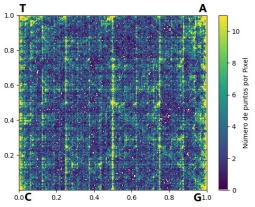
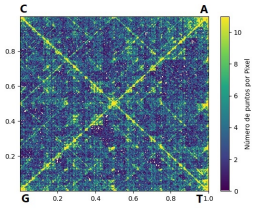
Especie		Cuadro 1	Cuadro 2	Cuadro 3
Haemophilus Influenzae	Genoma			
	Genes			
Staphylococcus Aureus	Genoma			
	Genes			
Staphylococcus Capitis	Genoma			
	Genes			

Cuadro 3.2: Representaciones de genomas y genes de bacterias 2.

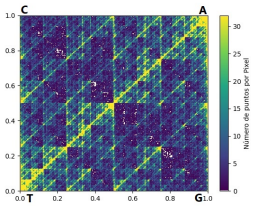
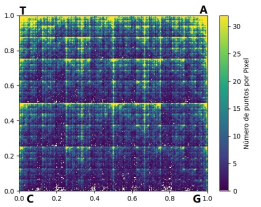
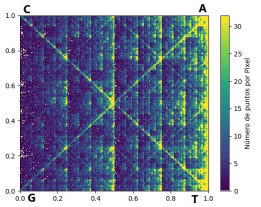
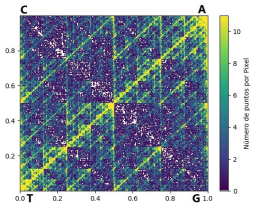
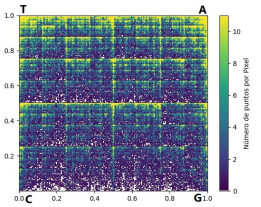
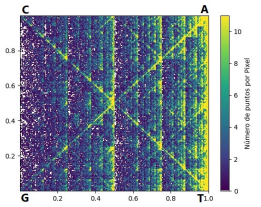
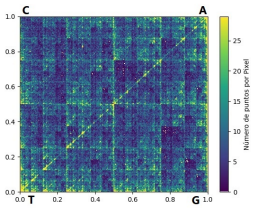
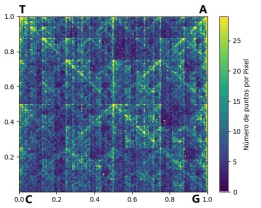
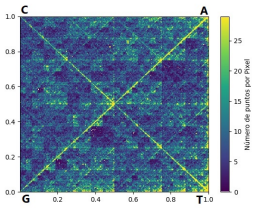
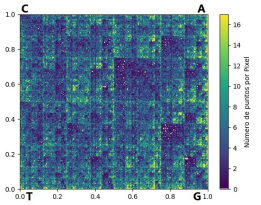
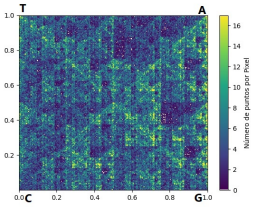
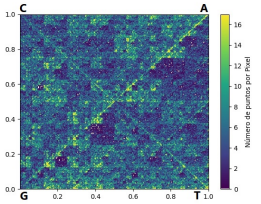
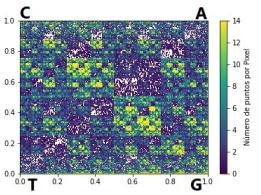
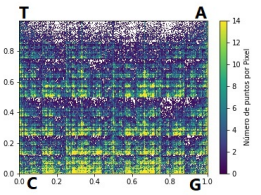
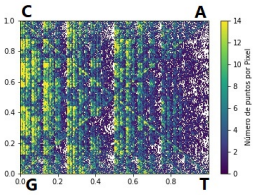
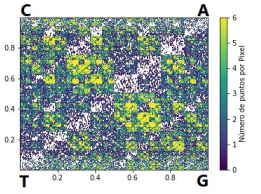
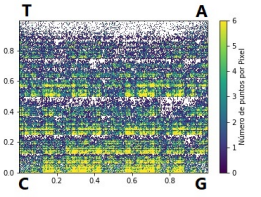
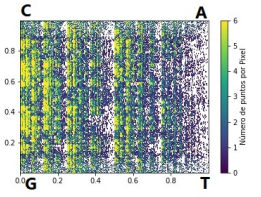
Información de Especies						
Dominio	Bacterias					
Clase	Actinobacteria		Gammaproteobacteria		Bacilli	
Orden	Bifidobacteriales		Pasteurellales		Bacillales	
Familia	Bifidobacteriaceae		Pasteurellaceae		Staphylococcaceae	
Género	*Bifidobacterium*		*Haemophilus*		*Staphylococcus*	
Especie	*Bif. Adolescentis*	*Bif. Angulatum*	*Haem. Haemolyticus*	*Haem. Influenzae*	*Staph. Aureus*	*Staph. Capitis*
% de Genes	88.0456 %	85.6784 %	89.2602 %	88.6342 %	83.3673 %	85.7066

Cuadro 3.3: Tabla con la taxonomía de las especies utilizadas del reino Monera.

3.4.2. Protistas

Especie		Cuadro 1	Cuadro 2	Cuadro 3
Giardia Intestinalis	Genoma			
	Genes			
Giardia Muris	Genoma			
	Genes			
Plasmodium Vivax	Genoma			
	Genes			

Cuadro 3.4: Representaciones de genomas y genes de protistas 1.

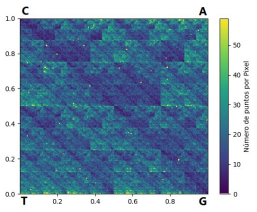
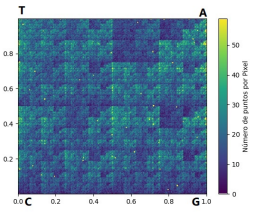
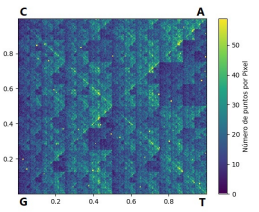
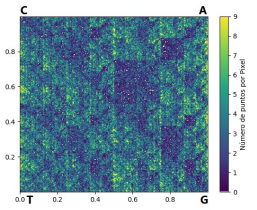
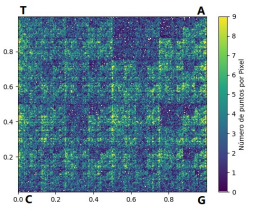
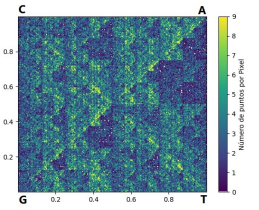
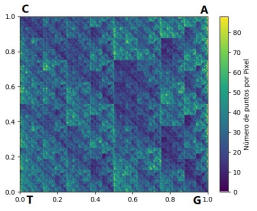
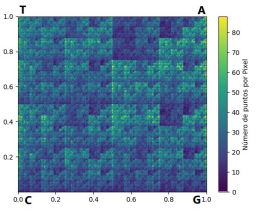
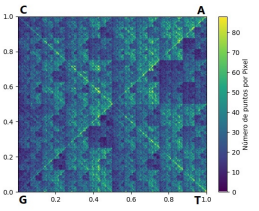
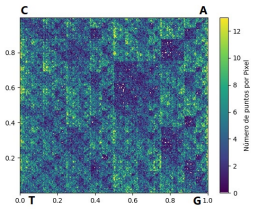
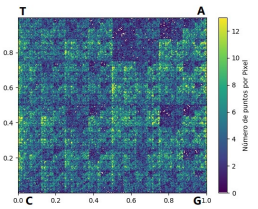
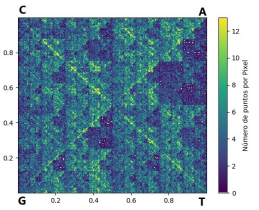
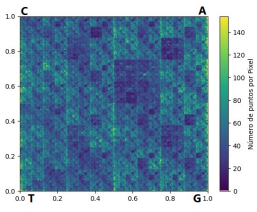
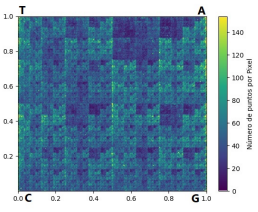
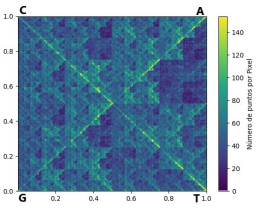
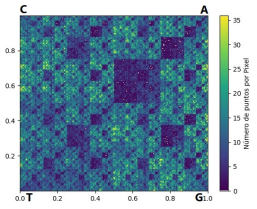
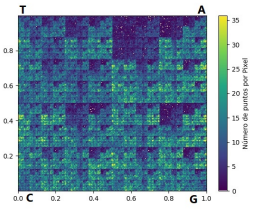
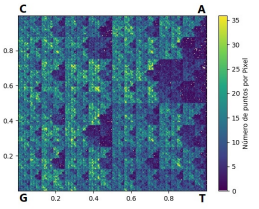
Especie		Cuadro 1	Cuadro 2	Cuadro 3
Plasmodium Ovale	Genoma			
	Genes			
Trypanosoma Brucei	Genoma			
	Genes			
Leishmania Donovani	Genoma			
	Genes			

Cuadro 3.5: Representaciones de genomas y genes de protistas 2.

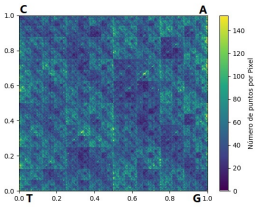
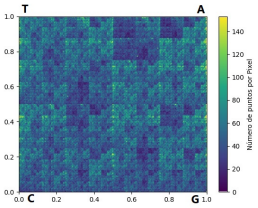
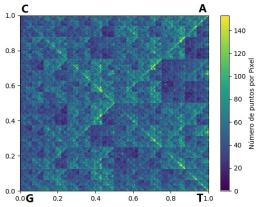
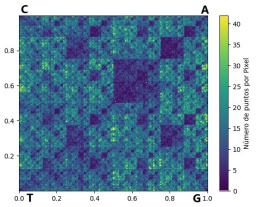
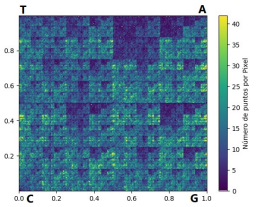
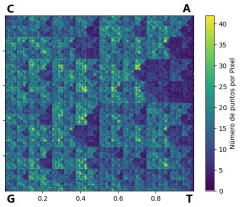
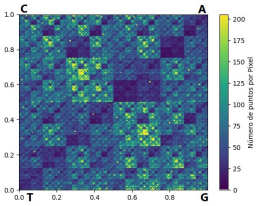
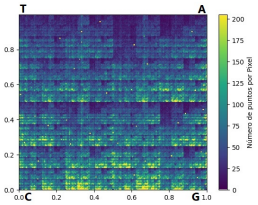
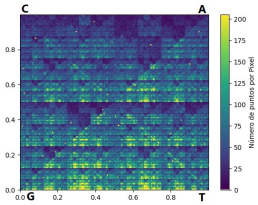
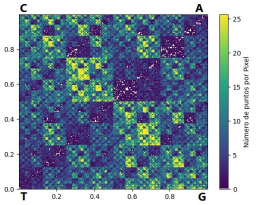
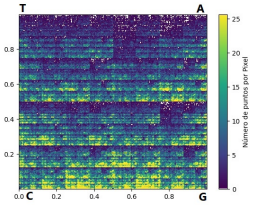
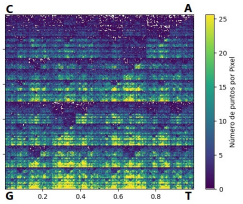
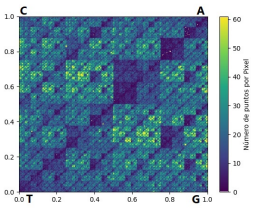
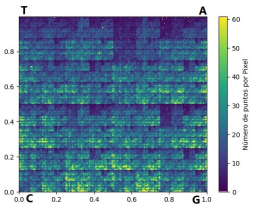
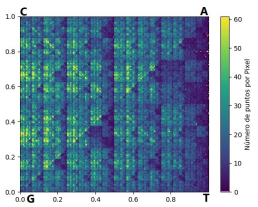
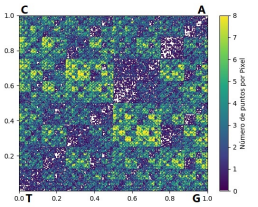
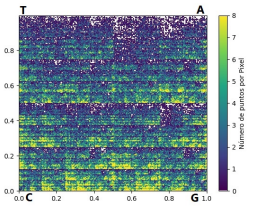
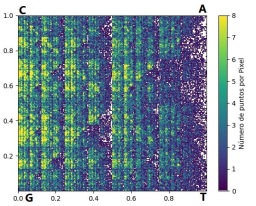
Información de Especies						
Dominio	Protista					
Clase	Zoomastigophorea		Aconoidasida		Kinetoplastea	
Orden	Diplomonadida		Haemosporida		Kinetoplastida	
Familia	Hexamitidae		Plasmodiidae		Trypanosomatidae	
Género	*Giardia*		*Plasmodium*		*Trypanosoma*	*Leishmania*
Especie	*Giardia Intestinalis*	*Giardia Muris*	*Plasm. Vivax*	*Plasm. Ovale*	*Tryp. Brucei*	*Leis. Donovani*
% de Genes	84.1668 %	89.5919 %	35.2030 %	35.5872 %	57.7022 %	48.6741 %

Cuadro 3.6: Tabla con la taxonomía de las especies utilizadas del reino Protista.

3.4.3. Hongos

Especie		Cuadro 1	Cuadro 2	Cuadro 3
Amanita Muscaria	Genoma			
	Genes			
Amanita Inopinata	Genoma			
	Genes			
Penicillium Oxalicum	Genoma			
	Genes			

Cuadro 3.7: Representaciones de genomas y genes de hongos 1.

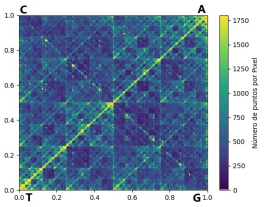
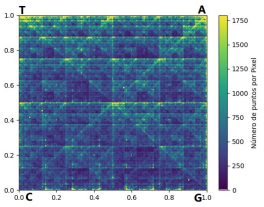
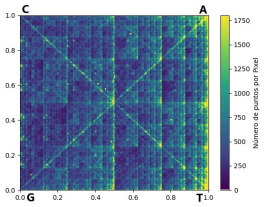
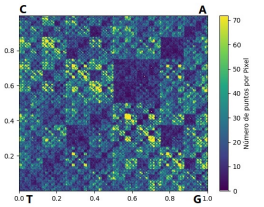
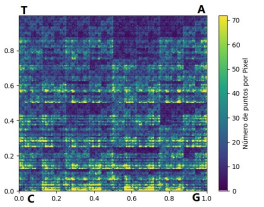
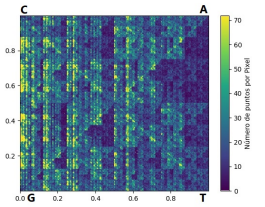
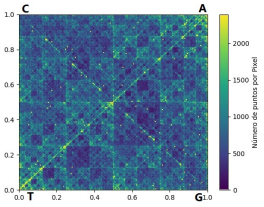
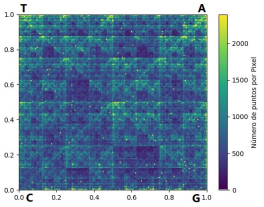
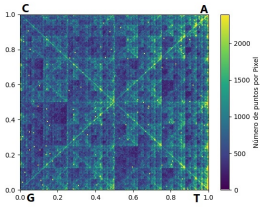
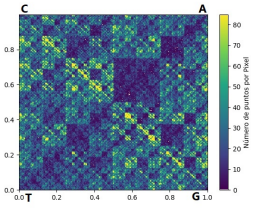
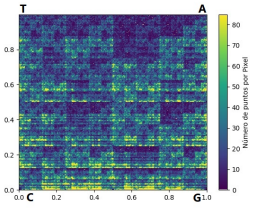
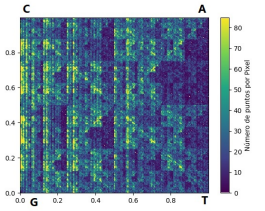
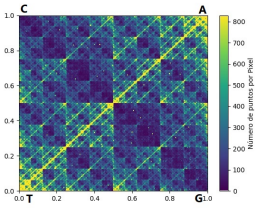
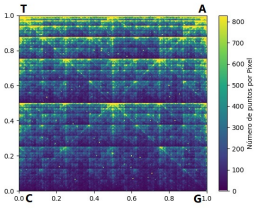
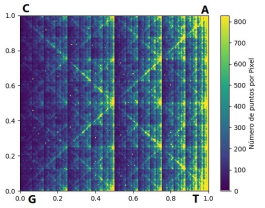
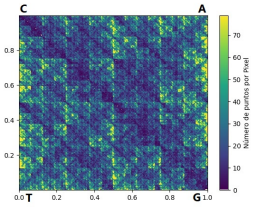
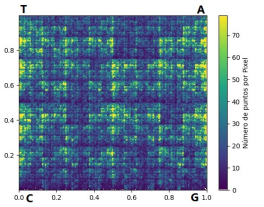
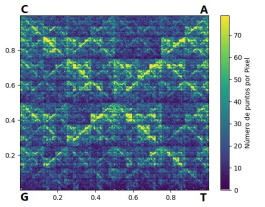
Especie		Cuadro 1	Cuadro 2	Cuadro 3
Penicillium Roqueforti	Genoma			
	Genes			
Trametes Versicolor	Genoma			
	Genes			
Trametes Coccinea	Genoma			
	Genes			

Cuadro 3.8: Representaciones de genomas y genes de hongos 2.

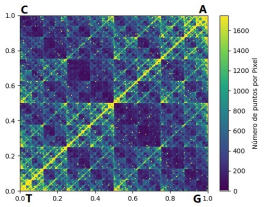
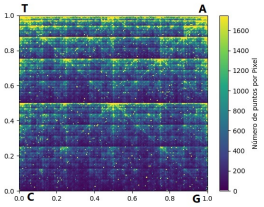
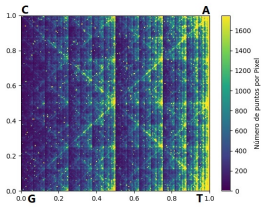
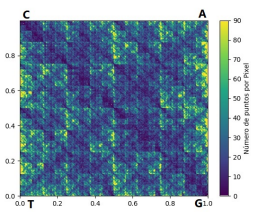
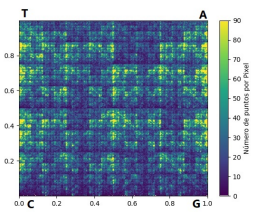
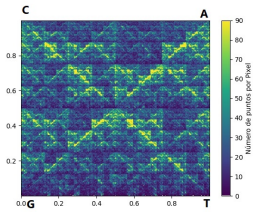
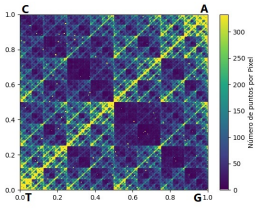
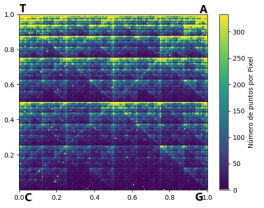
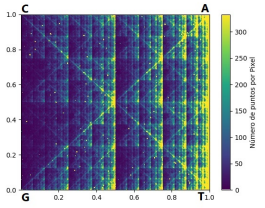
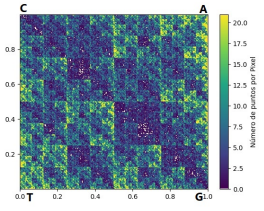
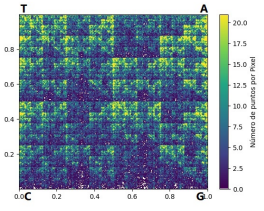
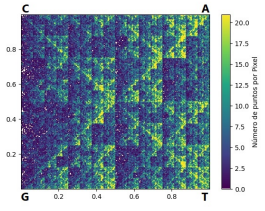
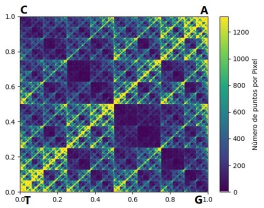
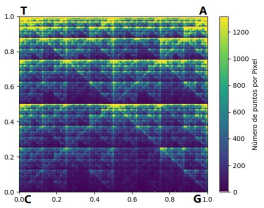
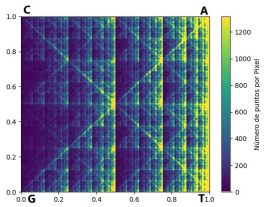
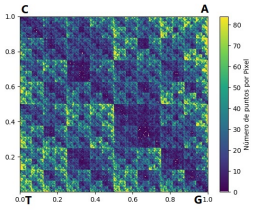
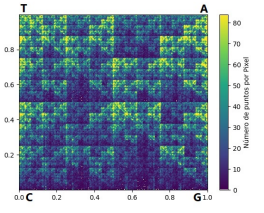
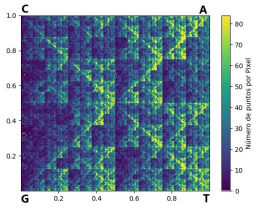
Información de Especies						
Dominio	Fungi					
Clase	Agaricomycetes		Eurotiomycetes		Homobasidiomycetes	
Orden	Agaricales		Eurotiales		Poriales	
Familia	Amanitaceae		Trichocomaceae		Coriolaceae	
Género	*Amanita*		*Penicillium*		*Trametes*	
Especie	*Amanita Muscaria*	*Amanita Inopinata*	*Penic. Oxalicum*	*Penic. Roqueforti*	*Trametes Versicolor*	*Trametes Coccinea*
% de Genes	16.4528 %	15.6727 %	23.4975 %	23.4975 %	12.4938 %	13.7107 %

Cuadro 3.9: Tabla con la taxonomía de las especies utilizadas del reino Fungi.

3.4.4. Plantas

Especie		Cuadro 1	Cuadro 2	Cuadro 3
Oryza Sativa	Genoma			
	Genes			
Panicum Hallii	Genoma			
	Genes			
Brassica Rapa	Genoma			
	Genes			

Cuadro 3.10: Representaciones de genomas y genes de plantas 1.

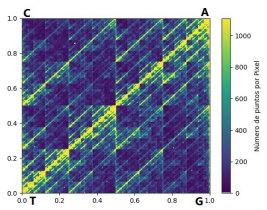
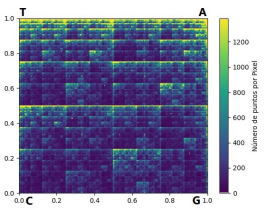
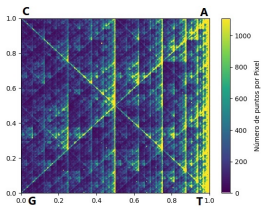
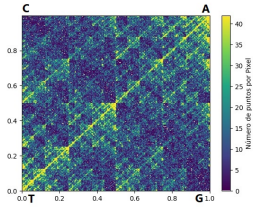
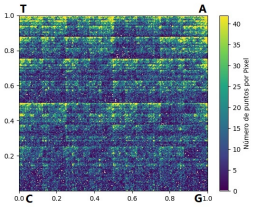
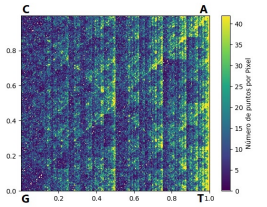
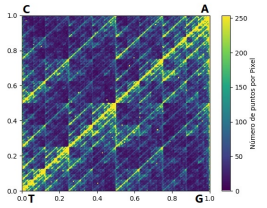
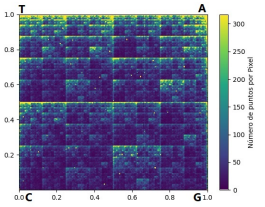
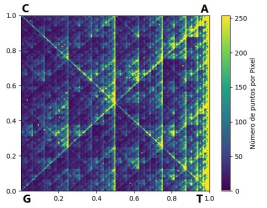
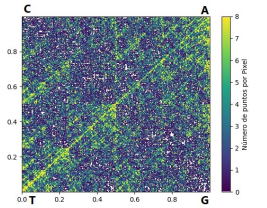
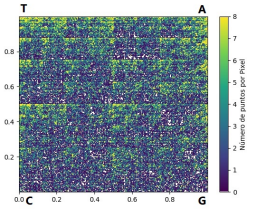
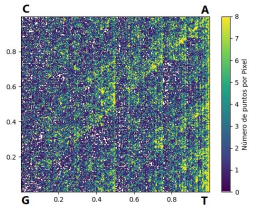
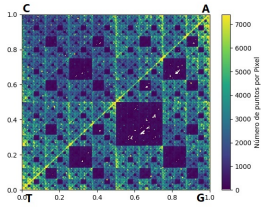
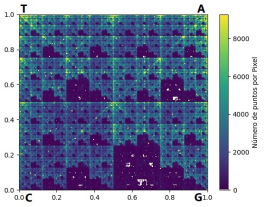
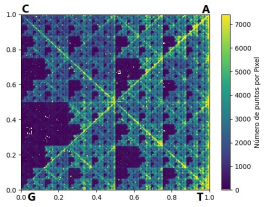
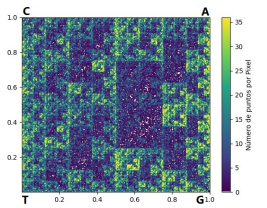
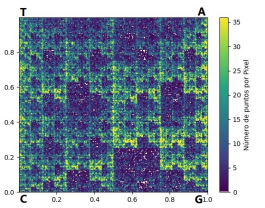
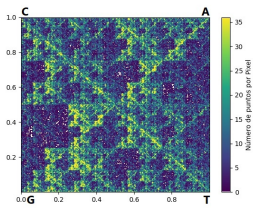
Especie		Cuadro 1	Cuadro 2	Cuadro 3
Brassica Oleracea	Genoma			
	Genes			
Cucumis Melo	Genoma			
	Genes			
Cucumis Sativus	Genoma			
	Genes			

Cuadro 3.11: Representaciones de genomas y genes de plantas 2.

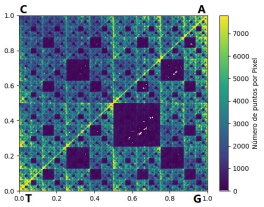
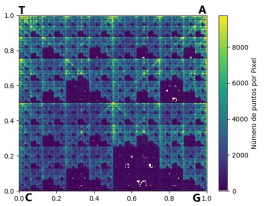
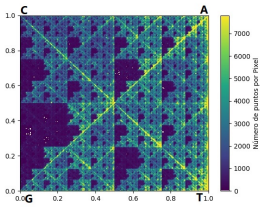
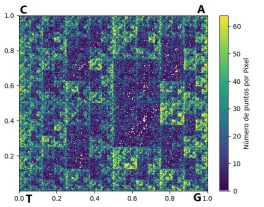
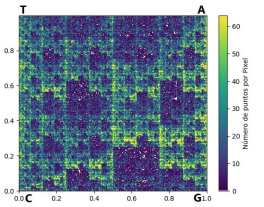
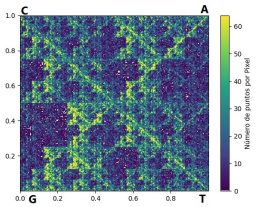
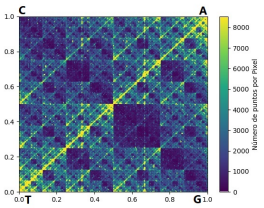
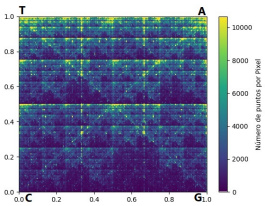
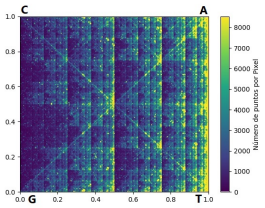
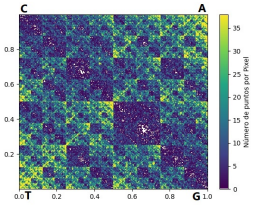
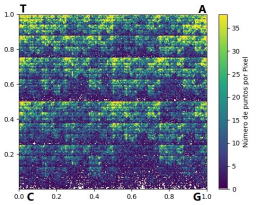
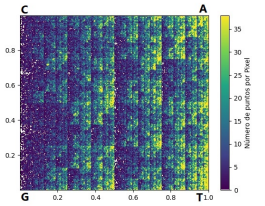
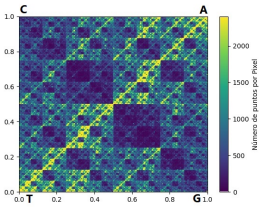
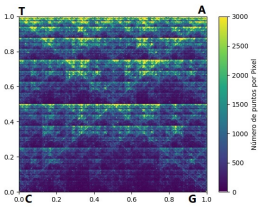
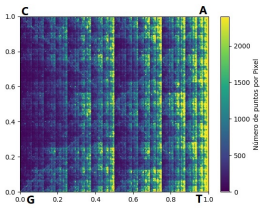
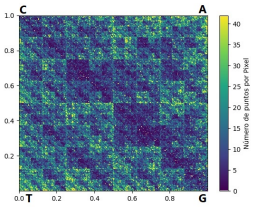
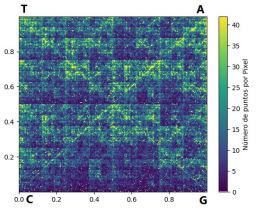
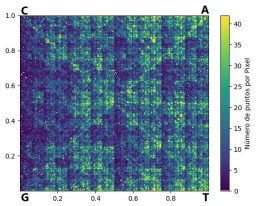
Información de Especies						
Reino	Plantae					
Clase	Liliopsida		Magnoliopsida			
Orden	Cyperales		Capparales		Violales	
Familia	Poaceae		Brassicaceae		Cucurbitaceae	
Género	*Oryza*	*Panicum*	*Brassica*		*Cucumis*	
Especie	*Oryza Sativa*	*Panicum Hallii*	*Brassica Rapa*	*Brassica Oleracea*	*Cucumis melo*	*Cucumis sativus*
% de Genes	4.0459 %	3.5502 %	9.6259 %	5.1454 %	6.5118 %	6.3928 %

Cuadro 3.12: Tabla con la taxonomía de las especies utilizadas del reino Plantae.

3.4.5. Animales

Especie		Cuadro 1	Cuadro 2	Cuadro 3
Apis Mellifera	Genoma			
	Genes			
Apis Cerana	Genoma			
	Genes			
Mus Caroli	Genoma			
	Genes			

Cuadro 3.13: Representaciones de genomas y genes de animales 1.

Especie		Cuadro 1	Cuadro 2	Cuadro 3
Mus Musculus	Genoma			
	Genes			
Octopus Sinensis	Genoma			
	Genes			
Pecten Maximus	Genoma			
	Genes			

Cuadro 3.14: Representaciones de genomas y genes de animales 2.

Información de Especies						
Reino	Animalia					
Clase	Insecta		Mammalia		Cephalopoda	Bivalvia
Orden	Hymenoptera		Rodentia		Octopoda	Pectinida
Familia	Apidae		Muridae		Octopodidae	Pectinidae
Género	*Apis*		*Mus*		*Octopus*	*Pecten*
Especie	*Apis Mellifera*	*Apis Cerana*	*Mus Musculus*	*Mus Caroli*	*Octopus Sinensis*	*Pecten Maximus*
% de Genes	3.7893 %	3.2176 %	0.8227 %	0.4945 %	0.4475 %	1.7546 %

Cuadro 3.15: Tabla con la taxonomía de las especies utilizadas del reino Animalia.

Capítulo 4

Conclusiones

La identificación de las regiones genéticas ha sido de gran utilidad para el estudio de enfermedades, creación de fármacos, entre otros, por lo que es de suma importancia conocerlas, sin embargo, no es una tarea sencilla de lograr.

Con este trabajo se aporta al reconocimiento de patrones genéticos partiendo como idea central el artículo *Chaos game representation of gene structure* escrito por J. H. Jeffrey[27], en que se hace uso del algoritmo llamado Juego de Caos y se obtiene una representación gráfica de cualquiera secuencia de ADN iniciando con la pregunta abierta *¿El patrón que se observa en el genoma varía con respecto al de los genes?* como eje central de la tesis.

La biología molecular es base fundamental para esta investigación, ya que, para darle una interpretación adecuada a las gráficas obtenidas por el Juego del Caos, primero se tiene que entender qué son las regiones codificantes. En este sentido, la bioinformática juega un papel fundamental para la obtención de las regiones codificantes contenidas en la base de datos *GenBank*.

Los fractales y los sistemas dinámicos han sido otra parte esencial para el desarrollo y formalización de esta investigación, ya que para comprender el algoritmo del Juego del Caos y los resultados fractales que se obtuvieron, se tuvo que hacer un meticuloso estudio de los fractales y de los sistemas de funciones iteradas y atractores.

Haciendo uso del lenguaje de programación Python3 y varias paqueterías de este lenguaje, se programó el algoritmo del Juego del Caos para realizar las representaciones gráficas, tanto del genoma, como de las regiones codificantes en los tres tipos de diferentes cuadros de seis especies pertenecientes a cada reino biológico, en total de treinta especies. Lo que se puede concluir, después de realizar un análisis cualitativo descrito en la Sección 3.3 de todas las figuras obtenidas, que estas se pueden dividir en dos partes correspondientes a la clasificación tipo de célula.

En las especies cuyo tipo de célula es procariota, no se logró distinguir visualmente la diferencia entre la estructura del genoma y la estructura perteneciente a las regiones codificantes. Sin embargo, hay que tener presente que una característica de este tipo de célula, es la alta densidad del material genético codificante en el genoma (alrededor del 90 %); por lo que, esto podría influir en la falta de claridad para caracterizar la arquitectura genética de estas especies.

En los cuadros de información de especies de reino Monera, se puede apreciar que los tres géneros que fueron considerados pertenecen a clases distintas, y además, las representaciones correspondientes a genomas son distintas entre clases. Lo cual genera una pregunta muy significativa: ¿Será que esta representación puede ayudar a la construcción de árboles filogenéticos?

En cuanto a las conclusiones obtenidas de células eucariotas, se puede divisar una gran diferencia en las representaciones gráficas de las regiones codificantes y el genoma, donde parece haber un patrón casi completamente distinto. Este resultado se puede deber a la baja densidad de regiones codificantes en genoma (en su mayoría es menor al 50 %); este rasgo es más notorio en las especies pertenecientes a los reinos Plantae y Animalia. Por otro lado, en los reinos Fungi y Protista, la diferencia observada entre la estructura genética y la estructura del genoma completo es muy sutil. Hay que mencionar que, entre más alto sea el porcentaje de la región codificante perteneciente al genoma, la distinción de patrones entre estas dos secuencias es más compleja.

A cerca de la similitud visual entre estructuras, se puede observar que en el reino protista las estructuras de genomas son distintas entre clases.

En el reino Fungi los cuadros de las representaciones del genoma perteneciente a las clases Agaricomycetes y Eurotiomycetes poseen un gran parecido visual. No obstante, las representaciones de los genes entre estas dos clases son medianamente diferente.

En el reino plantae hay un parecido visual entre las representaciones genómicas de los cuatro especímenes de la clase Magnoliopsida, sen cambio, las representaciones de los genes parecen ser diferente.

Por último, en el reino Animalia las clases Cephalopoda y Bivalvia, las representaciones graficas de los especímenes elegidos tienen un parecido, pero las representaciones de las regiones codificantes parecen diferir.

Para realizar un análisis con mayor exactitud se podría investigar más sobre medidas en imágenes para así, establecer una distancia entre imágenes y lograr medir cuantitativamente que tan distintas son las representaciones genómicas y las correspondientes a las regiones codificantes entre reinos, filos, clases, orden y familias.

El usar diferentes cuadros para ejecutar el Juego del Caos tuvo efectos positivos al realizar

el análisis, ya que, sirvieron para que visualmente se acentuaran los patrones; en particular, para examinar la densidad de los cuadros pertenecientes a las diagonales. Sin embargo, la información que revelan es la misma. Si se hiciera un programa de computo para analizar las figuras obtenidas bastaría con usar solamente un tipo de cuadro. También resultaría interesante realizar un programa que obtenga las subsecuencias más frecuentes.

Cabe mencionar que se trabajó con las regiones codificantes en el sentido de 5' a 3', por lo que no se hizo distinción en el sentido de la región codificante, pues los marcos abiertos de lectura también pueden encontrase en la hebra complementaria, para más detalles véase [51] y [55]. Si se hiciera esta distinción sería equivalente a trabajar con la hebra de ARNm y esto cambiaría las figuras obtenidas, ya que ciertas cantidades de secuencias serían graficadas en su base complementaria, es decir, algunos puntos serían reflejados o rotados dependiendo del cuadro con que se trabaje, por lo que este cambio tendría un gran impacto en la estructura de los fractales obtenidos.

Finalmente, sería muy atractivo y enriquecedor realizar la representación gráfica de las regiones no codificantes del ADN, y así poder comparar la estructura resultante con los patrones del genoma y la estructura de las regiones codificantes, en particular, para las especies que poseen un alto porcentaje de material genético codificante en el genoma. Por otra parte, para el caso de células eucariotas sería interesante observar las comparaciones entre las representaciones de los intrones y resto del ADN no codificador.

Apéndice A

Clasificación de los Seres Vivos

Para clasificar a los seres vivos, la sistemática como rama de la biología utiliza la taxonomía, que es el área del conocimiento encargada de establecer las reglas de clasificación. Los taxonomistas se basan en diversos aspectos para clasificar a los seres vivos, tales como: su distribución en la Tierra, por el estudio de los restos fósiles, por el funcionamiento de sus órganos, por la información genética que poseen y otros factores. [20]

Los reinos representan cada una de las grandes subdivisiones taxonómicas. Los reinos clásicos son Animalia, Plantae, Fungi, Protista y Monera. Sin embargo, en la actualidad estos reinos han sido refinados; como el reino Monera, que es una clasificación que incluye los dominios Archaea y Eubacteria, pero estas bacterias son muy distintas. En 1990 Woese presentó una nueva forma de clasificar los seres vivos en tres dominios: Bacterias, Arqueobacterias y Eucariontes.[20]

A.1. Reino Monera

Comprende las formas de vida más simples y primitivas que se conocen. Son seres unicelulares del tipo procariota; pueden ser autótrofos[1] o heterótrofos[2]. Estos seres pueden subsistir en todos los ambientes del planeta, incluso dentro de otros organismos. Los organismos más conocidos pertenecientes a este reino son las bacterias.[20]

Lo conforman: bacterias (eubacterias), algas verdeazuladas (Cyanobacterias) y arqueobacterias. Inicialmente las arqueobacterias eran consideradas dentro de este reino, pero

[1]Esta palabra es neologismo del griego, compuesta por *αυτοζ (autos) = que actúa por sí mismo o sobre sí mismo*, y *τροφοζ (trophos) = que nutre*. Se refiere a un organismo que es capaz de elaborar su propia materia orgánica a partir de sustancias inorgánicas.

[2]También es neologismo griego, compuesta por *ετεροζ (héteros) = distinto, otro* y *τροφοζ (trophos) = que nutre*. Son organismos que se alimentan de sustancias orgánicas elaboradas por otros y así obtienen la energía.

con los avances tecnológicos se dieron a conocer características especiales que las diferencian de las bacterias; por esta razón, se considera que conforman un Dominio separado: Archaea[3]. Sin embargo, en este trabajo no se hará esa distinción. También las algas y cianobacterias son productores primarios, es decir, forman la base de la alimentación de todas las cadenas tróficas. En la Figura A.1[4] se muestran algunos ejemplares. [20]

(a) *Salmonella*: es un género bacteriano, provoca la enfermedad salmonelosis que afecta el aparato intestinal en animales y humanos.

(b) El color amarillo es debido a la arquea *Halobacterium salinarum*. Salinas de Fuentecaliente en las Islas Canarias, España.

Figura A.1: Ejemplos de Monera.

A.2. Reino Protista

El reino protista comprende todos aquellos eucariotas que no pueden ser clasificados como animales, ni como plantas u hongos. Este es el reino más diverso. Agrupa a un conjunto muy diverso de organismos, generalmente monocelulares, tanto autótrofos como heterótrofos; con o sin pigmentos; móviles o sésiles; unicelulares, coloniales e incluso pluricelulares; además, son organismos de hábitat húmedo o acuático. En la Figura A.2[5] se muestran algunos ejemplares provenientes de este reino.[20]

La variedad mencionada hace que sea difícil caracterizarlos, excepto en los rasgos comunes de todo ser eucariótico; sin embargo, existen dos grandes grupos: los protozoos[6]que son seres unicelulares y heterótrofos, y las algas que son seres autótrofos que pueden ser unicelulares o pluricelulares. [20]

[3]Del griego $\alpha\rho\chi\alpha\iota\alpha$ *(arkhaía) = las antiguas*, pues son de las estructuras moleculares más antiguas que se haya estudiado.

[4]Figuras tomadas de [58] [40]

[5]Figuras tomadas de [47] y [5]

[6]Palabra que deriva del griego, $\pi\rho\omega\tau o\zeta$ *(protos) = primero, primitivo* y $\zeta\omega o\nu$ *(zoon) = animal*, ya que se les considera animales unicelulares primitivos.

(a) Algas Pardas. Los pigmentos fotosintéticos les permite realizar la fotosíntesis a diferentes niveles del océano, hasta en profundidades donde solo llegan las longitudes de onda más cortas (verde-azules) de la luz solar.

(b) Ameba 'come cerebros'. El parásito llega al cerebro y destruye el tejido cerebral.

Figura A.2: Ejemplos Protistas.

A.3. Reino Fungi

Este reino está compuesto por hongos, que son seres eucariotas y pluricelulares, como las levaduras, los mohos y las setas. Estos seres comparten características fundamentales con las plantas: tienen cada una de sus células rodeadas individualmente de una pared celular y viven fijos al sustrato; sin embargo, no pueden realizar la fotosíntesis. Por otro parte, los hongos son seres heterótrofos, como los animales, pero no pueden cazar. Pueden vivir en cualquier ecosistema del planeta, siempre que haya materia orgánica, agua y una temperatura apropiada.[20]

Estos seres habitan aguas dulces o saladas, pero la mayoría son terrestres y viven en lugares húmedos y ocultos del sol; además, la cantidad de formas, colores y tamaños que tienen los hongos es inmensa. En la Figura A.3 [7] se dan un par de ejemplos.

[7]Figuras tomadas de [23] y [7]

(a) *Mycena lucentipes*, prospera en la madera de los árboles con flor de los bosques lluviosos de Brasil y Puerto Rico.

(b) *Leratiomyces*. Hongo perteneciente los bosques de Nueva Gales del Sur, Australia.

Figura A.3: Ejemplos Hongos.

A.4. Reino Plantae

Se considera de este grupo a todas las formas de vida vegetal terrestre, son organismos pluricelulares, autótrofos, están adaptados a la vida terrestre, obtienen de la luz solar la energía necesaria para vivir y poseen células eucariotas. Suelen vivir fijas al sustrato y no se desplazan, sin embargo, realizan algunos movimientos, como curvaturas, o apertura y cierre de flores. Se diferencian de las algas por desarrollar tejidos especializados. En la Figura A.4 [8] se encuentras algunos ejemplares.

Estos organismos se caracterizan por realizar el proceso de la fotosíntesis[9], el cual es la forma de nutrición del reino vegetal y además forman la base de la alimentación de todas las cadenas tróficas. Para realizar la fotosíntesis necesitan dióxido de carbono, agua y sales minerales, además de la energía que obtienen de la luz solar.[20]

[8] Figuras tomadas de [29] y [15]

[9] Palabra de proveniente del griego, *φωτοζ (photo) = luz*, *συνθεσιζ (synthesis) = composición, combinación de elementos para formar algo.* Ya que se sintetizan sustancias orgánicas a partir de otras inorgánicas, utilizando la energía luminosa.

(a) *Nymphaeaceae* son una familia de angiospermas. Habitan en aguas quietas o de corriente lenta de hasta 2 m de profundidad, siendo tolerantes a la escasez de oxígeno.

(b) *Wisteria floribunda* o Glicina Japonesa. Con un superficie 1,990 metros cuadrados está localizada en Ashikaga Flower Park; se encuentra sostenida por pilares de acero, permitiendo a los visitantes caminar por debajo de ella y admirar los colores cuando florece.

Figura A.4: Ejemplos Plantas.

A.5. Reino Animalia

Los seres de este reino, se caracterizan por tener una enorme diversidad ecológica, morfológica y conductual, como el reconocimiento del peligro potencial. Se distinguen de los otros reinos eucariotas por carecer de clorofila (no hacen fotosíntesis) y pared celular (como las plantas y hongos), así como por su reproducción, casi enteramente sexual. Su capacidad de movimiento puede ser autónoma o fija al sustrato y han colonizado todos los ambientes terrestres.[2]

Los animales suelen dividirse en dos grandes grupos: los vertebrados, son aquellos que poseen un esqueleto interno formado por huesos o cartílago como los peces, anfibios, reptiles, aves y mamíferos; y los invertebrados que no poseen esqueleto interno, en este grupo se encuentra el 95 % de las especies conocidas de este reino, como artrópodos: insectos, arañas y crustáceos; moluscos: caracoles, calamares y pulpos; equinodermos: erizos y estrellas de mar; y anélidos como la lombriz de tierra. En la Figura A.5[10] se muestran un par de ejemplos [2].

[10]Figuras tomadas de [39] y [52]

(a) *Orcinus orca* son los mayores depredadores de los océanos, llegan a pesar más de cinco toneladas y medir hasta nueve metros.

(b) *Elephantidae* comúnmente conocidos como elefantes. Debido a la caza ilegal actualmente las mayoría de las poblaciones de elefantes africanos que sobreviven nacen sin colmillos [52].

Figura A.5: Ejemplos Animales.

Apéndice B

Transformaciones Lineales

Las transformaciones lineales son funciones que 'conservan' la estructura. En geometría una clase de transformaciones lineales son las rotaciones, reflexiones y proyecciones, estas se suelen usar para el estudio de movimientos rígidos en $\mathbb{R}^n$.

Definición B.0.1. Sea (X, d) un espacio métrico. La transformación en X es una función $f : X \to X$, que asigna exactamente un punto $f(x) \in X$ para cada punto $x \in X$. Si $S \subset X$ entonces $f(S) = \{f(x) : x \in S\}$. f es inyectiva si $x, y \in X$ con $f(x) = f(y)$ implica $x = y$. f es sobreyectiva si $\forall y \in f(X)$ existe $x \in X$ tal que $f(x) = x$. f se le llama invertible si es inyectiva y sobreyectiva. En este caso es posible definir la transformación $f^{-1} : X \to X$, llamada la inversa de f, por $f^{-1}(y) = x$, donde $x \in X$ es el único punto tal que $y = f(x)$.

Definición B.0.2. Sean $(V, +_v, \cdot_v)$ y $(W, +_w, \cdot_w)$ dos $k-$espacio vectoriales. Una función $f : V \to W$ se llama transformación lineal de V es W si cumple:

- $f(v_1 +_v v_2) = f(v_1) +_w f(v_2) \quad \forall v_1, v_2 \in V$
- $f(\lambda \cdot_v v) = \lambda \cdot_w f(v) \quad \forall \lambda \in K, \forall v \in V$

Ejemplo B.0.1. Para $0 \leq \theta < 2\pi$ se define $T_\theta : \mathbb{R}^2 \to \mathbb{R}^2$ mediante

$$T_\theta(a_1, a_2) = (a_1 cos\theta - a_2 sen\theta, a_1 sin\theta + a_2 cos\theta)$$

T_θ se le denomina rotación en θ y es una transformación lineal.

Para demostrar que T es lineal sean $(a_1, a_2), (b_1, b_2) \in \mathbb{R}^2$ y $c \in \mathbb{R}$.

$$T_\theta\left((a_1, a_2) + c(b_1, b_2)\right) = T_\theta\left(a_1 + cb_1, a_2 + cb_2\right)$$

$$= ((a_1 + cb_1)cos\theta - (a_2 + cb_2)sen\theta, (a_1 + cb_1)sin\theta + (a_2 + cb_2)cos\theta)$$
$$= (a_1 cos\theta - a_2 sen\theta, a_1 sin\theta + a_2 cos\theta) + c(b_1 cos\theta - b_2 sen\theta, b_1 sin\theta + b_2 cos\theta)$$
$$= T_\theta(a_1, a_2) + cT_\theta(b_1, b_2)$$

Por lo tanto es lineal.

B.1. Transformaciones Lineales en la Geometría

Las transformaciones lineales pueden ser usadas para proyectar, reflejar, rotar o girar. Son una herramienta para ajustar geométricamente objetos o describir movimiento.

Existe una correspondencia uno a uno con el espacio de la matrices y las transformaciones lineales, que permite estudiar las propiedades de una para estudiar las propiedades de la otra.

Definición B.1.1. Sea $\beta = \{x_1, \ldots, x_n\}$ una base ordenada para un espacio vectorial V dimensionalmente finito. Para $x \in V$ se define al vector ordenada de x relativo a β, denotado por $[x]_\beta$, mediante

$$[x]_\beta = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$$

Supóngase que V y W son espacios vectoriales dimesionalmente finitos con bases ordenadas $\beta = \{x_1, \ldots, x_n\}$ y $\gamma = \{y_1, \ldots, y_m\}$ respectivamente. Sea $T : V \to W$ lineal. Entonces existen escalares únicos $a_{ij} \in F(i = 1, \ldots, m \; y \; j = 1, \ldots, n)$ tales que

$$T(x_j) = \sum_{i=1}^{m} a_{ij} y_j \quad para \; 1 \leq j \leq n$$

Definición B.1.2. Utilizando la definición anterior, llamaremos a la matriz A de $m \times n$ definida mediante $A_{ij} = a_{ij}$ la matriz que representa a T en las bases ordenadas β y γ y se escribirá $A = [T]_\beta^\gamma$.

Definición B.1.3. Una transformación $w : \mathbb{R}^2 \to \mathbb{R}^2$ de la forma

$$w(x_1, w_2) = (ax_1 + bx_2 + e, cx_1 + dx_2 + f)$$

donde $a, b, c, d, e, f \in \mathbb{R}$, es llamada una transformación afín (en dos dimensiones).

Se suele usar la notación

$$w(X) = w\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = Ax + t$$

Estas matrices tienen propiedades geométricas muy importantes, ya que las transformaciones matriciales son reflexiones, compresiones-expansiones, cortes, rotaciones y proyecciones.

Siempre se puede reescribir la matriz de la forma:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 cos\theta_1 & -r_2 sin\theta_2 \\ r_1 sin\theta_1 & r_2 cos\theta_2 \end{pmatrix}$$

donde (r_1, θ_1) son las coordenadas polares del punto (a, c) y $\left(r_2, \theta_2 + \frac{\pi}{2}\right)$ son las coordenadas polares del punto (b, d).

En general, una transformación afín $w(x) = Ax + t \in \mathbb{R}^2$ consiste en una transformación lineal que deforma el espacio original seguido de una traslación o corrimiento especificado por el vector t.

B.1.1. Reflexiones

Una reflexión es una transformación lineal T de un espacio vectorial V en sí mismo, en el que existe un hiperplano de puntos fijos, es decir, de puntos cuyas imágenes por T coinciden con ellos mismos; tal conjunto se le denomina hiperplano de reflexión.

Si se quiere una reflexión sobre el eje x, llámese R_x, basta fijarse en la imagen que desea obtener de los vectores canónicos[24].

$$R_x(e_1) = R_x\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad R_x(e_2) = R_x\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

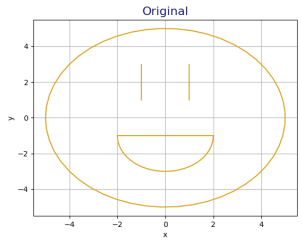
$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad R_x(x, y) = (x, -y)$$

De manera análoga se puede reflejar con respecto el eje x, la diagonal o al origen.

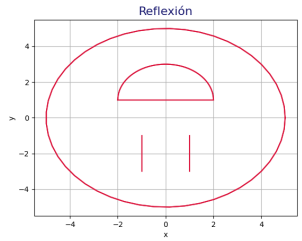
$$R_y(x, y) = (-x, y), \quad A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$R_d(x,y) = (y,x), \quad A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

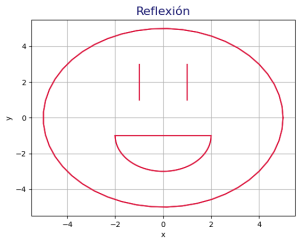
$$R_o(x,y) = (-x,-y), \quad A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$



(a) Imagen original



(b) Reflexión con respecto al eje x



(c) Reflexión con respecto al eje y

Figura B.1: Reflexiones

B.1.2. Rotaciones

Otro tipo de transformaciones en el plano es la rotación, que es un giro a cualquier punto en el plano. En general, cualquier rotación de un ángulo ϕ en sentido antihorario sobre el origen se escribe de la siguiente forma:

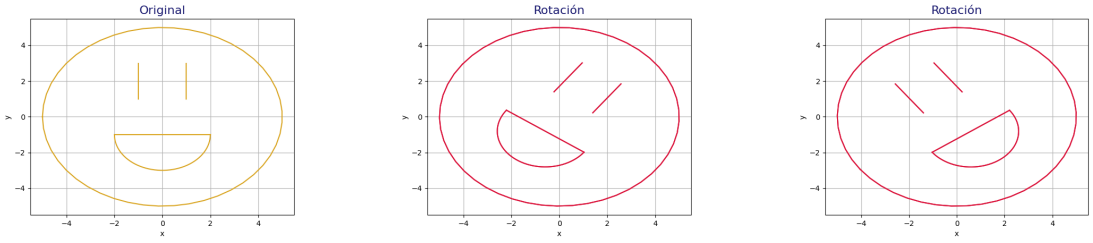
$$R_\phi(e_1) = R_\phi\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} cos(\phi) \\ sin(\phi) \end{bmatrix}, \quad R_\phi(e_2) = R_\phi\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} -sin(\phi) \\ cos(\phi) \end{bmatrix}$$

$$A = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}$$

Y cualquier rotación de un ángulo φ en sentido horario sobre el origen se escribe de la siguiente forma

$$R_\varphi(e_1) = R_\varphi\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} cos(\phi) \\ -sin(\phi) \end{bmatrix}, \quad R_\varphi(e_2) = R_\varphi\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} sin(\phi) \\ cos(\phi) \end{bmatrix}$$

$$A = \begin{bmatrix} cos(\phi) & sin(\phi) \\ -sin(\phi) & cos(\phi) \end{bmatrix}$$



(a) Imagen original (b) Expansión con $\theta = \pi/5$ (c) Expansión con $a = 1/8$ y $b = 3$

Figura B.2: Rotaciones

B.1.3. Compresiones y Expansiones

Las comprensiones y expansiones son escalamientos a lo largo de los ejes coordenados. Para un factor de escalamiento $c > 0$, que satisface $0 < c < 1$ entonces se trata de una compresión y si $c > 1$ se trata de una expansión [4].

Si solo se desea expandir o compresionar en la dirección del eje x con un factor c, entonces la transformación se puede deducir de la siguiente manera:

$$E_c(e_1) = E_c\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}c\\0\end{bmatrix}, \quad E_c(e_2) = E_c\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}0\\1\end{bmatrix}$$

$$A = \begin{bmatrix}c & 0\\0 & 1\end{bmatrix}$$

Por otro lado, si se desea expandir o compresionar en la dirección del eje y con un factor c, entonces la transformación se puede deducir de la esta manera:

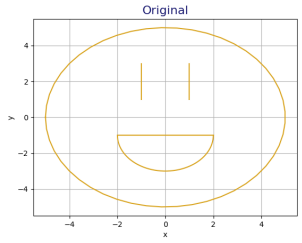
$$E_c(e_1) = E_c\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}, \quad E_c(e_2) = E_c\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}0\\c\end{bmatrix}$$

$$A = \begin{bmatrix}1 & 0\\0 & c\end{bmatrix}$$

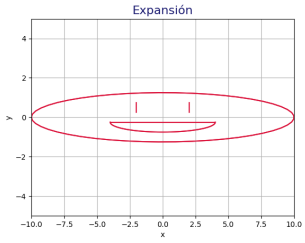
Finalmente, si se desea expandir o compresionar en la dirección del eje x y y con factores a y b respectivamente, entonces la transformación se puede deducir de la siguiente manera:

$$E_{a,b}(e_1) = E_{a,b}\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}a\\0\end{bmatrix}, \quad E_{a,b}(e_2) = E_{a,b}\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}0\\b\end{bmatrix}$$

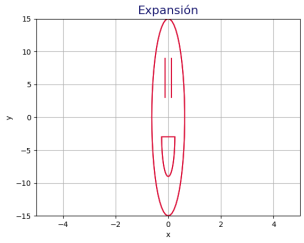
$$A = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$



(a) Imagen original



(b) Expansión con $a = 2$ y $b = 1/4$



(c) Corte con $a = 1/8$ y $b = 3$

Figura B.3: Expansiones y Compresiones

B.1.4. Cortes

Un corte es una transformación lineal que desplaza cada punto en una dirección fija, en una cantidad proporcional a su distancia desde la línea, que es paralela a esa dirección y pasa por el origen. Tiene la propiedad de que hay un vector w tal que $T(w) = w$ y $T(x) - x$ es un múltiplo de w para toda x; si u es ortogonal a w, entonces $T(x) = x + (u \cdot x)w$ [33].

Un corte sobre el eje x se ve de la siguiente manera, donde cada punto se mueve a lo largo de la dirección del eje x una cantidad proporcional a la distancia de x.

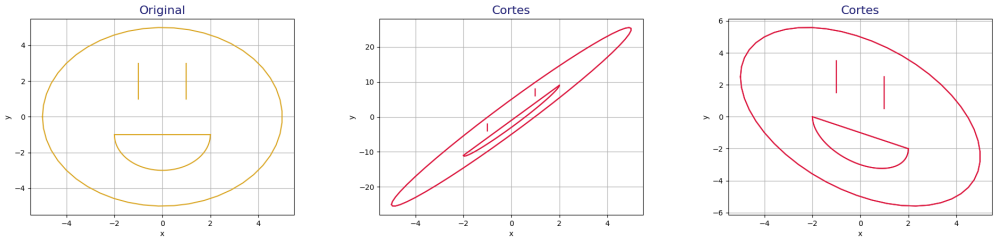
$$C_x(e_1) = C_x\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C_x(e_2) = C_x\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} c \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix}, \quad C_x(x, y) = (x + cy, y)$$

Un corte sobre el eje y se ve de la siguiente manera, donde cada punto se mueve a lo largo de la dirección y una cantidad proporcional a la distancia de y.

$$C_y(e_1) = C_y\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ c \end{bmatrix}, \quad C_y(e_2) = C_y\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}, \quad C_y(x, y) = (x, cx + y)$$



(a) Imagen original (b) Corte con $c = 5$ (c) Corte con $c = -1/2$

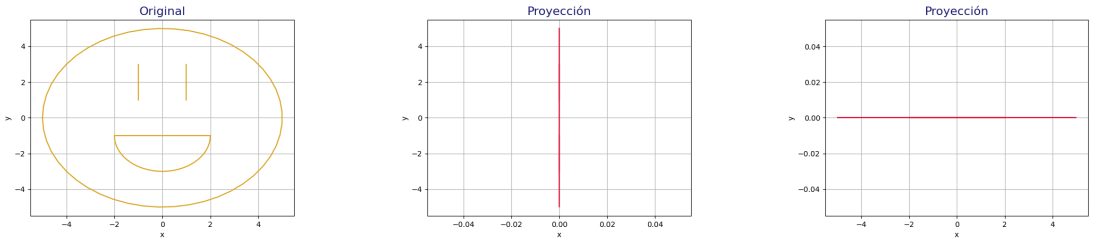
Figura B.4: Cortes

B.1.5. Proyecciones

Una proyección es una transformación lineal idempotente sobre un espacio vectorial. Estas transforman cualquier punto x del espacio vectorial en un punto de un subespacio imagen de la transformación [11]. La proyección sobre el eje x, llámese P_x se expresa de la siguiente forma:

$$P_x(e_1) = P_x\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}, \quad P_x(e_2) = P_x\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix}$$

$$A = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}, \quad P_x(x, y) = (x, 0)$$



(a) Imagen original (b) Proyección con respecto al eje y (c) Proyección con respecto al eje x

Figura B.5: Proyecciones

La proyección sobre el eje y, llamase P_x se expresa de la siguiente manera

$$P_y(e_1) = P_y\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix}, \quad P_y(e_2) = P_y\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}0\\1\end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad P_y(x, y) = (0, y)$$

Apéndice C

Medida y Medida en Fractales

C.1. Breve introducción a la Teoría de la Medida

Definición C.1.1. Sea X un espacio. $\mathcal{F}$ denota a la clase de subconjuntos no vacíos de X, tal que

- Si $X, \emptyset \in \mathcal{F}$
- $A, B \in \mathcal{F} \Rightarrow A \cup B \in \mathcal{F}$.
- $A \in \mathcal{F} \Rightarrow A \setminus X \in \mathcal{F}$

entonces $\mathcal{F}$ es una álgebra.

Definición C.1.2. Sea $\mathcal{F}$ una álgebra tal que

- $A_i \in \mathcal{F}$ para $i \in 1, 2, \cdots \Rightarrow \bigcup_{n=1}^{\infty} A_i \in \mathcal{F}$

entonces $\mathcal{F}$ es llamada $\sigma-$álgebra (o campo de Borel).

El par $(X, \mathcal{F})$ consiste en un arbitrario conjunto X y una $\sigma-$álgebra $\mathcal{F}$ subconjunto de X es llamado espacio medible. A los conjuntos en $\mathcal{F}$ se les llama conjuntos medibles (con respecto a la $\sigma-$álgebra $\mathcal{F}$).

Teorema C.1.1. Sea X un espacio y sea $\mathcal{G}$ un conjunto de subconjuntos de X. Sea $\{\mathcal{F}_\alpha : \alpha \in I\}$ denota al conjunto de todas las $\sigma-$álgebras en X que contienen a $\mathcal{G}$. Entonces $\mathcal{F} = \bigcap_\alpha \mathcal{F}_\alpha$ es una $\sigma-$álgebra.

Definición C.1.3. Sea $\mathcal{G}$ un conjunto de subconjuntos del espacio X. La mínima $\sigma-$álgebra que contiene a $\mathcal{G}$, definida en el teorema C.1.1 es llamada $\sigma-$álgebra generada por $\mathcal{G}$.

Definición C.1.4. Sea (X, d) un espacio métrico. Sea $\mathcal{B}$ la $\sigma-$álgebra generada por los conjuntos abiertos de X. $\mathcal{B}$ es llamado la álgebra de Borel asociada con el espacio métrico y cada elemento de $\mathcal{B}$ es llamado un subconjunto de Borel de X.

C.2. Funciones Medibles

Definición C.2.1. Considérese un espacio medible $(X, \mathcal{F})$. La función $f : X \to \mathbb{R}$ es medible con respecto a la σ-álgebra $\mathcal{F}$, o $\mathcal{F}$ medible, si la imagen inversa de (α, ∞) bajo f es medible es un conjunto medible para todo número real α:

$$f^{-1}((\alpha, \infty)) = \{x \in X : f(x) > \alpha\} \in \mathcal{F} \; \forall \; \alpha \in \mathbb{R}$$

Más aún se puede verificar que $f : X \to \mathbb{R}$ es medible si y solo si la imagen inversa de conjuntos abiertos en $\mathbb{R}$ es medible en $\mathcal{F}$

Ejemplo C.2.1. Sea $(X, \mathcal{F})$ un espacio medibles, toma cualquier conjunto $E \in \mathcal{P}(X)$ y considere la función característica $\mathcal{X}_E : X \to \{0, 1\} \in \mathbb{R}$, esto es:

$$\mathcal{X}_E(x) = \begin{cases} 1, & x \in E \\ 0, & x \in X \setminus E \end{cases}$$

La función característica del conjunto $E \subseteq X$ es una función $\mathcal{F}$ medible si solo si E es un conjunto $\mathcal{F}$ medible. En efecto

$$\mathcal{X}_E^{-1}((\alpha, \infty)) = \{x \in X : \mathcal{X}_E > \alpha\} = \begin{cases} \varnothing, & 1 \leq \alpha \\ E, & 0 \leq \alpha < 1 \\ X, & \alpha < 0 \end{cases}$$

C.3. Medidas

Definición C.3.1. Una medida μ, en una álgebra $\mathcal{F}$, es una función real no negativa $\mu : \mathcal{F} \to [0, \infty) \subset \mathbb{R}$, tal que para cualquier $A_i \in \mathcal{F}$ para $i = 1, 2, \ldots$ con $A_i \cap B_i = \varnothing$ para $i \neq j$ y $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ se tiene:

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i)$$

Definición C.3.2. Se llama espacio de medida a la terna $(X, \mathcal{F}, \mu)$, donde μ es una medida sobre la $\sigma-$álgebra $\mathcal{F}$ de X.

Proposición C.3.1. Sea $(X, \mathcal{F}, \mu)$ un espacio de medida, entonces para $A, B \in \mathcal{A}$:

- $\mu(B) = \mu(B \cap A) + \mu(B \cap A^c)$
- Si $A \subset B$ entonces $\mu(B) = \mu(A) + \mu(B \setminus A)$
- Si $A \subset B$ y $\mu(A) = \infty$, entonces $\mu(B) = \infty$
- Si $A \subset B$ entonces $\mu(A) \leq \mu(B)$
- $\mu(A \cup B) + \mu(A \cap B) = \mu(A) + \mu(B)$
- $\mu(A \cup B) \leq \mu(A) + \mu(B)$
- $A_1, \ldots, A_n \in X$, *entonces*

$$\mu(A_1 \cup \cdots \cup A_n) \leq \mu(A_1) + \ldots \mu(A_n)$$

Definición C.3.3. Sea (X, d) un espacio métrico. Sea $\mathcal{B}$ los subconjuntos de Borel de X. Sea μ una medida en $\mathcal{B}$, entonces μ es llamada la medida de Borel.

Definición C.3.4. Sea (X, d) un espacio métrico compacto. Sea μ una media de Borel en X. Si $\mu(X) = 1$, entonces se dice que μ está normalizada.

Definición C.3.5. La integral (con respecto a μ) de una función simple f es

$$\int_X f(x) d\mu(x) = \int_x f d\mu = \sum_{i=1}^{N} y_i \mu(I_i)$$

Ejemplo C.3.1. El triángulo de Sierpinski tiene medida 0.

Este ejemplo toma lugar en un rectángulo R en el plano Euclidiano. Sea $\mathcal{G}$ los subconjuntos de Borel del rectángulo R y $\mathcal{F}$ la álgebra generada por $\mathcal{G}$. Prueba que existe una única medida en $\mathcal{F}$ tal que $\mu(A) =$ área *de* A. Sea $\widehat{\mu}$ la extensión de μ en $\mathcal{F}$. Sea $\triangle$ que denota al triángulo de Sierpinski contenido en el rectángulo R. Probar que $\triangle \in \mathcal{F}$ y

$$\int_R \chi_{\triangle} = \widehat{\mu}(\triangle) = 0$$

Hay que recordar la construcción de este fractal, véase el Ejemplo 2.2.2 y la Figura 2.10, llámese $\triangle_i$ a la i-ésima iteración, siendo $\triangle_0$ el triángulo inicial de altura a y base b y $\triangle_\infty = \triangle$ denota al triángulo de Sierpinski.

Primero se probará que $\triangle \in \mathcal{F}$.

Es claro que el triángulo inicial es cerrado, por lo tanto, $\triangle_1 \in \mathcal{F}$. En cuanto la primera iteración es fácil ver que pertenece a $\mathcal{F}$ pues este conjunto está formado por tres triángulos y la unión de tres conjuntos cerrados pertene a $\mathcal{F}$.

Siguiendo el mismo razonamiento anterior $\triangle_n \in \mathcal{F}$ pues es una unión finita de conjuntos cerrados y $\triangle_\infty \in \mathcal{F}$ ya que es un unión numerable de conjuntos cerrados y como es sigma álgebra esta unión pertenece a $\mathcal{F}$.

Ahora se deducirá su medida. La medida de triángulo inicial corresponde a la clásica fórmula del área de un triángulo de altura a y base b.

$$\mu(\triangle_0) = \frac{b \cdot a}{2}$$

En la primera iteración, se tiene 3 triángulos de base $\frac{b}{2}$ y altura $\frac{a}{2}$. De esta manera el área es:

$$\mu(\triangle_1) = 3 \cdot \frac{\frac{b}{2} \cdot \frac{a}{2}}{2} = \frac{3}{4}\left(\frac{b \cdot a}{2}\right)$$

En la segunda iteración, ahora se tienen 9 triángulos de altura $\frac{a}{4}$ y base $\frac{b}{4}$, por lo que su medida está dada por:

$$\mu(\triangle_2) = 9 \cdot \frac{\frac{b}{4} \cdot \frac{a}{4}}{2} = \frac{3^2}{4^2}\left(\frac{b \cdot a}{2}\right)$$

Con lo anterior, resulta fácil calcular la medida de la tercera iteración y posteriormente de la $n-$ésima iteración.

$$\begin{aligned} \mu(\triangle_3) &= 27 \cdot \frac{\frac{b}{8} \cdot \frac{a}{8}}{2} = \frac{3^3}{4^3}\left(\frac{b \cdot a}{2}\right) \\ &\vdots \\ \mu(\triangle_n) &= \frac{3^n}{4^n}\left(\frac{b \cdot a}{2}\right) \end{aligned}$$

Y tomando el límite cuando $n \to \infty$ se obtiene la medida de triángulo de de Sierpinski.

$$\mu(\triangle_\infty) = \lim_{n\to\infty}\left(\frac{3}{4}\right)^n \frac{b \cdot a}{2} = \frac{b \cdot a}{2} \lim_{n\to\infty}\left(\frac{3}{4}\right)^n \to 0$$

$\therefore$ el triángulo de de Sierpinski tiene medida 0.

C.4. Medidas en Fractales

Definición C.4.1 (Métrica de Hutchinson)**.** Sea (X, d) un espacio métrico compacto. $\mathcal{P}(X)$ denota al conjunto de medidas normalizadas de Borel en X. La métrica de Hutchinson d_H en $\mathcal{P}(X)$ está definida por

$$d_H(\mu, v) = sup\left\{\left|\int_X f\ d\mu - \int_X f\ dv\right| : |f(x) - f(y)| \leq d(x, y)\ \forall x, y \in X\right\}$$

con $f : X \to \mathbb{R}$, f continua y $\mu, v \in \mathcal{P}(X)$.

La Métrica de Hutchinson brinda una noción de distancia entre dos medidas normalizadas en un espacio compacto [18].

Definición C.4.2. Sea (X, d) un espacio métrico compacto y sea $\mathcal{P}(X)$ el espacio normalizado de medidas de Borel en X. Sea

$$\{X; w_1, w_2, \ldots, w_N; p_1, p_2, \ldots, p_N\}$$

Un sistema de funciones iteradas hiperbólico. El operador de Markov asociado con el sistema de funciones iteradas es una función $M : \mathcal{P}(X) \to \mathcal{P}(X)$ definida como

$$M(v) = p_1 v \circ w_1^{-1} + p_2 v \circ w_2^{-1} + \cdots + p_N v \circ w_N^{-1}$$

para todo $v \in \mathcal{P}(X)$

Teorema C.4.1. Sea (X, d) un espacio métrico. Sea

$$\{X; w_1, w_2, \ldots, w_N; p_1, p_2, \ldots, p_N\}$$

Un sistema de funciones iteradas hiperbólico con probabilidades. Sea $s \in (0, 1)$ el factor de contracción del sistema de funciones iteradas. Sea $M : \mathcal{P}(X) \to \mathcal{P}(X)$ el operador de Markov asociado. Entonces M es un mapeo contractivo con factor s con respecto a la métrica de Huntchinson en $\mathcal{P}(X)$. Tal es

$$d_H(M(v), M(\mu)) = s \cdot d_H(v, \mu) \quad \forall v, \mu \in \mathcal{P}(X)$$

En particular, hay una única medida $\mu \in \mathcal{P}(X)$ tal que

$$M\mu = \mu$$

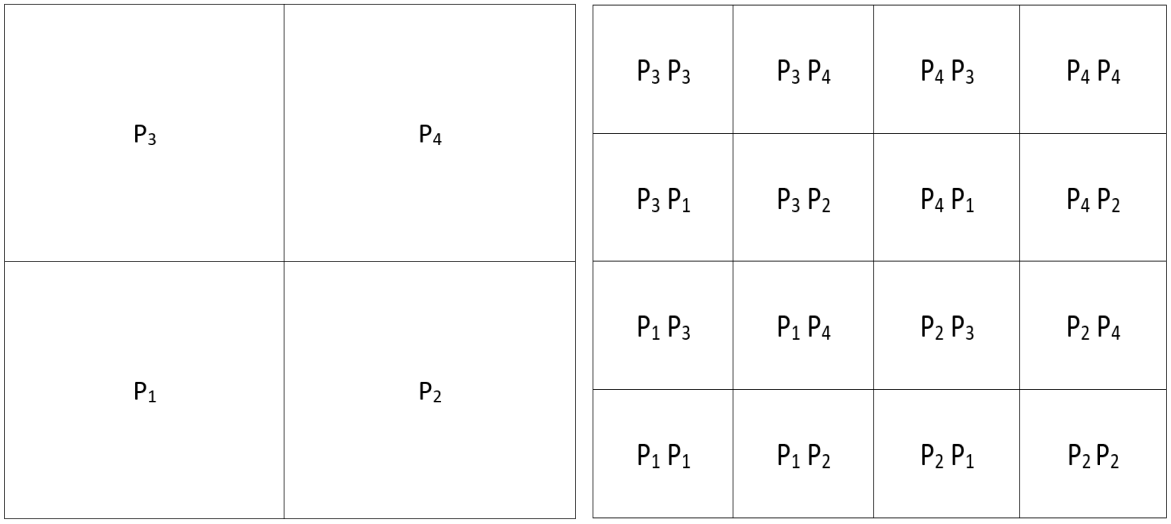
A esta medida se le llama la medida invariante del sistema de funciones iteradas con probabilidades. La demostración de este Teorema se puede leer en [8, Pág, 351].

Para ilustrar el Teorema C.4.1 se dará un ejemplo gráfico de como funciona el operador de Markov, tomado de *Fractals Everywhere* [8, Pág. 352].

Ejemplo C.4.1. Considérese el siguiente SFI

$$\{\blacksquare \subset \mathbb{R}^2; w_1, w_2, w_3, w_4; p_1, p_2, p_3, p_4\}$$

Donde los puntos fijos corresponden a los vértices del cuadrado. El atractor de SFI es ■, el sistema tiene 4 funciones para las cuales la probabilidad de elegir w_i es p_i para $i = 1, 2, 3, 4$.



(a) $\mu(w_i(\blacksquare)) = p_i$ (b) $\mu(w_i \circ w_j(\blacksquare)) = p_i P_j$ para $i, j = 1, 2, 3, 4$

Figura C.1: Ejemplo operador de Markov.

Sea M el operador de Markov asociado. Sea $\mu_0 \in \mathcal{P}(X)$ de modo que $\mu_0(\blacksquare) = 1$. μ_0 puede la medida uniforme, donde $\mu_0(S)$ es el área de $S \in \mathcal{P}(\blacksquare)$. Obsérvese qué sucede con la secuencia $\{\mu_n = M^n(\mu_0)\}$

La medida $\mu_1 = M(\mu_0)$ es tal que $\mu_1(w_i(\blacksquare)) = p_i$ para cada $i = 1, 2, 3, 4$ como se ve en la Figura C.1a. Esto mismo se tiene para $\mu_2 = M^2(\blacksquare)$ se sigue $\mu(w_i \circ w_j(\blacksquare)) = p_i \cdot p_j$ para $i, j = 1, 2, 3, 4$ como se ilustra en la Figura C.1b.

Cuando el operador de Markov es aplicado, la 'masa' en la celda $\blacksquare_{ij\ldots k} = w_i \circ w_j \circ \cdots \circ w_k$ es redistribuido entre las 4 celdas pequeñas $w_1(\blacksquare_{ij\ldots k}), w_2(\blacksquare_{ij\ldots k}), w_3(\blacksquare_{ij\ldots k})$ y $w_4(\blacksquare_{ij\ldots k})$.

Además, la masa de las otras celdas se mapea en subceldas de $\blacksquare_{ij\ldots k}$ de tal manera que la masa total de $\blacksquare_{ij\ldots k}$ permanece igual que antes de que se aplicara el operador de Markov.

De esta manera, la distribución de la 'masa' se define en escalas cada vez más finas a medida que se aplica repetidamente el operador de Markov.

El siguiente teorema y corolario fueron tomados de [8, Pág. 364 y 265].

Teorema C.4.2. Sea (X, d) un espacio métrico completo. Sea

$$\{X; w_1, w_2, \ldots, w_N; p_1, p_2, \ldots, p_N\}$$

un sistema de funciones iteradas con probabilidades. Sea $\{X_n\}_{n=0}^{\infty}$ la órbita producida por el algoritmo aleatorio, comenzado en x_0 producido con el sistema de funciones iteradas con probabilidades.

$$x_n = w_{\sigma_n} \circ w_{\sigma_{n-1}} \circ \ldots w_{\sigma_1}$$

donde los mapeos son elegidos aleatoriamente de acuerdo con las probabilidades,

$$p_1, p_2, \ldots, p_N, \quad para\ n = 1; 2, 3, \ldots$$

Sea μ la medida única invariante del sistema de funciones iteradas, entonces con probabilidad uno

$$\lim_{n\to\infty} \frac{1}{n+1} \sum_{k=0}^{n} f(x_k) = \int_X f(x) d\mu(x)$$

para toda función continua $f : X \to \mathbf{R}$ y toda x_0.

Corolario C.4.1. Sea B una bola abierta en X y sea $\mu(\partial B) = 0$. Sea $\mathcal{N}(B, n) =$ número de puntos en $\{x_0, x_1, \ldots, x_N\} \cap B$ para $n = 1, 2, \ldots$. Entonces con probabilidad uno

$$\mu(B) = \lim_{n\to\infty} \left\{ \frac{\mathcal{N}(B, n)}{(n+1)} \right\}$$

para todo punto x_0. Esto es, la 'masa' de B es proporcional al número de iteraciones producido por el algoritmo aleatorio en B.

Lo que dice que la masa en la bola B es proporcional al número de puntos producidos por el sistema de funciones iteradas con probabilidades, que caen dentro de la bola B.

Como se expone a lo largo de este capítulo, la elección de probabilidades determina la 'textura' de la imagen; es decir, lo que una persona observa como textura corresponde a las densidades de la medida μ asociada al sistema y sus probabilidades.

Apéndice D

Código del Capítulo Fractales y SDD

D.1. Algoritmo de Telaraña para la Función Logística.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import random

#Ecuacion logistica
def logistica(r, x):
    y = r*x*(1-x)
    return(y)

'''Parametros: r- corresponde al parametro de la ecuacion logistica, n el
   numero de iteraciones e inicio el punto de partida '''
r = 2
n = 50
inicio = 0.9

#Algoritmo
ruta1 = np.zeros(n)
ruta1[0] = inicio

for i in range(n-1):
    ruta1[i+1] = logistica(r, ruta1[i])

x, y = np.zeros(2*n), np.zeros(2*n)
x[0] = inicio
y[0] = 0
x[1] = inicio
j = 1
```

```
for i in range(n-1):
    x[j +1] = ruta1[i+1]
    x[j +2] = ruta1[i+1]
    y[j] = ruta1[i+1]
    y[j +1] = ruta1[i+1]
    j = j+2

y[j] = logistica(r, ruta1[n-1])

df = pd.DataFrame({'x':x, 'y':y} )

ec0= np.linspace(0, 1, n)
ecu0 = logistica(r, ec0)

df2 = pd.DataFrame({'x1':np.linspace(0, 1, n), 'y1':ecu0} )
# style
plt.style.use('seaborn-darkgrid')
num = 0

# create a color palette
palette = plt.get_cmap('Set1')
plt.plot(df['x'], df['y'], marker='', color=palette(0), linewidth=1, alpha
    =0.9)
plt.plot(df2['x1'], df2['y1'], marker='', color=palette(num+1), linewidth
    =1, alpha=0.9)
plt.plot(df2['x1'], df2['x1'], marker='', color='k', linewidth=1, alpha
    =0.9)

#plt.legend(loc=4, ncol=2)
# Add titles
plt.title("f(x) = 2x(1-x)", loc='left', fontsize=12, fontweight=0, color='k
    ')
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

Listing D.1: Algoritmo de Telaraña para la función logística

D.2. Algoritmo para Generar la Figura 2.13

```
#Paqueterias
import pandas as pd
```

```
import numpy as np
import plotnine as p9
import random

listaPto = []
actual = [1/2, 1]
'''
Se declaran todas las funciones lineales con diferentes puntos fijos y
    factor de contraccion 1/3
'''

def A1(x):
    return([1/3*x[0],1/3*x[1]])

def A2(x):
    return([1/3*(x[0]-1) + 1,1/3*x[1]])

def A3(x):
    return([1/3*(x[0]-1) + 1,1/3*(x[1]-1) + 1])

def A4(x):
    return([1/3*x[0],1/3*(x[1]-1)+1])

def A5(x):
    return([1/3*(x[0]-1/2)+1/2,1/3*(x[1]-1/2)+1/2])

'''
Funcion que se encarga de elegir las funciones al azar y guardarlas en un
    arreglo
'''

def Sistema2(secuencia, listaPto, actual):

    n =len(secuencia)
    listaPto.append(actual)
    aux = actual

    for i in range(n):
        if secuencia[i] == 1:
            aux = A1(aux)
        elif secuencia[i] == 2:
            aux = A2(aux)
        elif secuencia[i] == 3:
            aux =A3(aux)
        elif secuencia[i] == 4:
```

```
            aux = A4(aux)
        else:
            aux = A5(aux)
        listaPto.append(aux)

Sistema2(lista, listaPto, actual)

graf2 = (p9.ggplot(data = pd.DataFrame(listaPto, columns = ['x', 'y']),
    mapping = p9.aes(x='x', y= 'y')) + p9.geom_point(alpha = 0.1))
```

Listing D.2: Algoritmo para generar la Figura 2.13

D.3. Algoritmo para Generar la Figura 2.14

```
#Paqueterias
import pandas as pd
import numpy as np
import plotnine as p9
import random

#Declaramos la matriz de rotacion
rotacion = np.array([[np.cos(np.pi/2), np.sin(np.pi/2)],[-np.sin(np.pi/2),
    np.cos(np.pi/2)]])
rotacion

#funcion que construye las funciones dado un punto fijo
def AR1(pFijo, x):
    punto = 1/2*np.dot(rotacion,(x-pFijo))+pFijo
    return(punto)

listaPto = []
actual = np.transpose(np.array([[2, 1]]))

#Funcion que se encarga iterar el SFI
def Sistema3(secuencia, listaPto, actual, angulo):
    rotacion = np.array([[np.cos(angulo), -np.sin(angulo)],[np.sin(angulo),
        np.cos(angulo)]])
    pto1 = np.transpose(np.array([[0, 0]]))
    pto2 = np.transpose(np.array([[0, 1]]))
    pto3 = np.transpose(np.array([[1, 0]]))

    n =len(secuencia)
    listaPto.append([actual[0,0], actual[1,0]])
```

```
    aux = actual

    for i in range(n):
        if secuencia[i] == 1:
            aux = AR1(pto1, aux)
        elif secuencia[i] == 2:
            aux = AR1(pto2, aux)
        else:
            aux = AR1(pto3, aux)
        bandera = [aux[0,0], aux[1,0]]
        listaPto.append(bandera)

#Se llena el arreglo
Sistema3(lista, listaPto, actual, np.pi/4)

#Grafica
(p9.ggplot(data = pd.DataFrame(listaPto, columns = ['x', 'y']), mapping =
    p9.aes(x='x', y= 'y')) + p9.geom_point(alpha = 0.1))
```

Listing D.3: Algoritmo para generar la Figura 2.14

D.4. Algoritmo Determinista para Generar la Figura 2.15

```
import numpy as np
import matplotlib.pyplot as plt

a1, b1, c1, d1 = 0.6966-0.4607j, -0.3307-0.0497j, 0.2j, 1.0523 + 0.601j
a2, b2, c2, d2 = -0.8438, 0.5119j, -0.2j, -1.3064

#Funcion 1
def sistema1(z, a = a1, b = b1, c = c1, d = d1):
    return (a*z + b)/(c*z + d)
#Funcion 2
def sistema2(z, a = a2, b = b2, c = c2, d = d2):
    return (a*z + b)/(c*z + d)

# Se crea la base para el conjunto a iterar
x = np.linspace(0, 1, 1000)
zero = np.zeros(len(x))

#Funcion que convierte dos arreglos x, y a un arreglo z de numeros
    complejos.
```

```
def cartAImag(x, y):
    return [complex(x[i], y[i]) for i in range(len(x))]
#Funcion que convierte un arreglo z de numeros complejos a dos arreglos x,
    y.
def imagAcar(z):
    return [i.real for i in z], [i.imag for i in z]

'''
Funcion que aplica el algoritmo Determinista N veces y lo grafica.
'''
def cuadroAplica(n):
    #Conjunto inicial: El cuadrado.
    z1, z2, z3, z4 = cartAImag(x, zero), cartAImag(x, zero+1), cartAImag(
        zero, x), cartAImag(zero+1, x)

    lados = [z1, z2, z3, z4]
    for i in range(n):
        aux = []
        for funcion in [sistema1, sistema2]:
            for z in lados:
                X1, Y1 = imagAcar([funcion(i) for i in z])
                aux.append([X1, Y1])
                if i == n-1:
                    plt.plot(X1, Y1, color = 'black')
        lados = [cartAImag(par[0], par[1]) for par in aux]

    plt.show()
```

Listing D.4: Algoritmo Determinista para generar la Figura 2.15

D.5. Algoritmo Aleatorio para Generar la Figura 2.16

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap
import mpl_scatter_density

# Escalas de color
white_viridis = LinearSegmentedColormap.from_list('white_viridis', [
    (0, '#ffffff'),
    (1e-20, '#440053'),
    (0.2, '#404388'),
    (0.4, '#2a788e'),
    (0.6, '#21a784'),
```

```
    (0.8, '#78d151'),
    (1, '#fde624'),
], N=256)

#Funcion que hace la grafica de densidad con los anteriores colores
def using_mpl_scatter_density(fig, x, y, vmin, vmax):
    ax = fig.add_subplot(1, 1, 1, projection='scatter_density')
    density = ax.scatter_density(x, y, cmap=white_viridis, vmin=vmin, vmax=
        vmax)
    fig.colorbar(density, label='Numero de puntos por Pixel')

#Parametros de los sistemas
a1, b1, c1, d1 = 0.6966-0.4607j, -0.3307-0.0497j, 0.2j, 1.0523 + 0.601j
a2, b2, c2, d2 = -0.8438, 0.5119j, -0.2j, -1.3064

#funcion 1
def sistema1(z, a = a1, b = b1, c = c1, d = d1):
    return (a*z + b)/(c*z + d)
#funcion 2
def sistema2(z, a = a2, b = b2, c = c2, d = d2):
    return (a*z + b)/(c*z + d)

#Funcion que realiza el algoritmo aleatorio, con el sistema de funciones
    iteradas.
#Devuelve un arreglo de numeros complejos, correspondientes a lo orbita de
    z0.
def sistemaDeFucniones(z0, N = 1000):
    ruta = [z0]
    for i in range(0, N):
        funcion = np.random.choice([sistema1, sistema2], 1)
        z0 = funcion[0](z0)
        ruta.append(z0)
    return ruta

numeros = sistemaDeFucniones(0+0j, 500000)

#Grafica de densidad
fig = plt.figure()
using_mpl_scatter_density(fig, [i.real for i in numeros], [i.imag for i in
    numeros], vmin = 0, vmax = 100)
plt.show()
```

Listing D.5: Algoritmo Aleatorio para generar la Figura 2.16

Apéndice E

Código del Capítulo Representación del ADN

E.1. Juego del Caos con Polígonos para Generar las Figuras 3.1 y 3.9

```
import numpy as np
import plotnine as p9
import pandas as pd
'''
Funcion que toma el punto medio entre los puntos A y B.
Recibe como parametros el punto A y B. Devuelve un punto C.
'''
def ptomedio(A, B):
    C = [(A[0]+B[0])/2, (A[1]+B[1])/2]
    return(C)

#Puntos para el triangulo.
puntos = [[0, 0], [1, 0], [1/2, np.sqrt(2)/2]]
'''
Funcion que ejecuta el algoritmo del Juego del Caos.
Recibe como parametros:
- puntos: Los puntos de poligono.
- n: El numero de iteraciones.
- inicio: el punto incial.
- prob: las probas correspondientes a cada vertice.

Devuelve una lista con tas las iteraciones.
'''
```

```
def juegoCaos(puntos, n, incio, prob):
    listaPto = [inicio]

    for i in range(n):
        aux = np.random.choice(3, size = 1, p = prob)
        listaPto.append(ptomedio(puntos[aux[0]], listaPto[-1]))

    return listaPto

listaPto = juegoCaos(puntos, 15000, [0, 0], [1/3, 1/3, 1/3])
#Grafica
(p9.ggplot(data= pd.DataFrame(listaPto, columns = ['x', 'y']), mapping=p9.
    aes(x='x', y='y')) + p9.geom_point(alpha=0.05))
```

Listing E.1: Juego del Caos con Polígonos para generar las Figuras 3.1 y 3.9

E.2. Variante 1 del Juego del Caos en un Cuadrado, Figura 3.8b

```
import pandas as pd
import numpy as np
import plotnine as p9
'''
Funcion que obtiene el punto medio entre dos puntos.
Recibe dos puntos de la forma A = [a, b] y B = [c, d]
y devuelve un punto de la forma C = [e, f]
'''
def ptomedio(A, B):
    C = [(A[0]+B[0])/2, (A[1]+B[1])/2]
    return(C)
'''
Juego del Caos con Restriccion 1.
El vertice elegido no puede ser vecino del vertice elegido anteriormente si
    los dos vertices elegidos previamente son los mismos.
Recibe como parametros:
n: numero de iteaciones.
actual: punto de inicio de la forma [a, b]
puntos: es una lista con los puntos del poligono
Regresa: una lista de puntos lista para graficar
'''

def juegoCaosCuadrado(n, inicio, puntos):
    listaPto = [inicio]
```

```
    elecciones = []
    for i in range(n):
        if i > 2 and elecciones[i-2] == elecciones[i-1]:
            if elecciones[i-1] == 0:
                eleccion = np.random.choice([0, 2], 1)[0]
            elif elecciones[i-1] == 1:
                eleccion = np.random.choice([1, 3], 1)[0]
            elif elecciones[i-1] == 2:
                eleccion = np.random.choice([0, 2], 1)[0]
            elif elecciones[i-1] == 3:
                eleccion = np.random.choice([1, 3], 1)[0]
            elecciones.append(eleccion)
            listaPto.append(ptomedio(listaPto[i], puntos[eleccion]))
        else:
            eleccion = np.random.choice(range(4), 1)[0]
            elecciones.append(eleccion)
            listaPto.append(ptomedio(listaPto[i], puntos[eleccion]))
    return listaPto

#grafica
A, B, C, D = [0, 1], [1, 1], [1, 0], [0, 0]
puntos = [A, B, C, D]
listaPto = juegoCaosCuadrado(200000, [1, 1], puntos)
(p9.ggplot(data= pd.DataFrame(listaPto, columns = ['x', 'y']), mapping=p9.
    aes(x='x', y='y')) + p9.geom_point(alpha=0.01))
```

Listing E.2: Variante 1 del Juego Caos en un Cuadrado para generar Figura 3.8b

E.3. Variante 2 del Juego del Caos en un Cuadrado, Figura 3.8c

```
import pandas as pd
import numpy as np
import plotnine as p9
'''
Funcion que obtiene el punto medio entre dos puntos.
Recibe dos puntos de la forma A = [a, b] y B = [c, d]
y devuelve un punto de la forma C = [e, f]
'''
#Sacamos el punto medio
def ptomedio(A, B):
    C = [(A[0]+B[0])/2, (A[1]+B[1])/2]
    return(C)
```

```
'''
Juego del Caos con Restriccion 2.
Si estoy en el 0 no puedo pasar 1.
Recibe como parametros:
n: numero de iteaciones.
actual: punto de inicio de la forma [a, b]
puntos: es una lista con los puntos del poligono
Regresa: una lista de puntos lista para graficar
'''
def juegoCaos2(n, actual, puntos):
    #Juego del caos
    aux2 = np.random.choice(4 , 1)
    listaPto = [actual, ptomedio(actual, puntos[aux2[0]])]

    for i in range(n):
        if puntos[aux2[0]] == puntos[0]:
            #Si estoy en 0 no pueso ir a 1.
            aux2 = np.random.choice([0, 2, 3], 1)
        else:
            aux2 = np.random.choice([0, 1, 2, 3], 1)
        listaPto.append(ptomedio(puntos[aux2[0]], listaPto[-1]))
    return listaPto

A, B, C, D = [0, 1], [1, 1], [1, 0], [0, 0]
puntos = [A, B, C, D]
listapto = juegoCaos2(150000, [1, 1], puntos)
(p9.ggplot(data= pd.DataFrame(listapto, columns = ['x', 'y']), mapping=p9.
    aes(x='x', y='y')) + p9.geom_point(alpha=0.02))
```

Listing E.3: Variante 2 del Juego Caos Cuadrado para generar la Figura 3.8c

E.4. Variante 3 del Juego del Caos en un Cuadrado, Figura 3.8d

```
import pandas as pd
import numpy as np
import plotnine as p9
'''
Funcion que obtiene el punto medio entre dos puntos.
Recibe dos puntos de la forma A = [a, b] y B = [c, d]
y devuelve un punto de la forma C = [e, f]
'''
#Sacamos el punto medio
```

```
def ptomedio(A, B):
    C = [(A[0]+B[0])/2, (A[1]+B[1])/2]
    return(C)
'''
Juego del Caos con Restriccion 3.
Si estoy en 3 no puedo pasar 1 y si estoy en 2 no puedo pasar a 0.
Recibe como parametros:
n: numero de iteaciones.
actual: punto de inicio de la forma [a, b]
puntos: es una lista con los puntos del poligono
Regresa: una lista de puntos lista para graficar
'''

def juegoCaos3(n, actual, puntos):
    #Se inicializa
    aux2 = np.random.choice(4 , 1)
    listaPto = [actual, ptomedio(actual, puntos[aux2[0]])]

    for i in range(n):
        #Si estoy en 3 no puedo ir a 1.
        if puntos[aux2[0]] == puntos[3]:
            aux2 = np.random.choice([0, 2, 3], 1)
        #Si estoy en 4 no puedo ir a 0
        elif puntos[aux2[0]] == puntos[2]:
            aux2 = np.random.choice([1, 2, 3], 1)
        else:
            aux2 = np.random.choice([0, 1, 2, 3], 1)
        listaPto.append(ptomedio(puntos[aux2[0]], listaPto[-1]))
    return listaPto

A, B, C, D = [0, 1], [1, 1], [1, 0], [0, 0]
puntos = [A, B, C, D]
listapto = juegoCaos3(150000, [1, 1], puntos)
#Grafica
(p9.ggplot(data= pd.DataFrame(listapto, columns = ['x', 'y']), mapping=p9.
    aes(x='x', y='y')) + p9.geom_point(alpha=0.02))
```

Listing E.4: Variante 3 del Juego Caos Cuadrado para generar la Figura 3.8d

E.5. Funciones Auxiliares para la Extracción de CD's y Graficar

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
import numpy as np
import plotnine as p9
import Bio
from Bio import SeqIO
from Bio import Entrez
import re
from Bio.Seq import Seq
import math
import scipy.stats as ss
from timeit import timeit
import math
from Bio import SeqIO
from matplotlib.colors import LinearSegmentedColormap
import mpl_scatter_density # adds projection='scatter_density'


'''
Funcion que recibe la ubicacion de un archivo con terminacion .gbbf y
   devuelve un dataFrame con el inicio y fin y el marco de lectura de
   region codificante
'''

def obtenGen(archivo):
    #lista donde se guaradaran el incio y fin de cada CD
    cds_list_plus = []
    cds_list_minus = []
    intergenic_records = []
    aux = 0
    #variable donde se guardara el genoma
    genoma = ''
    #Se abre cada secuencia que se encuentre en el archivo
    for seq_record in SeqIO.parse(archivo, "genbank"):

        #Pegamos cada secuencia para juntar el genoma
        secuencia = (seq_record.seq)
        genoma = genoma + str(secuencia)

        rgene_length=1

        # Ciclo sobre cada secuencia del genoma, obtiene las
            caracteristicas

        for feature in seq_record.features:
            # Nos fijamos unicamente en los CDS
```

```
            if feature.type == "CDS":
                #Ciclamos sobre cada parte (por si hay mas de un fragmento
                    como en los exones)
                for f in feature.location.parts:
                    mystart = f._start.position
                    myend = f._end.position
                if feature.strand == -1:
                    cds_list_minus.append((mystart, myend, -1))
                elif feature.strand == 1:
                    cds_list_plus.append((mystart, myend, 1))
                else:
                    sys.stderr.write(
                        "No strand indicated %d-%d. Assuming +\n" % (
                            mystart, myend)
                    )
                    cds_list_plus.append((mystart, myend, 1))

        df = pd.DataFrame(cds_list_plus + cds_list_minus, columns = ['
            inicio','final', 'tipo'])
        return genoma, df.sort_values('inicio')

'''
Funcion que obtiene el punto medio entre dos puntos.
Recibe dos listas de la forma [x0, y0], [x1, y1] y devuelve otra lista [x,
    y]
'''
def ptomedio(A, B):
    C = [(A[0]+B[0])/2, (A[1]+B[1])/2]
    return(C)

'''
Funcion que junta los genes para utilizarlos en el juego del Caos.
Recibe genoma el cual es una cadena y genes que es un dataFrame con la
    posicion de los genes.
Devuelve una cadena con los genes
'''
def genesGenBank(genoma, genes):
    secuencia = ''
    for i in range(len(genes)):
        secuencia = secuencia + genoma[genes.iloc[i, 0]:genes.iloc[i,1]]
    return secuencia

'''
Funcion que ejecuta el juego del caos con los distintos cuadros a trabajar.
Recibe una secuencia, una cadena.
```

```
'''
def juegoCaos(secuencia, actual, tipoCuadro):
    if tipoCuadro == 1:
        T = [0, 0]
        C = [1, 0]
        A = [1, 1]
        G = [0, 1]
    elif tipoCuadro == 2:
        T = [0, 1]
        C = [0, 0]
        A = [1, 1]
        G = [1, 0]
    elif tipoCuadro == 3:
        T = [0, 1]
        C = [1, 0]
        A = [1, 1]
        G = [0, 0]
    else:
        print('Tipo de cuadro incorrecto')

    #Juego del caos
    #print(secuencia, 'Aqui')
    n =len(secuencia)
    aux = actual
    x, y = [actual[0]], [actual[1]]

    for i in range(n):
        if secuencia[i] == 'C':
            aux = ptomedio(C, aux)
        elif secuencia[i] == 'T':
            aux = ptomedio(T, aux)
        elif secuencia[i] == 'A':
            aux = ptomedio(A, aux)
        elif secuencia[i] == 'G':
            aux = ptomedio(G, aux)
        x.append(aux[0])
        y.append(aux[1])
    return x,y

# "Viridis-like" colormap with white background
white_viridis = LinearSegmentedColormap.from_list('white_viridis', [
    (0, '#ffffff'),
    (1e-20, '#440053'),
    (0.2, '#404388'),
    (0.4, '#2a788e'),
```

```
    (0.6, '#21a784'),
    (0.8, '#78d151'),
    (1, '#fde624'),
], N=256)
#Funcion que hace la grafica a color
def using_mpl_scatter_density(fig, x, y, vmin, vmax):
    ax = fig.add_subplot(1, 1, 1, projection='scatter_density')
    density = ax.scatter_density(x, y, cmap=white_viridis, vmin=vmin, vmax=
        vmax)
    fig.colorbar(density, label='Numero de puntos por Pixel')

'''
Funcion que aplica en juego del caos y grafica una secuencia de nucleotidos
    .
Recibe como parametros:
- tipoGraf : 1 si se quiere la grafica a color y 2 si se quiere en blanco y
    negro.
- secuencia : cadena con nucleotidos
- tipoCuadro : 1, 2, 3 de acuerdo a como se presentaron en el capitulo 3
- nombre : nombre del genoma para guardar las imagenes
- vmin, vmax: numeros para hacer la grafica de densidad.
- alpha : transparencia de la grafica en blanco y negro.
No devuelve nada en la carpeta, donde se tenga guardado el archivo se
    guardaran las imagenes creadas
'''
def grafica(tipoGraf, secuencia, tipoCuadro, nombre, vmin = 0, vmax = 150,
    alpa = 0.01):
    x, y = juegoCaos(secuencia, [1/2, 1/2], tipoCuadro)
    #Grafica a color
    if tipoGraf == 1:
        fig = plt.figure()
        using_mpl_scatter_density(fig, x, y, vmin, vmax)
        plt.savefig(nombre)

    #Grafica en blanco y negro
    elif tipoGraf == 2:
        myplot = (p9.ggplot(data= pd.DataFrame(x, y), mapping=p9.aes(x='x',
            y='y')) + p9.geom_point(alpha= alpa))
        # convierte el output de plotnine a un objeto de matplotlib
        my_plt_version = myplot.draw()
        plt.savefig(nombre)
```

Listing E.5: Funciones auxiliares para la extracción de CD's y graficar

E.5.1. Lectura de Archivos .gbbf y Graficación

```
import os
import genesDeGenBank as GB

def main(archivo):
    genoma, genes = GB.obtenGen(archivo)
    secuenciaGenes = GB.genesGenBank(genoma, genes)

    pn = 0.004
    n = int((len(genoma)*pn)/100)
    pm = 0.004
    m = int((len(secuenciaGenes)*pm)/100)

    #Se crea un archivo con informacion relevante del genoma de los CDs
    f = open(archivo+'info.txt','w')
    f.write('Largo Genoma: '+ str(len(genoma))+ ', CDS: '+ str(len(
        secuenciaGenes)) + '\n' +
    'Prop CDs: '+ str(len(secuenciaGenes)/len(genoma))+ '\n' +
    'Proporciones usadas para graficar: \n' +
    'genoma: ' + str(n) + ', usando el ' + str(pn) + '/100 %' + '\n' +
    'CDs: ' + str(m) + ', usando el ' + str(pm) + '/100 %' + '\n'
    )
    f.close()

    #GRAFICAMOS EL GENOMA
    GB.grafica(1, genoma, 1, archivo + 'C1G1Genoma.jpg', vmin = 0, vmax = n
        , alpa = 0.01)
    #GB.grafica(2, genoma, 1, archivo + 'C1G2Genoma.jpg', vmin = 0, vmax =
        n, alpa = 0.01)

    GB.grafica(1, genoma, 2, archivo + 'C2G1Genoma.jpg', vmin = 0, vmax = n
        , alpa = 0.01)
    #GB.grafica(2, genoma, 2, archivo + 'C2G2Genoma.jpg', vmin = 0, vmax =
        n, alpa = 0.01)

    GB.grafica(1, genoma, 3, archivo + 'C3G1Genoma.jpg', vmin = 0, vmax = n
        , alpa = 0.01)
    #GB.grafica(2, genoma, 3, archivo + 'C3G2Genoma.jpg', vmin = 0, vmax =
        n, alpa = 0.01)


    #GRAFICAMOS LOS GENES
    GB.grafica(1, secuenciaGenes, 1, archivo + 'C1G1Genes.jpg', vmin = 0,
        vmax = m, alpa = 0.01)
```

```
    #GB.grafica(2, secuenciaGenes, 1, archivo + 'C1G2Genes.jpg', vmin = 0,
        vmax = m, alpa = 0.01)#

    GB.grafica(1, secuenciaGenes, 2, archivo + 'C2G1Genes.jpg', vmin = 0,
        vmax = m, alpa = 0.01)
    #GB.grafica(2, secuenciaGenes, 2, archivo + 'C2G2Genes.jpg', vmin = 0,
        vmax = m, alpa = 0.01)

    GB.grafica(1, secuenciaGenes, 3, archivo + 'C3G1Genes.jpg', vmin = 0,
        vmax = m, alpa = 0.01)
    #GB.grafica(2, secuenciaGenes, 3, archivo + 'C3G2Genes.jpg', vmin = 0,
        vmax = m, alpa = 0.01)

main('BrassicaOleracea.gbff')
```

Listing E.6: Lectura de archivos .gbbf y Graficación

Referencias Bibliográficas

[1] Alexey A. Tuzhilint. *Lectures on Hausdorff and Gromov–Hausdorff Distance Geometry*. 1ra Edición. Peking University, 2019, pág. 108. DOI: https://doi.org/10.48550/arXiv.2012.00756.

[2] Portal Académico. *REINO ANIMALIA*. 2015. URL: https://portalacademico.cch.unam.mx/biologia2/caracteristicas-generales-dominios-y-reinos/reino-animalia (visitado 11-12-2020).

[3] Saylor Academy. *19.1 Nucleotides*. 2012. URL: https://saylordotorg.github.io/text_the-basics-of-general-organic-and-biological-chemistry/s22-01-nucleotides.html (visitado 12-08-2020).

[4] Edwin Alfonso Adame Sarmiento. *Sistemas de funciones iteradas y los fractales*. 2005. URL: http://www.konradlorenz.edu.co/images/stories/suma_digital_matematicas/SFI%5C%20y%5C%20los%5C%20Fractales.pdf (visitado 11-12-2020).

[5] Ciencias Ambientales. *La primera granja de energía solar y de algas marinas en el Mar del Norte*. 2021. URL: https://www.cienciasambientales.com/es/noticias-ambientales/la-primera-granja-de-energia-solar-y-de-algas-marinas-en-el-mar-del-norte-19800 (visitado 09-06-2021).

[6] Bellet Antonio. *La Estructura del ADN, los genes y el código genético*. 2016. URL: https://www.chilebio.cl/el-adn-los-genes-y-el-codigo-genetico/amp/ (visitado 12-08-2020).

[7] Steve Axford. *Hongos: habitantes de otro reino*. 2019. URL: https://www.nationalgeographic.com.es/naturaleza/grandes-reportajes/hongos-habitantes-de-otro-reino-2_9068/9 (visitado 12-08-2020).

[8] Michael Barnsley. *Fractals Everywhere*. pub-ACADEMIC:adr: pub-ACADEMIC, 1988. ISBN: 0-12-079062-9.

[9] biotechmind. *Transcripción del ADN. La molécula de ARN / DNA transcription. RNA molecule*. 2014. URL: https://biotechmind.wordpress.com/page/4/ (visitado 15-03-2022).

[10] Instituto Nacional del Cáncer. *Metilación*. 2000. URL: https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/metilacion (visitado 22-06-2022).

[11] Sergio Casas Yrurzum, Ignacio Pareja Montoro y Manuel Pérez Aixendri. *Proyecciones y Sistemas de Representación*. 2018. URL: http://ocw.uv.es/ingenieria-y-arquitectura/expresion-grafica/eg_tema_8.pdf (visitado 11-12-2020).

[12] Almudena M. Castro. *La forma fractal de mapas y pulmones*. 2018. URL: https://culturacientifica.com/2018/10/11/la-forma-fractal-de-mapas-y-pulmones/ (visitado 10-12-2020).

[13] Carlos A. Coello Coello y col. «El misterio de los Introns». En: *Revista Digital Universitaria* 2 (2001). URL: http://www.revista.unam.mx/vol.2/num3/art1/index.html.

[14] A.J. Crilly y col. *Fractals and Chaos*. Graduate Texts in Mathematics; 122. Springer New York, 1991. ISBN: 9783540973621. URL: https://books.google.com.mx/books?id=PFfkCvhjqIMC.

[15] A. Daniel. *Esta Hermosa Planta De 144 Años Es Simplemente Espectacular*. 2019. URL: https://www.recreoviral.com/naturaleza/planta-wisteria-de-144-anos/.

[16] Robert Devaney. *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. 1ra Edición. Addison-Wesley Publishing Company Inc., 1948.

[17] DGRU. «Estándar para Datos de Secuencias de ADN (GenBank)». En: *Dirección General de Repositorios Universitarios. SDI-UNAM* (2020).

[18] Vasileios Drakopoulos y Nikolaos Nikolaou. «Efficient Computation of the Hutchinson Metric Between Digitized Images». En: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 13 (ene. de 2005), págs. 1581-8. DOI: 10.1109/TIP.2004.837550.

[19] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications - Second Edition*. 2.ª ed. 2003.

[20] Luisa Fernández Fernández y col. *Biología y Geología 4 ESO*. McGraw Hill, 2020.

[21] Jennifer Fogaça. «Nomenclatura de Compuestos Orgánicos. Química Orgánica I.» En: *Universidad Nacional de La Plata.* (2014), pág. 18. URL: https://www.frontiersin.org/article/10.3389/fgene.2014.00162.

[22] Rodríguez Gema. *Los ácidos nucleícos*. 2018. URL: http://biologiadivertidaenclase.blogspot.com/2018/09/los-acidos-nucleicos.html (visitado 12-08-2020).

[23] National Geographic. *Hongos Bioluminiscentes*. 2019. URL: https://www.nationalgeographic.com.es/fotografia/visiones-de-la-tierra/hongos-bioluminiscentes_8165 (visitado 09-06-2021).

[24] John E. Gilbert. *GEOMETRIC TRANSFORMATIONS*. 2015. URL: https://web.ma.utexas.edu/users/gilbert/M340L/LA05GeometricTrans.pdf (visitado 11-12-2020).

[25] Roderic Guigó y col. *An Assessment of Gene Prediction Accuracy in Large DNA Sequences*. 2000. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC310940/ (visitado 12-08-2020).

[26] Grupo de Invocación sobre la Docencia en Diversidad Biológica. *El Código Genético*. 2014. URL: http://www.innovabiologia.com/biodiversidad/diversidad-animal/el-codigo-genetico (visitado 12-08-2020).

[27] H.Joel Jeffrey. «Chaos game representation of gene structure». En: *Nucleic Acids Research* 18.8 (abr. de 1990), págs. 2163-2170. ISSN: 0305-1048. DOI: 10.1093/nar/18.8.2163. eprint: https://academic.oup.com/nar/article-pdf/18/8/2163/7059915/18-8-2163.pdf. URL: https://doi.org/10.1093/nar/18.8.2163.

[28] H.Joel Jeffrey. «Chaos game visualization of sequences». En: *Computers & Graphics* 16.1 (1992), págs. 25-33. ISSN: 0097-8493. DOI: https://doi.org/10.1016/0097-8493(92)90067-6. URL: https://www.sciencedirect.com/science/article/pii/0097849392900676.

[29] Eiko Jones. *Fresh water art images*. 2019. URL: https://www.eikojones.com/freshwater-life/.

[30] Myjak Józef y Tomasz Szarek. «Attractors of iterated function systems and Markov operators». En: *Abstract and Applied Analysis* 2003 (abr. de 2003). DOI: 10.1155/S1085337503212033.

[31] Ruiza Julio. *Estructura y Regulación de expresión génica*. 2018. URL: https://www.slideserve.com/vivien/area-ciencias-sector-biolog-a-nivel-iv-medio-diferenciado (visitado 12-08-2020).

[32] Jefferson Edwin King Dávalos y Hector Méndez Lango. *Sistemas dinámicos Discretos*. Primera Edición. México: Las Prensas de Ciencias, 2014.

[33] O. Knill. *LINEAR TRANSFORMATIONS IN GEOMETRY*. 2018. URL: http://abel.math.harvard.edu/~knill///teaching/math21b2018/05-geometry.pdf (visitado 11-12-2020).

[34] H. Lodish. *Biologıa celular y molecular*. Médica Panamericana, 2005. ISBN: 9789500613743. URL: https://books.google.com.mx/books?id=YdyMSxY2LjMC.

[35] B. B. Mandelbrot. *The fractal geometry of nature*. 3.ª ed. New York: W. H. Freeman y Comp., 1983.

[36] Benoit Mandelbrot. «How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension». En: *Science* 156.3775 (1967), págs. 636-638. DOI: 10.1126/science.156.3775.636. eprint: http://www.sciencemag.org/content/156/3775/636.full.pdf. URL: http://www.sciencemag.org/content/156/3775/636.abstract.

[37] Elliott Margulies. *Anticodón*. 2018. URL: https://www.genome.gov/es/genetics-glossary/Anticodon (visitado 12-08-2020).

[38] Clapp Mónica. *Introducción al Análisis Real*. Primera Edición. México: Papiros. Instituto de Matemáticas, UNAM, 2010. URL: http://lya.fciencias.unam.mx/tamariz/notas/notasam2011-1.pdf.

[39] Paul Nicklen. *Orcas, estrategias de caza*. 2015. URL: https://www.nationalgeographic.com.es/naturaleza/grandes-reportajes/orcas-estrategias-de-caza-2_9453.

[40] NOTIMEX. *Identifican bacteria que protege contra salmonella*. 2019. URL: https://www.elsiglocoahuila.mx/coahuila/noticia/281767.identifican-bacteria-que-protege-contra-salmonella.html.

[41] José Ortiz Bejar. *Simulación Cualitativa sobre diagramas de bifurcación*. Primera Edición. México: Universidad Michoacana de San Nicolás de Hidalgo, 2008.

[44] Julián Pérez Porto y Ana Gardey. *Definición de Ribosoma*. 2016. URL: https://definicion.de/ribosomas/ (visitado 12-08-2020).

[45] Mauricio Rodríguez Dorantes y col. «Metilación del ADN: un fenómeno epigenético de importancia médica». es. En: *Revista de investigación clínica* 56 (feb. de 2004), págs. 56-71. ISSN: 0034-8376. URL: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0034-83762004000100010&nrm=iso.

[46] Yair Román Tizapa, Javier G. Mendieta e Isaí Cantor Jimón. «Una construcción alternativa de la curva de sierpinski». En: *Revista digital — Matemática, Educación e Internet*. 18 (2018). URL: https://tecdigital.tec.ac.cr/revistamatematica/ARTICULOS_V18_N2_2018/RevistaDigital_Mendieta_V18_n2_2018/RevistaDigital_Mendieta_V18_n2_2018.pdf (visitado 12-08-2020).

[47] Medicina y Salud Pública. *Qué es la ameba çome cerebros& cómo puedes evitar que te contagie*. 2019. URL: https://medicinaysaludpublica.com/noticias/general/que-es-la-ameba-come-cerebros-y-como-puedes-evitar-que-te-contagie/4509 (visitado 09-06-2021).

[48] Santiago Sánchez-Migallón Jiménez. *La costa de Gran Bretaña*. 2012. URL: https://vonneumannmachine.wordpress.com/2012/11/28/la-costa-de-gran-bretana/ (visitado 16-08-2020).

[49] ScholarsArk. *Comparison, Functions, Structure, Reactivity and Key Differences*. 2019. URL: https://scholarsark.com/question/i-need-insights-into-rna-vs-dna-comparison-functions-structure-reactivity-and-key-differences/.

[51] Patricia Sieber, Matthias Platzer y Stefan Schuster. «The Definition of Open Reading Frame Revisited». En: *Trends in Genetics* 34.3 (2018), págs. 167-170. ISSN: 0168-9525. DOI: https://doi.org/10.1016/j.tig.2017.12.009. URL: https://www.sciencedirect.com/science/article/pii/S0168952517302299.

[52] Sophimania. *Comienzan a nacer más elefantes sin colmillos debido al tráfico de marfil*. 2016. URL: https://sophimania.pe/medio-ambiente/naturaleza-y-animales/comienzan-a-nacer-mas-elefantes-sin-colmillos-debido-al-trafico-de-marfil/.

[53] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.

[54] Visionlearning. *ADN II: La estructura del ADN*. 2009. URL: https://www.visionlearning.com/es/library/Biologia/2/ADN-II/160 (visitado 12-08-2020).

[55] vlab.amrita.edu. *Finding ORF of a Given Sequence*. 2012. URL: vlab.amrita.edu/?sub=3&brch=273&sim=1432&cnt=1 (visitado 11-07-2020).

[56] Jin Xiong. *Essential Bioinformatics*. Cambridge University Press, 2006. DOI: 10.1017/CBO9780511806087.

[57] FES Zaragoza. *La estructura química de las moléculas y su importancia para el estudio de los procesos biológicos*. 2014. URL: https://blogceta.zaragoza.unam.mx/biomoleculas/acidos-nucleicos/ (visitado 12-08-2020).

[58] Ana Zita. *Arqueas y bacterias*. 2019. URL: https://www.diferenciador.com/arqueas-y-bacterias/.

Bibliografía

[42] P.U.E.M.A.C. *La dimensión topológica*. 2008. URL: https://arquimedes.matem.unam.mx/PUEMAC/PUEMAC_2008/fractales/html/dimtop.html (visitado 12-08-2020).

[50] Steven J. Schrodi y col. «Genetic-based prediction of disease traits: prediction is very difficult, especially about the future†». En: *Frontiers in Genetics* 5 (2014), pág. 162. ISSN: 1664-8021. DOI: 10.3389/fgene.2014.00162. URL: https://www.frontiersin.org/article/10.3389/fgene.2014.00162.