



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Iztapalapa

APLICACIÓN DEL JUEGO DEL CAOS PARA SECUENCIAS DE AMINOÁCIDOS

Entrega: Fis. Angelina Nohemi Mendoza Tavera

Asesor: Dr. José Luis Del Río Correa

Índice

1. Introducción	5
2. Geometría Fractal	6
2.1. Medida y Dimensión de Hausdorff	7
2.2. Richardson y la longitud de las costas	12
2.3. Dimensión Fractal	16
2.4. Algoritmo de Mandelbrot o construcción de fractales por un iniciador-generator	17
2.4.1. La metáfora de la Máquina Copiadora de Reducción Múltiple (MRCM)	19
2.5. Propiedad de Autosimilaridad	22
2.5.1. Conjunto Autosimilar	23
2.6. Sistemas de funciones iteradas (IFS)	24
2.6.1. Operador de Hutchinson	29
2.6.2. Punto fijo de un IFS	31
2.6.3. Distancia entre elementos de una secuencia y su atractor	32
2.7. Sistema de Funciones Iteradas con Probabilidad (IFSP)	33
2.7.1. Máquina copiadora de reducción aleatoria (FRCM) . .	34
2.7.2. Sistema de Funciones Iteradas Recurrentes	35
3. Juego del Caos	41
3.1. El Juego del Caos de Jeffrey (CGR)	43
3.1.1. Algoritmo de Jeffrey para representación de puntos pa- ra secuencias de genómicas	44
3.1.2. Generación de regiones mediante el uso de la CGR . .	46
3.1.3. Representación gráfica de secuencias genómicas usando IFS	48
4. Cadenas de Markov	56
4.1. Condiciones para la matriz de transición	64
4.2. Clasificación de posibles estados	66
4.2.1. Probabilidades limitantes.	67
5. Revisión de algunos aspectos de Biología Molecular	72
5.1. Función y estructura de los ácidos Nucleicos	73
5.1.1. Características del ADN y ARN	74

5.1.2.	Replicación del DNA y formación del RNA nucleolar	75
5.1.3.	Tipos y funciones del ARN	77
5.2.	Síntesis de Proteínas	78
5.2.1.	Transcripción Genética	79
5.2.2.	Traducción Genética	82
6.	Extensión del Juego del Caos para Proteínas	84
6.1.	Algoritmo CGR de Proteínas	87
6.2.	Conteo de rejilla de la CGR de Proteínas	90
6.3.	Aplicación del método CGR de 12 vértices	91
7.	Fractales estrictamente autosimilares compuestos por polígonos estelares	93
7.1.	Algoritmo Tzanov	94
7.1.1.	Parámetros P y m	95
7.2.	Generación de fractales usando IFS	97
7.3.	Dimensión fractal	101
7.3.1.	Algoritmo Tzanov para Secuencias Proteicas	102
8.	Conclusiones	105
9.	Perspectivas a futuro	106
10.	Apéndice A	107
10.1.	Dimensión del conjunto de Cantor	107
11.	Apéndice B	109
11.1.	Dimensión del conjunto de Koch	109
12.	Apéndice C	110
12.1.	Imágenes Fractales aplicando el Operador de Hutchinson	110
13.	Apéndice D	112
13.1.	CGR de Familias de Proteínas	112
14.	Referencias	114

Dedicado a mi familia, gracias por nunca perder la fe en mi.

Resumen

El desarrollo de este trabajo comienza con una breve descripción de las ideas de la geometría fractal, donde se explica de forma breve las ideas principales, las cuales dan origen a una geometría que describe mas realistamente formas presentes en la naturaleza, ya que “las nubes no son esferas, las montañas no son conos, ni las costas son círculos” (Mandeldrot, 1982). Los trabajos del matemático Felix Hausdorff y del físico Lewis Fry Richardson, ayudaron al matemático Benoît Mandelbrot a crear una nueva geometría llamada Geometría Fractal. La sección 2 habla sobre los orígenes y la construcción de esta geometría, ademas se describen conceptos importantes como el Operador de Hutchinson, dimensión fractal y auto similaridad, conceptos que son importantes para la obtención de imágenes fractales.

En la sección 3 estudiamos el algoritmo del Juego del Caos y como este algoritmo puede ser usado para observar gráficamente como secuencias aparentemente aleatorios como lo es el ADN presentan patrones muy bien definidos, siendo estas imágenes fractales a las cuales llamaremos firmas genómicas. Deben tenerse algunas consideraciones importantes como el punto semilla, el tablero de juego y la localización de puntos en las imágenes resultantes.

Posteriormente estudiamos brevemente las cadenas de Markov y la relación que estas presentan con las secuencias de ADN, y como las propiedades de las cadenas Markovianas pueden ser aplicadas al ADN, también son estudiadas las matrices de transición y algunos ejemplos que ilustran este tipo de cadenas.

En la sección 5 se exponen algunos aspectos biológicos importantes como lo es la síntesis de proteínas la cual fue el parteaguas para la investigación de la sección 6 en la cual se estudia la exención del Juego del Caos para secuencias proteicas, en la cual se presentan algunas limitaciones, siendo dos de ellas que que las regiones dentro de las firmas genómicas no son homogéneas y por otro lado la superposición de puntos en la misma, motivo por el cual en la sección 7 se presenta un nuevo algoritmo el cual es usado para generar fractales estrictamente autosimilares. Posteriormente obtenemos algunas imágenes de secuencias de proteínas las cuales son comparadas con las mismas imágenes de secuencias de proteínas obtenidas en la sección 6.

1. Introducción

La geometría fractal es una de las ramas de las matemáticas más conectada con la naturaleza ya que esta puede reproducir algunos fenómenos de la misma, tal es el caso de las secuencias genómicas y de aminoácidos. La geometría fractal nos permite visualizar patrones subyacentes en dichas secuencias. Es en 1990 que H. Joel Jeffrey presenta una técnica mediante la cual es posible representar imágenes complejas mediante la iteración de transformaciones contractivas (sistemas de funciones iteradas), transformando así una secuencia de letras (ADN) en una secuencia de puntos las cuales se encontrarán dentro de un cuadrado unitario dando como resultado una imagen única para cada secuencia, esta será la huella digital de la misma. Las firmas genómicas son obtenidas mediante un mapeo, el cual transforma una secuencia genómica en una distribución de puntos los cuales pueden presentar regiones de saturación o escasez de puntos. Son estas regiones las que caracterizan a las secuencias y con las cuales se puede hacer un estudio de las mismas.

Después de la publicación del trabajo de Jeffrey fue publicado el primer trabajo en el cual se proponía una generalización del método anterior para investigar regularidades y patrones en la estructura primaria de proteínas, otra de las motivaciones para este nuevo campo de investigación es para probar métodos de predicción de estructuras (Fiser et al., 1994), pero este método presenta algunas limitaciones como lo es la superposición de puntos, es por esto que en este trabajo es usado un algoritmo con el cual es posible obtener fractales estrictamente autosimilares (Tzanov, 2015) usando un factor de contracción característico, de tal forma que la secuencia de letras de las secuencias proteicas son codificadas a una secuencia de puntos los cuales vivirán dentro de un polígono regular de 12 lados, elegido de esta forma ya que fueron agrupados los aminoácidos dependiendo de su grado de hidrofobicidad. De esta forma es posible obtener firmas genómicas es las que los puntos no se sobreponen, lo cual ayudará a darle una caracterización estadística apropiada.

2. Geometría Fractal

Cuando nos detenemos a observar la naturaleza podemos observar que gran parte de los elementos que la conforman son objetos poco uniformes, por lo que puede surgir una pregunta: ¿Estos objetos poco simétricos pueden medirse mediante métodos convencionales?, es decir ¿Se mide igual la longitud de una recta que la longitud de una costa? Esta es una pregunta que es resuelta mediante la Geometría Fractal. Esta es una rama de las matemáticas que se considera relativamente joven ya que en 1982 el matemático polaco Benoît Mandelbrot acuña el término “Fractal” el cual proviene del latín (*fractus*) y significa fraccionado o irregular. La característica más poderosa de esta geometría es su capacidad de explicar, modelar y reproducir muchos fenómenos presentes en la naturaleza, los cuales no pueden ser estudiados por la Geometría Euclidiana.

Para llegar a esta poderosa forma de ver las cosas fueron necesarias tres mentes poderosas las cuales se desarrollaron en diferentes campos de la ciencia, por un lado se tenía un desarrollo sobre medidas en todas las dimensiones posibles y por otro lado una investigación práctica sobre la medición de costas, hasta ese momento eran dos trabajos sin relación aparente, pero años después un matemático prodigioso encontraría la relación entre estos dos trabajos, culminando en una nueva formulación de la Geometría.

En esta sección se describirán brevemente los desarrollos que realizaron Felix Hausdorff, Lewis Fry Richardson y Benoît Mandelbrot y la generación de fractales matemáticos.

A pesar de que el término “Fractal” nace en 1982 es mucho tiempo atrás que nacen figuras geométricas peculiares para la época, las cuales fueron marginadas por los estudiosos de la época, una de estas creaciones fue desarrollada en el siglo *XIX* por el matemático ruso Georg Ferdinand Ludwig Philipp Cantor el cual cuestionó el concepto de dimensión. En matemáticas la dimensión de un espacio se define como el número de coordenadas que se necesitan para determinar la posición de un punto, esto llevó a Cantor a un extraño tipo de geometría inspirada en la naturaleza como puede observarse en el conjunto que lleva su nombre.

Contemporáneos a este conjunto nacieron otras curvas que presentan características peculiares las cuales eran extrañas para la época, una de estas curvas fue creada por el matemático Giuseppe Peano al exponer sus ideas sobre

curvas continuas que recubrían todo el plano siendo un **conjunto denso** del plano, este tipo de curvas se obtendría mediante una secuencia de curvas continuas sin intersecciones las cuales convergen a una curva limite, mientras que en 1904 Helge Von Koch publica su artículo titulado “Acerca de una curva continua que no posee tangentes y obtenida por los métodos de la geometría elemental” presentando así la llamada curva de Koch la cual es una curva cerrada continua en todos sus puntos pero no diferenciable en ningún punto. Debido a las irregularidades y peculiaridades de estas curvas es que fue imposible clasificarlas de tal forma que fueron consideradas como “monstruos matemáticos” sin utilidad, pero estos serían ejemplos tempranos de lo que se conocerían como fractales.

2.1. Medida y Dimensión de Hausdorff

El problema de medición de longitudes de objetos irregulares (como los que se encuentran en la naturaleza) evidenció la necesidad de redefinir en una forma más precisa los conceptos de medición asociados a un objeto y la dimensión del mismo.

Imaginemos que se desea medir el largo de una curva, una primera aproximación para medir esta curva se obtiene al reemplazar dicha curva por una poligonal como se muestra en la Figura 1, para esto se utiliza una escala y se toma un punto en la curva de tal forma que manteniendo una abertura fija en un compás se recorrerá todo el perímetro de la curva sustituyendo así la curva original por un polígono de N lados. Para obtener una primera aproximación de la longitud se tomará ε como la longitud de cada uno de los lados del polígono, teniendo así que la longitud se obtendrá con el número de lados de la poligonal y el tamaño de las mismas, es decir $N \cdot \varepsilon$. Para hacer entonces una mejor aproximación de la medición se tendría que dibujar un polígono con mayor número de lados, los cuales serán más cortos que el de la poligonal anterior, por lo que el número de lados dependerá de la longitud de los mismos, de modo que la aproximación para la longitud de la curva se puede escribir como

$$L = N(\varepsilon) \cdot \varepsilon \tag{1}$$



Figura 1: Poligonal de lado ε .

Dado que esto es solo una aproximación podemos definir la medida de la longitud de la curva mediante el límite a que tiende la expresión (1) cuando el número de lados tiende a infinito o dicho de otra forma cuando la longitud de los lados del polígono tiende a cero, siendo esta última la definición que usaremos. En lugar de longitud de la curva definiremos M_1 como la medida de un conjunto que sabemos que es unidimensional en el ejemplo que estamos analizando de una curva, escribiremos entonces la siguiente expresión.

$$M_1 = \lim_{\varepsilon \rightarrow 0} N(\varepsilon) \cdot \varepsilon \quad (2)$$

En forma análoga cuando tratamos de determinar el área de una región irregular, lo que se haría es usar una cuadrícula con cuadros de área ε^2 y entonces contar cuántos de estos cuadros están dentro de la región de interés, de modo que la medida del área estará dada por la siguiente expresión:

$$M_1 = \lim_{\varepsilon \rightarrow 0} N(\varepsilon) \cdot \varepsilon^2 \quad (3)$$

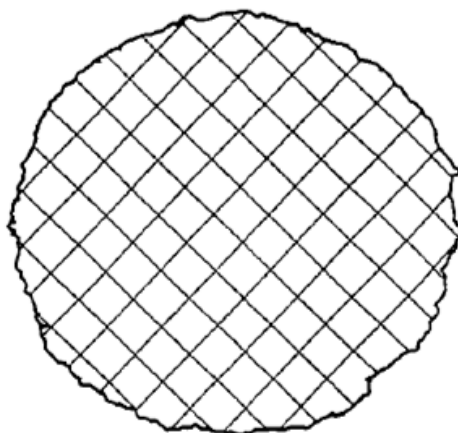


Figura 2: Cuadrícula de una superficie irregular.

Donde $N(\varepsilon)$ es el número de cuadros de arista ε como se muestra en la Figura 2. Del mismo modo si tenemos un volumen se usará una red tridimensional de cubos de volumen ε^3 como se muestra en la Figura 3, de tal forma que la medida de volumen estará definida como se muestra en la siguiente expresión.

$$M_1 = \lim_{\varepsilon \rightarrow 0} N(\varepsilon) \cdot \varepsilon^3 \quad (4)$$

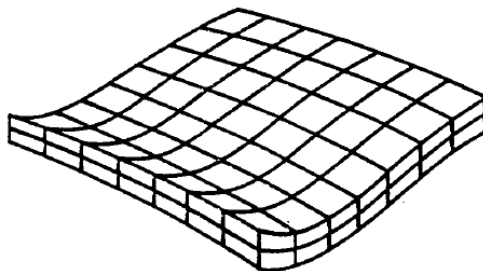


Figura 3: Cuadrícula para un volumen.

Las expresiones (2), (3) y (4) constituyen la base de nuestras mediciones usuales de perímetro, área y volumen. En general estamos acostumbrados a pensar que las dimensiones son números enteros pero son matemáticos notables como Hausdorff y Besicovich los que se mostraron interesados por refinar

la idea de magnitud, proponiendo así diversas generalizaciones de la idea de **dimensión**. El principal problema de los conjuntos irregulares radicaba en que las medidas utilizadas hasta entonces eran incapaces de plasmar su tamaño. Una de estas medidas era la medida de Lebesgue la cual se utilizaba normalmente para medir conjuntos en R^n y se asocia con el concepto de dimensión topológica, la cual nos permite hacernos una idea del tamaño que tiene un conjunto y poder así compararlo con otros conjuntos. Es hasta 1919 que el matemático alemán Felix Hausdorff desarrolló una generalización métrica del concepto de dimensión de un espacio topológico introduciendo la notación de medida de un conjunto en todas las dimensiones posibles, partiendo de la idea utilizada para llegar a las expresiones (2), (3) y (4). Dado un conjunto M la medida de Hausdorff de un conjunto ($H_\alpha(M)$), donde α es la dimensión del conjunto, estará dada por el número mínimo de esferas de diámetro ε que adosan al conjunto como se muestra en la Figura 4, esto se expresa en la ecuación (5).

$$M_1 = \lim_{\varepsilon \rightarrow 0} N(\varepsilon) \cdot \varepsilon^3 \quad (5)$$

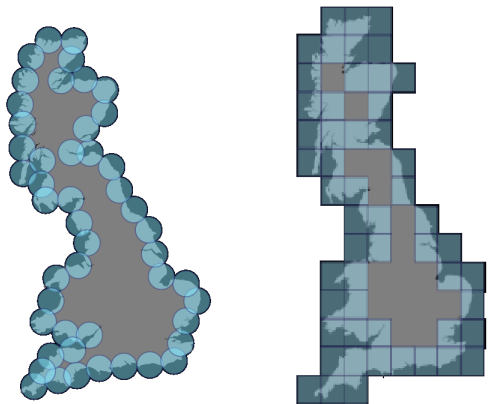


Figura 4: Tipos de adoquines para la costa de Inglaterra.

Es necesario encontrar una relación entre el número de esferas (o cualquiera que sea la figura geométrica que adosque al conjunto) y el diámetro de estas al momento de adosar un conjunto u objeto. Cada conjunto tendrá una medida específica que depende de la dimensión α del conjunto. Algunos

ejemplos se muestran a continuación:

1. Objetos unidimensionales como una línea o el intervalo unitario, se tiene:

$$\begin{aligned} N(\varepsilon) \cdot \varepsilon &= L \\ N(\varepsilon) &= L \cdot \varepsilon^{-1} \end{aligned}$$

2. Objetos bidimensionales:

$$\begin{aligned} N(\varepsilon) \cdot \varepsilon^2 &= A \\ N(\varepsilon) &= A \cdot \varepsilon^{-2} \end{aligned}$$

3. Objetos tridimensionales:

$$\begin{aligned} N(\varepsilon) \cdot \varepsilon^3 &= V \\ N(\varepsilon) &= V \cdot \varepsilon^{-3} \end{aligned}$$

Se observa que $N(\varepsilon)$ cumple una ley de potencias donde el exponente es la dimensión del objeto, al generalizar se esperaría que $N(\varepsilon)$ para valores pequeños de ε , se comporte como:

$$N(\varepsilon) = A \left(\frac{1}{\varepsilon}\right)^D \tag{6}$$

Donde D es una cantidad característica del conjunto M .

Una vez encontrada la relación del número de esferas necesarias para adoquinar un objeto es posible encontrar la expresión de la medida de Hausdorff, para esto se sustituye la expresión (6) en la expresión (5), lo cual implica la siguiente ecuación:

$$H_\alpha(M) = A \lim_{\varepsilon \rightarrow 0} \varepsilon^{\alpha-D} = \begin{cases} \infty & \text{si } \alpha < D \\ A & \text{si } \alpha = D \\ 0 & \text{si } \alpha > D \end{cases} \tag{7}$$

La expresión (7) implica que existe una dimensión característica del conjunto M para la cual la medida de Hausdorff proporciona un valor diferente

de cero o infinito (Figura 5), donde $H_\alpha(M)$ es nula para dimensiones mayores que la **Dimensión de Hausdorff** (D) y es infinita para dimensiones menores que D , es decir, si se mide un área con esferas la dimensión de Hausdorff será cero y si se mide un área con líneas la dimensión de Hausdorff será infinita.

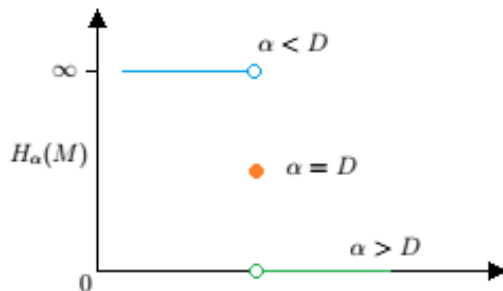


Figura 5: Dimensión de Hausdorff.

2.2. Richardson y la longitud de las costas

Históricamente muchos países han entrado en conflictos debido a la medición de fronteras, la discrepancia que cada lado de las fronteras reportaba se presentaba como una causa culminante en muchos problemas territoriales. Fue hasta 1961 que se comisionó al físico Lewis Fry Richardson para determinar la longitud de la costa oeste de Inglaterra, para esto Richardson tomó un mapa a escala y procedió a medir la costa usando el *Método de Arquímedes*, este consiste en aproximar la costa mediante una poligonal a escala ε_1 con lados iguales y contar el número de lados $N(\varepsilon)_1$ de la poligonal, obteniendo así la longitud de la costa a escala ε_1 , como se observa en la expresión (1) donde $L(\varepsilon)_1$ es la longitud de la costa.

$$L(\varepsilon)_1 = N(\varepsilon)_1 \varepsilon_1 \tag{8}$$

Pero esta cantidad es sólo una estimación de la longitud de la costa, para obtener una mejor aproximación Richardson construyó poligonales cada vez más pequeñas tomando escalas más finas, obteniendo así una secuencia de longitudes $\{L(\varepsilon_1), L(\varepsilon_2), \dots, L(\varepsilon_n)\}$ con $\varepsilon_n < \varepsilon_{n-1} < \dots < \varepsilon_2 < \varepsilon_1$ observando así que el valor de la longitud aumentaba conforme se reducía la escala

de la poligonal, es decir, la longitud no convergía a un valor finito si no que crecía con tendencia al infinito como se describe en 9.

$$L = \lim_{\varepsilon \rightarrow 0} N(\varepsilon)\varepsilon = \infty \quad (9)$$

De este modo Richardson observó que la longitud dependía de la escala con la que se midiera un objeto, esta relación de dependencia fue obtenida al graficar el logaritmo de la longitud total contra el logaritmo de la longitud del lado de la poligonal obteniendo así una relación lineal:

$$\ln(L) = -|m|\ln\varepsilon + \ln b \quad (10)$$

Reescribiendo la expresión anterior tenemos:

$$L = b\varepsilon^{-|m|} \quad (11)$$

donde L es la longitud de la costa y ε es la longitud de un lado de la poligonal o escala.

La expresión (11) muestra claramente que se satisface una ley de potencias, así que al usar esta expresión se pueden comparar dos segmentos de una costa para un mismo valor de ε aun cuando la longitud de cada una sea infinita, es decir:

$$L_1(\varepsilon) = b_1\varepsilon^{-|m|} \quad L_2(\varepsilon) = b_2\varepsilon^{-|m|} \quad (12)$$

Comparando las dos secciones de la costa se tendrá:

$$\frac{L_2(\varepsilon)}{L_1(\varepsilon)} = \frac{b_2}{b_1} \quad (13)$$

donde esta razón no dependerá de la longitud de la poligonal. Por lo tanto la extensión de ciertas partes de la costa pueden ser comparadas, no por sus longitudes si no por sus coeficientes . Así, encontró que la forma de medir la extensión de la costa por el método de Arquímedes la cual se utiliza para curvas lisas no puede aplicarse para medir las costas de Inglaterra, por lo que Richardson investigó en enciclopedias y almanaques la longitud de las fronteras terrestres entre diferentes países y para la sorpresa de Richardson

llego a la conclusión que los países limítrofes habían medido con diferentes escalas la longitud de sus fronteras, como se observa en la figura 6.

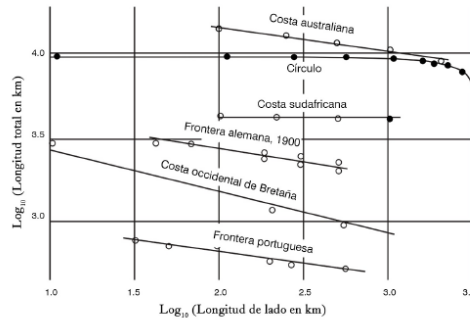


Figura 6: Resultados de Richardson.

Dado un objeto las cubiertas que pueden usarse para cubrir al objeto pueden ser esferas o cuadrados, para el caso de objetos planos será más simple cubrirlos con cuadrados, si tomamos la Costa de Gran Bretaña y la cubrimos con cuadrículas de diferentes tamaños como se muestra en la figura (7) y queremos medir su longitud como se planteó en el problema de Richardson para cada cuadrícula que dibujemos sobre la costa tendremos un número $N(\varepsilon_1)$ de cuadrados que se necesitarán para cubrir la costa, al tomar cuadrículas cada vez más finas se encuentra un arreglo de la forma:

$$\begin{pmatrix} \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_q \\ N(\varepsilon_1) & N(\varepsilon_2) & \cdots & N(\varepsilon_q) \end{pmatrix}$$

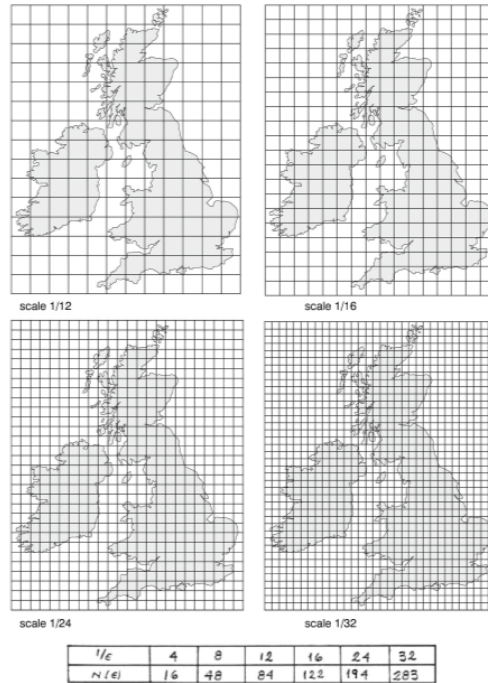


Figura 7: Diferentes cuadrículas aplicadas al mapa de la Costa de Gran Bretaña.

Siguiendo entonces el procedimiento que realizó Richardson se debe proceder a graficar los puntos de coordenadas:

$$(-\ln \epsilon_r \quad , \quad \ln N(\epsilon_r)) \tag{14}$$

A estos puntos se les ajustará una recta usando el método de mínimos cuadrados, así que la pendiente de la recta será la dimensión de Hausdorff y su ordenada al origen es la medida de Hausdorff de la costa en dimensión D , esto puede verse en la Figura (8).

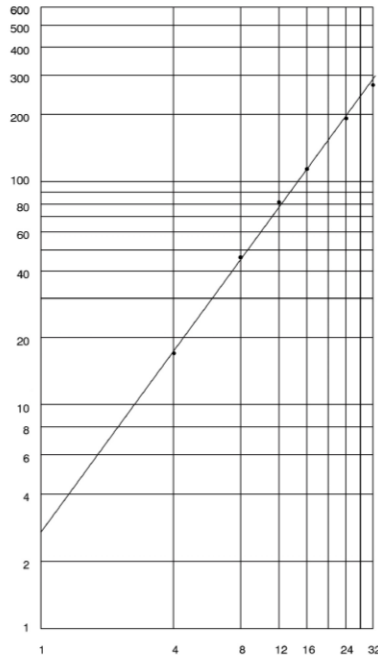


Figura 8: Grafica log-log de los resultados mostrados en la Figura 7.

2.3. Dimensión Fractal

Los resultados de Richardson y Hausdorff aparecen en áreas distintas del conocimiento, pero es en 1967 cuando Benoit Mandelbrot encuentra la conexión entre estos trabajos, empezando por la presencia de leyes de potencias en ambos trabajos, por otro lado Mandelbrot descubrió que el extraño comportamiento del que hablan Richardson y Hausdorff es habitual en la naturaleza. Para ahondar más en la conexión entre los dos trabajos partamos por observar que en el caso de Richardson se presenta una dependencia entre la longitud y la escala, mientras que en el caso de Hausdorff la ley de potencias gobernará el número mínimo de esferas que se requieren para adoquinar efectivamente un conjunto, como se muestra en el siguiente recuadro.

Richardson	Hausdorff
Relación entre la escala y el número de segmentos de la poligonal óptima: $L(\varepsilon) = b\varepsilon^{- m }$	Número de esferas para adoquinar: $N(\varepsilon) = A\varepsilon^{- D }$
Longitud: $L = \lim_{\varepsilon \rightarrow 0} N(\varepsilon)\varepsilon$	Medida: $M_\alpha(M) = \lim_{\varepsilon \rightarrow 0} N(\varepsilon)\varepsilon^\alpha$

Podemos observar que la longitud es la medida de Hausdorff en dimensión 1, de tal forma que esto nos permite establecer una relación entre la pendiente de la recta trazada de la gráfica $\log - \log$ de Richardson y la dimensión de Hausdorff.

Es decir Richardson encuentra que la longitud de la costa tiende a infinito cuando $\varepsilon \rightarrow 0$ siguiendo una ley de potencias:

$$N(\varepsilon) = b\varepsilon^{-|m|-1} \quad (15)$$

En la cual es posible identificar el número de lados de la poligonal (15) con el número óptimo de esferas para adoquinar la costa, es decir de la ley de potencias de Hausdorff se obtiene:

$$b\varepsilon^{-|m|-1} = A\varepsilon^{-|D|} \quad (16)$$

donde $A = b$ por lo tanto $D = 1 + |m|$, siendo esta la *Dimensión de la costa*. De esta forma las ideas de Hausdorff nos permiten entender los resultados de Richardson, mientras el trabajo de Richardson nos permite encontrar un algoritmo para encontrar la dimensión de Hausdorff de un objeto.

2.4. Algoritmo de Mandelbrot o construcción de fractales por un iniciador-generador

Hasta ahora hemos hablado de curvas que presentan características peculiares por las cuales fueron llamadas monstruos matemáticos, pero ahora es necesaria una definición que nos permita agrupar todas las características que presentan estas curvas o conjuntos fractales. Describir todas las características que debe tener un conjunto fractal es muy complicado, el matemático británico Kenneth Falconer afirma que en lugar de dar una definición de un

fractal se debe considerar que un fractal debe cumplir ciertas condiciones, las cuales pueden o no cumplirse para otros conjuntos fractales ya que no todos los fractales cumplen todas las condiciones. Un fractal debe considerarse como un conjunto con ciertas propiedades características, algunas de estas propiedades son las siguientes:

1. Estructuras finas: Esto significa que no importa cuántas veces amplie-mos una parte de la imagen fractal, siempre se obtendrá una nueva imagen que mostrará más detalles.
2. Autosimilaridad: En algún sentido los conjuntos son autosimilares, en la naturaleza aparecen conjuntos fractales que no son exactamente iguales en sí mismos pero muestran una autosimilaridad estadística, es decir que una ampliación de una pequeña parte del fractal mostrará propiedades estadísticas similares a la de todo el fractal pero no exactamente lo mismo.
3. Dimensión topológica menor que su dimensión de Hausdorff.
4. Definición algorítmica recursiva sencilla.
5. Son conjuntos muy irregulares para ser descritos en lenguajes geométricos tradicionales.

Para comprender mejor estos conjuntos fractales y algunas de sus propiedades es necesario conocer los procesos dinámicos que los crean. Uno de los hechos más sorprendentes de la geometría fractal y la teoría del caos es que en presencia de un patrón complejo es muy probable que un proceso muy simple se encuentre detrás. El algoritmo que se usará para generar un fractal puede comenzar con un pedazo de recta, un triángulo, un cuadrado o alguna otra figura L , este segmento inicial se conoce como el *iniciador*, después se toma una regla (ya sea de reducción o traslación) la cual es aplicada al iniciador, de tal forma que el iniciador se transformará completando así la etapa de construcción base conocida como *generador*, una vez completada esta etapa se aplicará nuevamente al objeto tantas veces como se desee. Para observar gráficamente como se genera un fractal veamos un pequeño ejemplo: Tomaremos la recta unitaria $[0, 1]$ como nuestro iniciador, el cual se dividirá en tres partes iguales sustituyendo la parte central con dos segmentos de igual tamaño que los segmentos restantes. Los segmentos que se

agregarán al centro formarán un triángulo equilátero. Este proceso se repetirá en cada segmento resultante variando la escala en cada repetición, en este caso la escala hará referencia a la longitud de los segmentos. El resultado de este proceso se observa en la Figura 9, dos propiedades interesantes de esta curva son que la longitud entre dos puntos cualesquiera es infinita y que aunque esta curva tenga una longitud infinita tiene un área cero. Esta curva fue creada por el matemático Helge Von Koch.

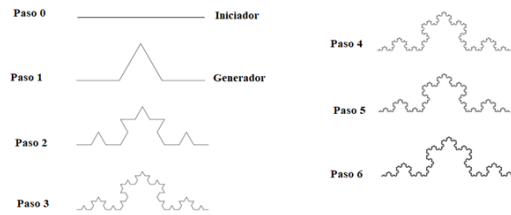


Figura 9: Construcción de la curva de Koch.

La curva de Koch es un ejemplo de una curva sin rectas tangentes que está conformada de esquinas en todas partes de tal forma que no se puede adaptar una recta tangente a ninguno de los puntos. Si se realiza un algoritmo de este estilo sucesivamente para cada uno de los conjuntos resultantes se construirá paso a paso un objeto fractal, este tipo de algoritmos estarán relacionados con la metáfora de la Máquina Copiadora de Reducción Múltiple la cual se discutirá en la siguiente sección.

2.4.1. La metáfora de la Máquina Copiadora de Reducción Múltiple (MRCM)

La geometría fractal proporciona un medio para descomponer los patrones y formas de la naturaleza en elementos primitivos, de esta forma es posible visualizar a la naturaleza en una forma peculiar, la cual tiene su propio lenguaje donde sus elementos son las *Transformaciones Primitivas* y sus palabras son *Algoritmos Primitivos*, estas son representadas por la **Metáfora de la Máquina Copiadora de Reducción Múltiple (MRCM)**. La metáfora plantea una copiadora que tiene un sistema de lentes independientes, cada uno de los cuales reducen una imagen de entrada y la colocan en un lugar en específico que caracteriza al sistema de lentes que se esté usando,

generando así un ensamble con todas las copias reducidas que producirá una imagen final como salida para una MRCM. Para obtener una imagen final determinada usando la MRCM se deben de especificar las siguientes condiciones:

1. Número de lentes del sistema.
2. Ajuste del factor de reducción para cada sistema de lentes individualmente.
3. Configuración del sistema de lentes para el montaje de las copias.

Cabe mencionar que la idea crucial de esta máquina es que funcione en un *Ciclo de retroalimentación*, es decir su propia salida se retroalimenta como su nueva entrada una y otra vez, de tal forma que una copia de una máquina de este tipo revela todas las características geométricas de la máquina, veamos un ejemplo para comprender como funcionan estas máquinas.

Como fue mencionado anteriormente se necesita una imagen inicial la cual se muestra en la Figura 10 (A), ahora debemos especificar el sistema de lentes, este reducirá $\frac{1}{3}$ la imagen de entrada y la copiará 3 veces, después colocará cada copia en los vértices de un triángulo equilátero como se observa en la Figura 10 (B), esta será ahora la imagen de entrada del sistema de lentes de la MRCM, de tal forma que ahora se reducirá $\frac{1}{3}$ el ensamble mostrado en (B) y se copiará tres veces colocando cada copia en las esquinas del mismo triángulo equilátero en el que se colocaron las primeras copias (Figura 10 (C)), este proceso se repetirá tantas veces como se desee, como aparece en la Figura 10 (E). Si esta máquina se repite un gran numero veces el resultado será una imagen conocida como el famoso Triángulo de Sierpinski (Figura 10 (F)).

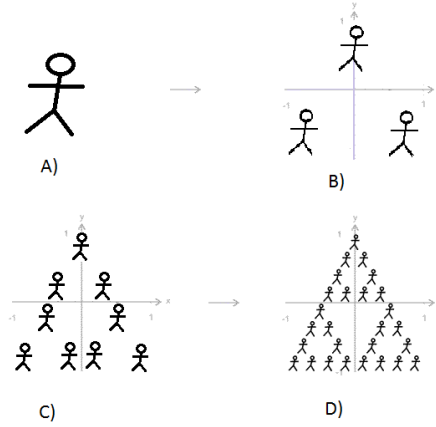


Figura 10: Ejemplo de una Máquina Copiadora de Reducción Múltiple con un sistema de tres lentes.

La estructura principal de la imagen final de una MRCM se observa en la primera copia que produce la máquina, esta copia es un collage de las imágenes transformadas por los lentes que conforman la MRCM (Figura 10 (B)). Al aplicar la MRCM a la imagen iniciadora también se determina la calidad de la aproximación, cuando la distancia de la copia al objetivo es pequeña entonces sabemos por el *Principio de mapeo de contracción* que la imagen final no está lejos de la imagen generadora.

La parte fundamental de los MRCM será el sistema de lentes, ya que no importa la imagen inicial que se le introduzca a la máquina siempre se obtendrá una secuencia de imágenes que tienden hacia una misma imagen final llamada *atractor* la cual caracteriza a esa máquina copiadora de reducción múltiple. Una característica peculiar de estas máquinas es que si se comienza el proceso con el atractor este no cambiará, con esto se dice que el atractor se deja invariante ante la MRCM. Esto puede quedar más claro si lo comparamos con un experimento: imaginemos que tenemos un tazón y dentro de este hay una pequeña pelota la cual se puede colocar en diferentes posiciones iniciales de donde puede ser soltada, la pelota por consiguiente descansará en el fondo del tazón sin importar de donde parta, pero si se empieza colocando la pelota justo en la parte inferior del tazón esta no se moverá (Peitgen, 2002). En esta analogía el tazón corresponderá a la MRCM y las posiciones iniciales de la pelota corresponden a las imágenes iniciales que se le introducen a la

máquina, por otro lado observar el camino de la pelota a cada tiempo corresponde a ejecutar la máquina repetidamente y el punto de descanso de la pelota corresponde a la imagen resultante en esa repetición, mientras que el fondo del tazón representará el atractor de la máquina. Podríamos preguntarnos si es posible tener más de un atractor y esto pasaría si tuviéramos un tazón con más de un fondo pensando en el ejemplo anterior.

Algunos resultados hechos por Felix Hausdorff y Stefan Banach demostraron que “cualquier MRCM siempre tiene una imagen final única, es decir un único atractor” no importa la combinación de los sistemas de lentes de la MRCM una vez que el sistema de lentes contraiga las imágenes. Esta máquina proporciona una introducción a lo que se conoce como sistemas de funciones iteradas (IFS) los cuales están representados por el sistema de lentes de la MRCM, cada lente representará a una Transformación Lineal Afín. En resumen la MRCM es un arreglo de sistemas de lentes que contraen imágenes, la cual genera un sistema dinámico ya que la máquina se ejecuta en un entorno de retroalimentación que conduce a una secuencia de imágenes $A_0, A_1, A_2, \dots, A_\infty$, donde A_0 es una imagen inicial arbitraria. La secuencia de imágenes conducirá a una imagen final A_∞ que será independiente de la imagen inicial A_0 , si se elige A_∞ como imagen inicial la MRCM se dejará a la imagen A_∞ invariante, es decir no se modificará la imagen bajo las transformaciones lineales. Podemos decir así que A_∞ es un *punto fijo* del IFS o dicho de otra forma, A_∞ es el atractor del sistema dinámico (IFS).

2.5. Propiedad de Autosimilaridad

En el ejemplo de la curva de Koch pudimos observar que si contáramos con un lente de buena calidad y observáramos con él una pequeña parte del conjunto, esta parte sería congruente¹ con la imagen global. Recordemos que dos de las características principales de un conjunto fractal son que dichos conjuntos tienen detalles en todas las escalas y que presenten autosimilaridad. Un conjunto F en R^n es autosimilar cuando F está compuesto de N copias idénticas a él, las cuales son el resultado de transformaciones contractivas con factores de escala r_1, r_2, \dots, r_q . Cuando $r_1 = r_2 = \dots = r_q = r$ se dice que F es autosimilar con respecto a la razón r y al entero q , de tal forma

¹Se dice que dos conjuntos son congruentes ($F \sim G$), si son idénticos, excepto por desplazamientos o rotaciones, es decir será semejante consigo mismo o *autosemejante*.

que la aplicación de esta transformación al subconjunto F de $H(x)$ define un nuevo conjunto $r(F) = r(x) : x \in F$. [9] Esta propiedad de autosimilaridad se observa en la curva de Koch (Figura 9) la cual es el resultado de un proceso iterativo donde cada copia del generador está escalada, para la curva de Koch el factor de contracción es de $r = \frac{1}{3}$. Podemos entonces definir a un objeto autosimilar de la siguiente forma:

$$F = \cup_{i=1}^q F_i | F_i \sim r_i(F) \quad \text{donde} \quad F_i \cap F_j \neq \emptyset \quad (17)$$

Hasta el momento hemos analizado el caso en que los factores de contracción son iguales, donde la ecuación de dimensión de autosimilaridad se aplica ya que todas las copias tienen la misma escala, pero estos factores pueden ser diferentes; Los conjuntos autosimilares fueron estudiados por P. A. P. Moran en 1946, Moran estudió aquellos conjuntos fractales cuyos elementos están escalados por diferentes cantidades o razones, lo que lo llevó a desarrollar una ecuación mediante la cual fue posible calcular la dimensión de autosimilaridad de objetos fractales autosimilares más generales. La ecuación de Moran está escrita de la siguiente forma: $1 = r_1^d + \dots + r_N^d$, donde cada uno de los r_i satisface que $0 < r_i < 1$. La ecuación de Moran tiene una solución única y dicha solución es la dimensión de autosimilaridad $d = d_s$.

Este tipo de autosimilaridad no será la única, en la naturaleza podemos encontrar objetos como las nubes, montañas, plantas, el cuerpo humano, etc., los cuales no presentan una autosimilaridad geométrica o exacta (con todos los factores de escala iguales). Existe otro tipo de autosimilaridad, esta es la autosimilaridad estadística la cual se presenta en la naturaleza, estos conjuntos están compuesto por N copias escaladas de sí mismo y es idéntica en todos los momentos estadísticos.

2.5.1. Conjunto Autosimilar

Un conjunto acotado S , es autosimilar con respecto a la razón r y a un entero n cuando:

$$S = S_1 \cup S_2 \cup \dots \cup S_n \quad \text{con} \quad S_i \cap S_k = \emptyset \quad (18)$$

Donde S_k es congruente con el conjunto $r(S)$ el cual se obtiene al aplicar

una compresión r al conjunto S , es decir $r(S) = \{C(x)|x \in S\}$. Dos conjuntos son congruentes cuando ambos son idénticos excepto por rotaciones o traslaciones, de tal manera que los conjuntos congruentes S_k se pueden escribir de la siguiente manera:

$$S_k = \mathbf{R} \cdot \mathbf{C}(S) = \omega_k(S) = \mathbf{A}(S) + \mathbf{b} \quad (19)$$

La matriz \mathbf{A} es el producto de las matrices de compresión \mathbf{C} por la matriz de rotación \mathbf{R} y el vector \mathbf{b} se obtiene de la matriz de traslación, esto nos permite obtener las transformaciones afines ω_k asociadas con el subconjunto S_k .

2.6. Sistemas de funciones iteradas (IFS)

Un sistema de funciones Iteradas (IFS) está formado por un conjunto de n transformaciones contractivas:

$$\mathbf{W} = (\omega_1(x), \dots, \omega_n(x)) \quad \text{tal que} \quad |\omega_i(y) - \omega_i(x)| < r_i|x - y|$$

donde $x \in \mathbf{R}^n$ y la distancia entre dos puntos $d = |x - y|$, así:

$$\mathbf{W} = \cup_{i=1}^P \omega_i(x); \quad \text{genera } P \text{ puntos}$$

$$\mathbf{W}^2(x) = \mathbf{W} \circ \mathbf{W}(x) = \mathbf{W} \circ \cup_{i=1}^P \omega_i(x) = \cup_{j=1}^P \cup_{i=1}^P \omega_j(x)\omega_i(x); \quad \text{genera } P^2 \text{ puntos}$$

Y entonces $\mathbf{W}^n(x)$ genera P^n puntos, para transformaciones contractivas:

$$F = \text{Lim}_{n \rightarrow \infty} \mathbf{W}^n(x)$$

Donde F es un conjunto fractal que no depende del punto inicial y es único para un IFS contractivo.

Un sistema de funciones iteradas (IFS) es un conjunto de mapeos lineales, ya que estos tienen la propiedad de transformar las rectas en rectas, estas transformaciones están expresadas de la siguiente forma:

$$\omega(\mathbf{r}) = \mathbf{A} \cdot \mathbf{r} + \mathbf{b} \quad (20)$$

Donde \mathbf{r} es el vector de posición de un punto en el plano, \mathbf{A} es una matriz 2×2 con elementos a_{ij} y \mathbf{b} es un vector de traslación con componentes b_i . Los sistemas de funciones iteradas están conformados por transformaciones afines, las cuales pueden verse como un sistema dinámico en el plano, como se muestra en la siguiente expresión:

$$\begin{aligned} x_{n+1} &= a_{11}x_n + a_{12}y_n + b_1 \\ y_{n+1} &= a_{21}x_n + a_{22}y_n + b_2 \end{aligned} \quad (21)$$

Esta transformación es un mapeo que transforma a un punto (x_n, y_n) del plano en otro punto (x_{n+1}, y_{n+1}) . Estas transformaciones modificarán la distancia entre los objetos a los cuales se les aplique, es decir, dados dos puntos P y Q la distancia entre ellos cambiará al aplicarles la expresión (20) de la siguiente forma:

$$\|\omega(P) - \omega(Q)\| \leq s \cdot \|P - Q\| \quad (22)$$

El número s el cual satisface la desigualdad anterior se conoce como *Constante de Lipschitz* para W , esta constante es necesaria para la clasificación de las transformaciones afines. Hay tres tipos de transformaciones lineales afines, estas son: *Contractiva* para $s < 1$, de *simetría* para $s = 1$ y *Expansivas* para $s > 1$. La constante de Lipschitz para las transformaciones (21) estará dada por la siguiente expresión:

$$s = \sqrt{a_{11}a_{22} - a_{12}a_{21}} \quad (23)$$

Nosotros estaremos interesados en las transformaciones lineales afines contractivas para la generación de fractales, en el ejemplo de la metáfora

de la MRCM las transformaciones lineales afines contractivas están representadas por los lentes de la máquina, de tal forma que el sistema de lentes de la maquina sería un IFS. Los fractales como hemos mencionado anteriormente están presentes en la naturaleza como en montañas, costas, nubes, plantas, etc., estos pueden ser generados mediante la iteración de una o más transformaciones afines. Un IFS es una transformación recursiva del tipo:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (24)$$

Para obtener una imagen final deseada se pueden usar varias transformaciones afines, este método se conoce como un *sistema de funciones iteradas (IFS)*, un ejemplo puede verse en la siguiente figura.

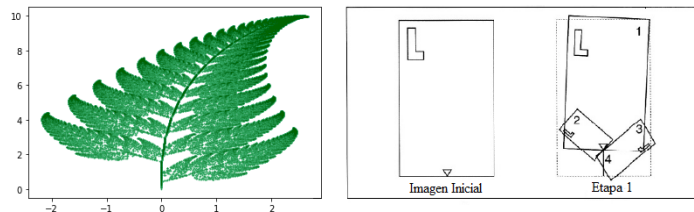


Figura 11: Helecho de Barnsley.

Para encontrar el conjunto atractor de \mathbf{W} usaremos la siguiente definición:

Definición 1

Si el conjunto atractor es el mismo para todos los puntos en una región C , se dice que C es la cuenca de atracción del atractor A :

$$C = \{r : A = \lim_{m \rightarrow \infty} W^m(r)\}$$

Pero ¿qué pasará cuando el IFS se aplica a un conjunto de puntos, en particular cuando se aplica al intervalo unitario?

Ejemplo 2.1: El conjunto de Cantor

Consideremos la transformación de conjunto valuado (se le denomina así ya que a cada punto r se le asocian dos puntos que se obtienen al aplicar ω_1

y ω_2 al vector r) formado por las siguientes transformaciones:

$$\begin{aligned}\omega_1(x) &= \frac{x}{3} \\ \omega_2(x) &= \frac{x}{3} + \frac{2}{3}\end{aligned}\tag{25}$$

Estas transformaciones afines están definidas en el intervalo $[0, 1]$. Expresemos todos los valores del intervalo $[0, 1]$ en base 3 para poder realizar un análisis de la función de conjunto valuado $\mathbf{W} = \{\omega_1, \omega_2\}$, entonces tenemos:

$$x_{base3} = 0.b_1b_2b_3 \cdots = \sum_{i=1}^N \frac{b_i}{3^i}\tag{26}$$

Donde b_i solo podrá tomar valores 0, 1 y 2, así al aplicar cada transformación tendremos:

$$\left(\frac{x}{3}\right)_{base3} = 0.0b_1b_2b_3 \cdots\tag{27}$$

$$\left(\frac{x}{3} + \frac{2}{3}\right)_{base3} = 0.2b_1b_2b_3 \cdots\tag{28}$$

De esta forma podemos ver que al aplicar la transformación ω_1 a un número en el intervalo $[0, 1]$ se agrega un 0 después del punto, en tanto que al aplicar ω_2 se agregará un 2 después del punto, de tal forma que se correrán todas las demás cifras significativas un lugar a la derecha. Utilizando estos resultados para las primeras 4 iteraciones de la función \mathbf{W} tendremos:

$$\mathbf{W}(x) = \{0.0b_1b_2b_3 \cdots, 0.2b_1b_2b_3 \cdots\}$$

$$\mathbf{W}^2(x) = \{0.00b_1b_2b_3 \cdots, 0.02b_1b_2b_3 \cdots, 0.20b_1b_2b_3 \cdots, 0.22b_1b_2b_3 \cdots\}$$

$$\mathbf{W}^3(x) = \{0.000b_1b_2b_3 \cdots, 0.002b_1b_2b_3 \cdots, 0.020b_1b_2b_3 \cdots, 0.022b_1b_2b_3 \cdots, \\ 0.200b_1b_2b_3 \cdots, 0.202b_1b_2b_3 \cdots, 0.220b_1b_2b_3 \cdots, 0.222b_1b_2b_3 \cdots\}$$

$$\mathbf{W}^4(x) = \{0.0000b_1b_2b_3 \cdots, 0.0002b_1b_2b_3 \cdots, 0.0020b_1b_2b_3 \cdots, 0.0022b_1b_2b_3 \cdots, \\ 0.0200b_1b_2b_3 \cdots, 0.0202b_1b_2b_3 \cdots, 0.0220b_1b_2b_3 \cdots, 0.0222b_1b_2b_3 \cdots, \\ 0.2000b_1b_2b_3 \cdots, 0.2002b_1b_2b_3 \cdots, 0.2020b_1b_2b_3 \cdots, 0.2022b_1b_2b_3 \cdots, \\ 0.2200b_1b_2b_3 \cdots, 0.2202b_1b_2b_3 \cdots, 0.2220b_1b_2b_3 \cdots, 0.2222b_1b_2b_3 \cdots\}$$

Así después de n iteraciones se tendrá un conjunto de 2^n elementos, los cuales se obtienen al considerar todos los posibles arreglos de los primeros n dígitos significativos, a este conjunto se le conoce como el *Conjunto de Cantor*.

Una propiedad importante de estos atractores es que es independiente del punto inicial. Al realizar $\mathbf{W}(L_0) = \omega_1 \cup \omega_2(L_0)$ con L_0 el intervalo unitario lo que hará el operador \mathbf{W} es dividir a L_0 en tres partes iguales y eliminando el intervalo del centro por lo que los intervalos restantes tendrán una longitud de $\frac{1}{3}$ como se observa en la Figura 12.

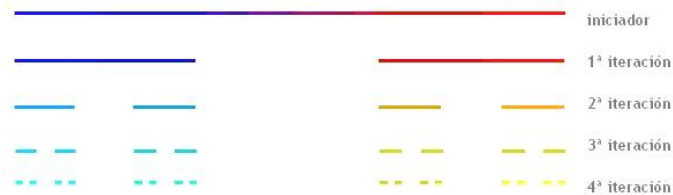


Figura 12: Conjunto de Cantor.

Una vez que se tiene el intervalo con el agujero en el centro como se ve en la primera iteración de la Figura 12, ω_2 tomará los intervalos de la primera iteración y hará dos copias reducidas de longitud $\frac{1}{3}$, colocando una copia en el origen y la segunda en $\frac{2}{3}$ (segunda iteración), así la longitud de las copias será $L_2 = \frac{L_0}{3^2}$. Al realizar cada iteración se reducirá la longitud de las copias

por un factor de $\frac{1}{3}$, por lo que para la n -ésima iteración cada segmento tendrá longitud $L_n = \frac{L_0}{3^n}$.

Los *mapeos contractivos* satisfacen la condición de *contractividad promedio de Elton*, la cual entre otras cosas garantiza que el atractor es independiente del punto inicial. Por otro lado cabe mencionar que todos los puntos del dominio de definición deforman la cuenca del atractor A , es por esto que el atractor no depende del punto inicial por que todos los puntos llevarán al atractor.

2.6.1. Operador de Hutchinson

En la sección anterior se habló acerca de las transformaciones lineales afines las cuales son el mecanismo que hace funcionar a las MRCM's, si el proceso se realiza iterativamente lo que se obtiene es la unión del conjunto de transformaciones ω_n las cuales formarán lo que es llamado el Operador de Hutchinson \mathbf{W} el cual tiene la siguiente forma:

$$\mathbf{W}(A) = \omega_1(A) \cup \omega_2(A) \cup \omega_3(A) \cup \dots \cup \omega_N(A) \quad (29)$$

Donde ω_i con $i = 1, \dots, N$ representan a las transformaciones lineales afines contractivas con factor de contracción c_i , las cuales pueden no ser iguales. Las imágenes resultantes de la aplicación de las transformaciones ω_i nos pueden llevar a preguntarnos ¿Cómo se pueden comparar imágenes? Esta es una pregunta fundamental para comprender a los Sistemas de Funciones Iteradas (IFS). Felix Hausdorff propuso un método para determinar la distancia entre imágenes la cual lleva su nombre: *Distancia de Hausdorff* $h(A, B)$, introducir esta distancia nos lleva a hablar acerca de secuencias de imágenes $A_0, A_1, A_2, \dots, A_k$ las cuales tenderán a un límite A_∞ siempre y cuando la distancia de Hausdorff $h(A_\infty, A_k) \rightarrow 0$ cuando $k \rightarrow \infty$, donde las imágenes A_k estarán cada vez más cerca unas de otras tras cada iteración del IFS.

Por otro lado, Hutchinson demostró que \mathbf{W} es una contracción con respecto a la distancia de Hausdorff por lo que el principio de mapeo de contracción se puede aplicar a la iteración del operador de Hutchinson (\mathbf{W}), como se muestra en la ecuación (30):

$$\mathbf{h}(\mathbf{W}(A), \mathbf{W}(B)) \leq c \cdot \mathbf{h}(A, B) \quad \text{con } 0 \leq c < 1 \quad (30)$$

Debido a esto al tomar cualquier imagen A_0 para iniciar la iteración del IFS la secuencia generada será:

$$A_{k+1} = \mathbf{W}(A_k), \quad \text{con } k = 0, 1, 2, 3, \dots \quad (31)$$

Una de las características fundamentales de este principio indica que la rapidez con la cual se llegará al atractor al aplicar el sistema de funciones iteradas dependerá de la *métrica* que se use ya que cuanto menor sea la relación de contracción mejor será la estimación de la velocidad de convergencia del IFS. Pero como fue mencionado anteriormente el principio de mapeo de contracción se aplicará a \mathbf{W} así que la secuencia tenderá a una imagen distinguida que será el atractor A_∞ del IFS. La siguiente imagen muestra un atractor el cual fue obtenido mediante un operador de Hutchinson, el cual se ejemplifica gráficamente en los cuadros pequeños de la derecha.²

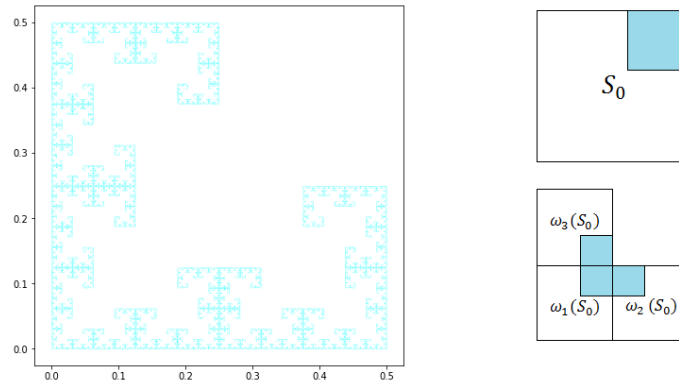


Figura 13: Ejemplo de la aplicación del Operador de Hutchinson a un cuadrado.

El principio de mapeo de contracción también tiene como característica la estimación de la velocidad con la que se llegará a la imagen límite A_∞ , pero ¿Cómo se verá esto reflejado en el operador de Hutchinson?, como se

²Para más ejemplos dirigirse al Apéndice C.

mencionó anteriormente este operador es conformado por la unión de varias transformaciones de contracción ω_i que tendrán un factor de contracción c_i el cual nos ayudará a calcular que tan lejos estamos del atractor. Lo que se estimará será la rapidez con la que el IFS producirá la imagen final (atractor) al aplicar el operador de Hutchinson. Para esto se aplicará el operador de Hutchinson una vez a la imagen A_0 ya que el factor de contracción c de \mathbf{W} es determinado por la contracción ω_i pero como son varias contracciones se tendrán c_i factores de contracción y solo necesitamos uno. Este factor de contracción será el de valor más grande de todos los factores de contracción, es decir, $c = \max\{c_i\}$ así la eficiencia del IFS está determinada por esta contracción individual.

2.6.2. Punto fijo de un IFS

Una IFS asocia a cada punto en el plano un conjunto de puntos en R^2 . Como fue mencionado anteriormente, un Fractal autosimilar puede generarse por medio de un sistema de funciones iteradas siempre y cuando las transformaciones sean contractivas y todas las regiones $\omega_i(A)$ no tengan puntos en común con cada $\omega_k(A)$, es decir, que no haya superposiciones entre las imágenes, así el atractor de la transformación \mathbf{W} será un fractal autosimilar. El conjunto atractor es la unión de todos los conjuntos $\omega_1(A), \omega_2(A), \dots, \omega_n(A)$ con ω_i los mapeos afines, cuando las n transformaciones afines contractivas se aplican a un conjunto en el plano se obtienen copias reducidas de este conjunto, obteniendo así:

$$\mathbf{W}(A) = \omega_1(A) \cup \omega_2(A) \cup \dots \cup \omega_n(A) \quad (32)$$

El conjunto atractor A_∞ de una IFS se define como:

$$A_\infty = \lim_{m \rightarrow \infty} \mathbf{W}^m(\mathbf{r}_0) \quad (33)$$

Hutchinson demostró que si se considera como el conjunto inicial el propio fractal, es decir el atractor A_∞ , entonces se cumple que el conjunto será invariante y al aplicar el operador \mathbf{W} este arrojará como resultado al conjunto A_∞ , esto se expresa en la ecuación (34).

$$\mathbf{W}(A_\infty) = \omega_1(A_\infty) \cup \omega_2(A_\infty) \cup \cdots \cup \omega_n(A_\infty) = A_\infty \quad (34)$$

2.6.3. Distancia entre elementos de una secuencia y su atractor

Dada una secuencia de elementos $a_{n+1} = \omega_k(a_n)$ en un espacio métrico completo (X) para que estos elementos conformen un conjunto autosimilar deben cumplir las siguientes propiedades:

1. Hay un atractor único $A_\infty = \lim_{n \rightarrow \infty} (a_n)$
2. A_∞ es invariante $\omega_k(A_\infty) = A_\infty$
3. Hay una estimación a priori para la distancia desde a_n al atractor, $d(a_n, A_\infty) \leq c^n \frac{d(a_0, a_1)}{(1-c)}$

En esta sección estudiaremos más a detalle la propiedad 3 de contracción de ω , para lo cual derivamos:

$$d(\omega(a_0), A_\infty) = d(\omega(a_0), \omega(A_\infty)) \leq C \cdot d(a_0, A_\infty) \quad (35)$$

Aplicando la desigualdad del triángulo se obtiene que:

$$\begin{aligned} d(a_0, A_\infty) &\leq d(a_0, \omega(a_0)) + d(\omega(a_0), A_\infty) \\ &\leq d(a_0, \omega(a_0)) + c \cdot d(a_0, A_\infty) \end{aligned} \quad (36)$$

Así:

$$d(a_0, A_\infty) \leq \frac{d(a_0, \omega(a_0))}{1 - c}$$

De la misma manera:

$$d(a_n, A_\infty) \leq \frac{d(a_n, a_{n+1})}{1 - c} \quad (37)$$

Para todo $n = 0, 1, 2, \dots$ finalmente:

$$\begin{aligned} d(a_n, a_{n+1}) &\leq C \cdot d(a_{n-1}, a_n) \\ &\leq C^2 \cdot d(a_{n-2}, a_{n-1}) \\ &\leq \dots \\ &\leq C^n \cdot d(a_0, a_1) \end{aligned} \quad (38)$$

Llegando así a la siguiente propiedad:

$$d(a_n, A_\infty) \leq \frac{c^n}{1 - c} d(a_0, a_1) \quad (39)$$

Esto nos permite predecir el número de iteraciones n que se deben hacer para acercarse al atractor.

2.7. Sistema de Funciones Iteradas con Probabilidad (IFSP)

Como se ha visto en secciones anteriores en la metáfora de la máquina copiadora de reducción múltiple los sistemas de funciones iteradas pueden aplicarse repetidas veces a un punto o imagen de entrada, pero hasta el momento no hemos hablado sobre el orden en el cual deben aplicarse las transformaciones afines ω_n a estos puntos o imágenes. Existen dos posibles casos en los cuales se pueden aplicar las transformaciones a un conjunto de entrada (o iniciador), el primero de ellos es siguiendo un orden establecido (este caso será estudiado con más detalle en la sección 3), el segundo caso

es elegir a cada transformación afín aleatoriamente, esta última será el caso que se estudiará en esta sección, para esto comencemos con una metáfora que nos ayudará a visualizar este tema.

2.7.1. Máquina copidora de reducción aleatoria (FRCM)

En la Sección 2.4.1 hablamos sobre la metáfora de la Máquina Copidora de Reducción Múltiple (MRCM) la cual es útil para construir imágenes fractales mediante algoritmos simples, estas máquinas constan de un sistema de lentes que reducirán una imagen de entrada, posteriormente copiarán la imagen y desplazaran estas copias a posiciones que estarán determinadas por la configuración de la MRCM, esto puede verse gráficamente en la Figura 14, en donde la imagen de entrada será un círculo y la configuración de la MRCM será un triángulo equilátero.

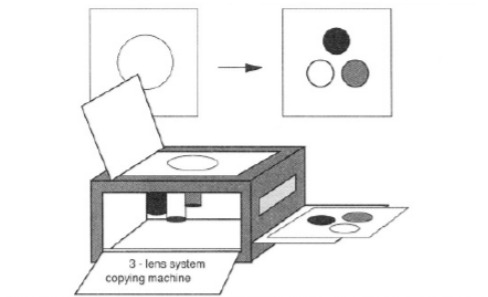


Figura 14: Representación de una MRCM con una configuración triangular (Peitgen, 2002).

Pero ahora se hará una modificación a esta metáfora en la cual para cada iteración se elegirá una transformación ω_N al azar, para esto a cada ω_N se le asignará una probabilidad, esta será una MRCM “aleatoria” o “Fortune Wheel Reduction Copy Machine (FRCM)”, en la que los puntos acumulados formarán la imagen final. Un esquema del funcionamiento de esta máquina puede verse en la Figura 15.

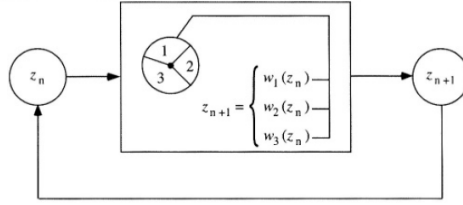


Figura 15: Esquema de la Fortune Wheel Reduction Copy Machine (FRCM) (Peitgen, 2002).

Una característica interesante de esta máquina es que la imagen final de una MRCM es decir el atractor puede ser generado también por una FRCM. Como fue mencionado en secciones anteriores una MRCM está determinada por N contracciones lineales afines $\omega_1, \omega_2, \dots, \omega_N$, las cuales generan una secuencia de imágenes donde la imagen final de la máquina está descrita por el *Operador de Hutchinson*, una FRCM está dada por las mismas transformaciones afines contractivas $\omega_1, \omega_2, \dots, \omega_N$, pero ahora a estas transformaciones se les asignarán las probabilidades $p_1, p_2, \dots, p_N > 0$. Al asignar estas probabilidades a las transformaciones se tendrá un IFS distinto al usado en secciones anteriores, el cual llamaremos sistema de funciones iteradas recurrentes.

2.7.2. Sistema de Funciones Iteradas Recurrentes

Estos sistemas fueron introducidos por M. Barnsley, J. Elton y D. Hardin[9], estos sistemas son una generalización de los sistemas de funciones iterados en los que se utiliza una Cadena de Markov ³, esta cadena es utilizada para impulsar un sistema de mapeos $\omega_i : X \rightarrow X$ con $i = 1, 2, \dots, N$ donde X es un espacio métrico completo (Barnsley, 1993), esto es descrito en la siguiente definición:

³Para más información dirigirse a la sección 4.

Definición 2

Un Sistema de funciones iteradas recurrente consiste de un IFS $\{X; \omega_1, \omega_2, \dots, \omega_N\}$ con X un espacio subyacente y una matriz $\{\mathbf{P}_{i j} \in [0, 1] : i, j = 1, 2, \dots, N\}$, la cual describe las probabilidades de transición entre estados para un proceso de Markov a tiempos discretos (Barnsley, 1993).
--

Ejemplo 2.2: Algoritmo para un IFS Recurrente

Imagine que se tiene un sistema de tres transformaciones afines $(\omega_1, \omega_2, \omega_3)$, las cuales etiquetaremos con los números 1, 2 y 3 respectivamente, a las cuales se les aplicará el siguiente procedimiento:

1. Tome un punto inicial $(x_0, y_0) \in R^2$, es recomendable que el punto de partida se encuentre lo más cerca posible del atractor, de tal forma que se elegirá $(x_0, y_0) = (0, 0)$.
2. Elija un estado inicial s_1 del conjunto 1, 2, 3, es decir elija una transformación afín de partida.
3. Asigne una probabilidad de transición $p_{s_1|s_j}$ (donde s_j será el siguiente estado del sistema) y aplique al punto inicial, de tal forma que se obtendrá un punto asociado al estado en el que se desee estar, es decir:

$$(x_1, y_1) = \omega_{s_1}(x_0, y_0)$$

4. Seleccione $s_2 \in 1, 2, 3$ con probabilidad $p_{s_2|s_j}$ y aplique al punto anterior de tal forma que se obtendrá el siguiente punto:

$$(x_2, y_2) = \omega_{s_2}(x_1, y_1)$$

5. Repetir los pasos anteriores tantas veces se desee, de tal forma que en la n -ésima iteración se tendrá $s_n \in 1, 2, 3$ con probabilidad $p_{s_n|s_j}$, obteniendo:

$$(x_n, y_n) = \omega_{s_n}(x_{n-1}, y_{n-1})$$

De tal forma que se obtendrá un conjunto de puntos que darán como resultado una imagen final que será el atractor del sistema. Las probabilidades de transición entre estados para este tipo de sistemas deberán cumplir las siguientes condiciones:

1. La suma de las probabilidades de transición desde un estado dado hacia los demás estados será:

$$\sum_{j=1}^N p_{i|j} = 1 \quad (40)$$

donde los subíndices indican en qué estado está el sistema y hacia qué estado va a transicionar respectivamente.

2. Para cualquier estado inicial i y cualquier estado final j existe una secuencia de números enteros $k, l, \dots, m \in \{1, 2, \dots, N\}$ tal que:

$$p_{i|k} \ p_{k|l} \ \cdots \ p_{m|j} > 0 \quad (41)$$

La condición 1 expresa que si el sistema se encuentra en algún estado i en la n -ésima iteración en la $(n+1)$ -ésima iteración el sistema se encontrará en algún estado dado; Por otro lado la condición 2 indica que si el sistema está en el estado i , entonces existe una probabilidad finita de llegar al estado j en un número finito de pasos para cualquier par de enteros $i, j \in \{1, 2, \dots, N\}$.⁴

En las Tablas 1 y 2 se muestran algunos ejemplos de algoritmos de IFS's recurrentes, en cada caso el espacio métrico es R^2 y las transformaciones lineales afines serán de la forma:

$$\omega_n(v_i) = \begin{pmatrix} a_n & b_n \\ c_n & d_n \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} e_n \\ f_n \end{pmatrix} = \begin{pmatrix} x_f \\ y_f \end{pmatrix} \quad (42)$$

Tabla 1. Sistema con transiciones prohibidas.

⁴Para más referencias dirigirse a la Sección 4

ω_n	a_n	b_n	c_n	d_n	e_n	f_n	$p_{n 1}$	$e_{n 2}$	$e_{n 3}$
1	0.5	0	0	0.5	0	0	0.3	0.7	0
2	0.5	0	0	0.5	0	128	0	0.6	0.4
3	0.5	0	0	0.5	128	128	0.5	0	0.5

Para el sistema descrito en la Tabla 1 algunos de los estados serán prohibidos si la transición es desde un estado en específico, por ejemplo la probabilidad de pasar del estado 2 al estado 1 es cero por lo tanto no hay transición del estado 2 al estado 1 como se observa en la Figura 16.

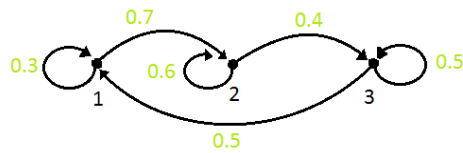


Figura 16: Esquema de transiciones posibles entre estados para el sistema descrito en la Tabla 1.

Tabla 2. Sistema sin transiciones prohibidas.

ω_n	a_n	b_n	c_n	d_n	e_n	f_n	$p_{n 1}$	$e_{n 2}$	$e_{n 3}$
1	0.5	0	0	0.5	0	0	0.3	0.6	0.1
2	0.5	0	0	0.5	0	128	0.1	0.5	0.4
3	0.5	0	0	0.5	128	128	0.4	0.4	0.2

En este sistema todas las transiciones entre estados son posibles, un esquema de estas transiciones se muestra en la Figura 17.

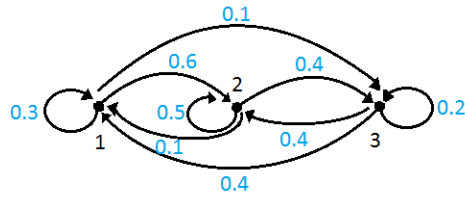


Figura 17: Esquema de transiciones entre estados correspondiente al sistema descrito en la Tabla 2.

Una característica peculiar de estos sistemas es que es posible conocer el estado en el que se encuentra el sistema en la n -ésima iteración si se conoce la última transformación que se aplicó a la última entrada (ya sea punto o imagen) del sistema y viceversa, es decir, se dice que el sistema está en el estado i si la última transformación que se aplicó a la entrada fue ω_i . En las figuras anteriores se mostró de forma gráfica como es que el sistema puede transicionar, pero ¿Cómo se vería el atractor de un IFS recurrente? La Tabla 3 muestra un sistema que consta de 4 transformaciones afines las cuales representan a los estados del sistema y sus respectivas probabilidades de transición.

Tabla 3. Ejemplo de IFS Recurrente.

ω_n	a_n	b_n	c_n	d_n	e_n	f_n
1	0.5	0	0	0.5	0	0
2	0.5	0	0	0.5	0	0.5
3	0.5	0	0	0.5	0.5	0.5
4	0.5	0	0	0.5	0.5	0

A cada transformación se le asigna una probabilidad, para este ejemplo son:

p_1	0.1
p_2	0.5
p_3	0.4
p_4	0

El resultado obtenido después de 10,000 iteraciones tomando como punto inicial $(0,0)$ será el conocido Triángulo de Sierpinski el cual se muestra en la Figura 18, este será el atractor del IFS recurrente presentado en la Tabla 3, en la figura podemos observar cómo se presentan regiones en las que hay más puntos y se ven un poco más azules y secciones en las cuales se observan muy pocos puntos, esto se debe a las probabilidades que se le asignaron a cada transformaciones.

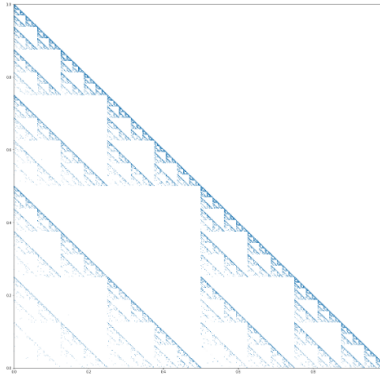


Figura 18: Triángulo de Sierpinski con probabilidades.

3. Juego del Caos

La idea que tenemos sobre la aleatoriedad en imágenes nos dice que las estructuras o patrones que se crean aleatoriamente se ven más o menos arbitrarios o tal vez pueden presentar alguna estructura pero que probablemente no sea muy interesante, pero esta afirmación no es del todo cierta. En los años ochenta del siglo pasado el matemático británico Michael Barnsley propuso un método de generación de fractales a partir de los vértices de un polígono y un punto “al azar”, esta iteración geométrica es conocida como *El Juego del Caos*, este método demuestra como la aleatoriedad puede generar imágenes con patrones muy interesantes.

El juego del caos nos ayuda a visualizar los procesos de aleatoriedad que presentan estructuras complejas, este juego tiene un algoritmo sencillo matemáticamente hablando pero muy poderoso, el cual describiremos a continuación: Imaginemos que tenemos un dado cuyas seis caras están etiquetadas con los números 1, 2 y 3, identificando 6 con 1, 5 con 2 y 4 con 3. Este dado será el generador de números aleatorios de base 3 para nuestro ejemplo. Los números aleatorios que aparecen a medida que echamos el dado (por ejemplo 2, 3, 2, 2, 1, 2, 3, 2, 3, 1, \dots) impulsarán al proceso. Dicho proceso será regido por un tablero de juego con una configuración específica para cada juego, de tal forma que es posible obtener una figura resultante característica para cada tablero de juego, en este caso debido a la base 3 de nuestro dado tomaremos como tablero a un triángulo equilátero, el cual será etiquetado con los valores 1, 2 y 3 en cada uno de sus vértices, como se observa en la Figura 19.

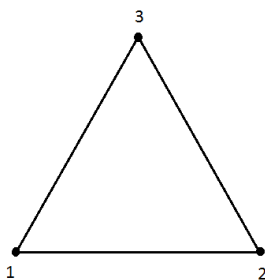


Figura 19: Tablero para el juego del caos de base 3.

El juego del caos no solo está regido por un tablero si no por un conjunto de reglas que se describen a continuación:

1. Se elegirá un punto inicial arbitrario en el tablero o cerca del tablero, denotaremos a este punto como z_0 .
2. Se tirará el dado y dependiendo de la etiqueta que salga (1, 2, o 3) se trazará una recta imaginaria del punto z_0 a la esquina que haya salido. Supongamos que sale un 2, para generar el nuevo punto z_1 trazaremos una recta imaginaria entre el punto anterior z_0 y la esquina marcada con la etiqueta 2 y se marcará el punto medio de dicha recta imaginaria.
3. Repetiremos el proceso k veces, después de k tiradas se habrá generado una secuencia de puntos z_1, z_2, \dots, z_k , de tal forma que la siguiente tirada generará el punto z_{k+1} , el cual se colocará en el punto medio entre el punto anterior z_k y la esquina aleatoria que resulte de lanzar el dado. Este proceso es visualizado en la Figura 20.

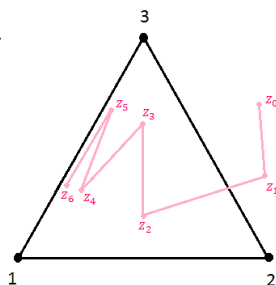


Figura 20: Representación de algunos puntos del juego de caos.

Parecería que se obtendrá solo un conjunto de puntos que no mostraran nada interesante pero esto no es así, si se realiza este proceso un buen número de veces digamos para 10,000000 iteraciones se podrá observar un patrón muy interesante como se muestra en la Figura 21, la imagen resultante es conocida como el Triángulo de Sierpinski.

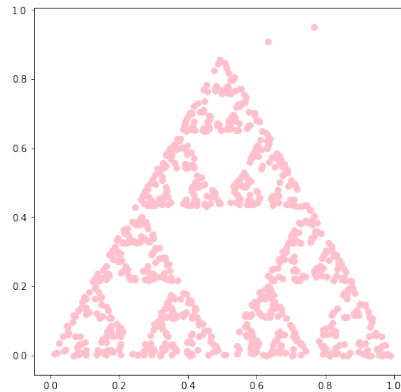


Figura 21: Imagen resultante después de 10,000000 iteraciones.

De esta manera se observa como la aleatoriedad puede crear una forma perfectamente determinista, es decir podemos observar una relación entre aleatoriedad y fractales deterministas. Como observamos al lanzar el dado no es posible saber dónde se dibujará el punto debido a que será un punto aleatorio, sin embargo, el patrón que dejará el conjunto de puntos resultantes es absolutamente predecible.

3.1. El Juego del Caos de Jeffrey (CGR)

El estudio de secuencias biológicas es una rama de investigación en crecimiento, pero el estudio de estas secuencias es sumamente complicado debido a que estas son cadenas de caracteres simbólicos conformadas por una gran cantidad de elementos, pero es en 1990 que H. Joel Jeffrey en su trabajo “Chaos Game Representation of Gene Structure” desarrolló un método que permite transformar una secuencia biológica (ADN o ARN) a una secuencia numérica mediante una extensión del juego del caos, dichas secuencias numéricas conservan las mismas características de la secuencia original, este método es conocido como *Representación del juego del caos (CGR)* o *Juego del caos de Jeffrey*. Este algoritmo está conformado por un sistema de cuatro funciones contractivas que estarán asociadas a cada base de la secuencia, de esta forma se establece una relación biunívoca entre una secuencia biológica y un conjunto de puntos dentro de un cuadrado unitario, obteniendo así un diagrama de dispersión el cual será una representación gráfica de la secuencia

genómica. Este método nos permite representar e investigar patrones presentes en una secuencia los cuales no pueden ser observados mediante métodos clásicos de estadística. Las representaciones gráficas de las secuencias genómicas serán figuras características de cada secuencia o dicho de otra forma serán la firma genómica de la secuencia, la cual muestra patrones globales de la secuencia al igual que patrones de subsecuencias. Dicha firma genómica será el “atractor” de su respectivo operador de Hutchinson.

Este método es utilizado como una herramienta para visualizar secuencias genómicas o incluso partes de secuencias. En este caso el campo de interés serán las secuencias de ADN y ARN las cuales están formadas por 4 nucleótidos (Adenina, Citosina, Guanina, Timina/Uracilo) así la base estará conformada por 4 letras $\{A, C, G, T/U\}$.

3.1.1. Algoritmo de Jeffrey para representación de puntos para secuencias de genómicas

Como fue mencionado anteriormente este método es una extensión del juego del caos de modo que se construirá un tablero de juego para la representación de secuencias genómicas (ADN y ARN), el tablero será un cuadrado unitario Q cuyos vertices estarán etiquetados con las bases del alfabeto que conforma a las secuencias genómicas cuyas coordenadas son $A(0, 0); C(0, 1); G(1, 1); T/U(1, 0)$, definiremos a los vértices presentes en la secuencia genómica como $V(x_V, y_V)$ con $V = A, C, G, T$ para secuencias de ADN, una vez establecido el tablero de juego tomaremos un punto semilla el cual estará localizado en el centro del cuadrado unitario Q , este punto inicial tendrá coordenadas $P_0 = (x_0 = \frac{1}{2}, y_0 = \frac{1}{2})$. Para determinar el siguiente punto $P_1(x_1, y_1)$ se tomará el punto medio del segmento $\overline{P_0V}$ cuyas coordenadas están dadas por:

$$\begin{aligned} x_1 &= \frac{1}{2}(x_0 + x_V) \\ y_1 &= \frac{1}{2}(y_0 + y_V) \end{aligned} \tag{43}$$

Donde la forma matricial de la expresión (43) estará definida de la siguiente manera:

$$P_1 = \omega_V(P_0) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} \frac{x_V}{2} \\ \frac{y_V}{2} \end{pmatrix} \quad (44)$$

De tal forma que los siguientes puntos serán producidos por la aplicación sucesiva del mismo proceso, esto se expresa en la ecuación (45).

$$P_n = \omega_{V_n}(P_{n-1}) \quad (45)$$

Para la representación del juego del caos se trazará un punto para cada base de la secuencia, los puntos son trazados a medio camino entre el punto previo y la esquina correspondiente a la base de cada subsecuencia sucesiva, es decir se tomarán las bases en el orden en el que aparezcan en la secuencia genómica, de esta forma la secuencia de ADN podrá expresarse como una secuencia de puntos producidos por la aplicación iterativa de (45). Es posible notar una relación entre el Algoritmo del Juego de Caos de Jeffrey y el **Mapeo de Bernoulli**, esta relación nace de una de las características más importantes del algoritmo del juego del caos de Jeffrey, está es la correspondencia biunívoca entre la secuencia de puntos $P = \{P_1 \ P_2 \ \dots \ P_n\}$ y la secuencia genómica (SG), debido a que la CGR es un mapeo uno a uno, P es única para cada genoma. Así conociendo las coordenadas de un punto cualquiera \mathbf{P}_n , es posible obtener las coordenadas de P_{n-1} usando una representación binaria para las coordenadas X y Y de los vértices. El proceso analítico para la construcción de la CGR es tomar el punto medio entre el punto anterior y uno de los vértices, es decir:

$$\begin{aligned} x_n &= \frac{1}{2}(x_{n-1} + x_V) \\ y_n &= \frac{1}{2}(y_{n-1} + y_V) \end{aligned} \quad (46)$$

Si tomamos la relación inversa de la expresión anterior, tendremos:

$$\begin{aligned}
 x_{n-1} &= 2x_n \text{ mod } 1 \\
 y_{n-1} &= 2y_n \text{ mod } 1
 \end{aligned}
 \tag{47}$$

Donde *mod1* lo que implica es que los valores obtenidos de los puntos no saldrán del cuadrado unitario, recordemos que el mapeo de Bernoulli estará dado por la expresión (48) y este mapeo se muestra gráficamente en la Figura 22.

$$S(x) = \begin{cases} 2x & \text{si } x < 0.5 \\ 2x - 1 & \text{si } x \geq 0.5 \end{cases}
 \tag{48}$$

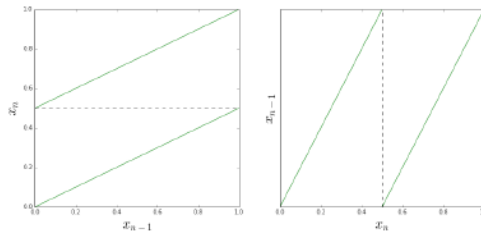


Figura 22: Mapeo de Bernoulli.

3.1.2. Generación de regiones mediante el uso de la CGR

En la sección anterior analizamos el mapeo del juego del caos de Jeffrey para generar secuencias de puntos, si recordamos en la sección 2.1 se describió la idea de Hausdorff sobre las cubiertas que pueden adoquinar a un objeto y como estas ayudan a la obtención de su dimensión, para el caso de las CGR es posible adoquinar con cubiertas cuadradas los puntos que conforman a la imagen final, estas serán regiones dentro del cuadrado Q , para generar estas regiones se usará un conjunto de transformaciones afines asociadas a la secuencia genómica. Se asociará una función contractiva a cada base principal de una secuencia, es decir que a una secuencia de ADN se le asociará un IFS, las cuales están dadas por las siguientes expresiones:

$$\begin{aligned}
\omega_A &= \omega_V(P_0) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
\omega_C &= \omega_V(P_0) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} \\
\omega_G &= \omega_V(P_0) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \\
\omega_T &= \omega_V(P_0) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}
\end{aligned} \tag{49}$$

De esta forma al aplicar las transformaciones al cuadrado unitario Q se obtienen los subcuadrados: $\omega_A(Q) = Q_A, \omega_C(Q) = Q_C, \omega_G(Q) = Q_G$ y $\omega_T(Q) = Q_T$ de lado $\frac{1}{2}$. Estas transformaciones conforman el *Operador de Hutchinson* \mathbf{W} (ecuación 50), el cual se observa en la Figura 23.

$$\mathbf{W}(Q) = \omega_A(Q) \cup \omega_C(Q) \cup \omega_G(Q) \cup \omega_T(Q) \tag{50}$$

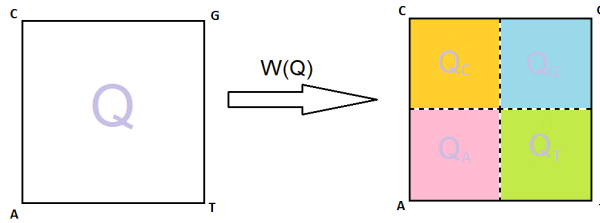


Figura 23: Aplicación del operador de contracción al cuadrado unitario.

El operador de Hutchinson es aplicado también a regiones que se encuentran dentro de Q , este operador es usado para construir cubiertas de diferentes tamaños para los puntos obtenidos de la aplicación de las transformaciones afines, es decir cuando \mathbf{W} es aplicado dos veces a Q se tiene:

$$\mathbf{W}^2(Q) = \cup_{V_2} \cup_{V_1} \omega_{V_2} \circ \omega_{V_1}(Q) = \cup_{V_2} \cup_{V_1} Q_{V_2V_1} \quad (51)$$

Donde $Q_{V_2V_1}$ representa los subcuadrados o cubiertas los cuales estarán definidos por los subíndices V_2V_1 , así tendremos una cuadrícula con 4^2 cuadrados de lado $(\frac{1}{2})^2$ donde $Q_{V_2V_1} \subset Q_{V_2}$ es decir el sub-subcuadrado $Q_{V_2V_1}$ estará dentro del subcuadrado Q_{V_2} así para k aplicaciones tenemos 4^k subcuadrados de lado $(\frac{1}{2})^k$ que conformarán Q , siendo Q invariante bajo \mathbf{W} . La secuencia de ADN generará un gran número de cuadrados de lado $[\frac{1}{2}, (\frac{1}{2})^2, \dots, (\frac{1}{2})^N]$, dichos cuadrados satisfacen la siguiente propiedad:

$$Q_{V_N V_{N-1} \dots V_2 V_1} \subset Q_{V_N V_{N-1} \dots V_2} \subset \dots \subset Q_{V_N V_{N-1}} \subset Q_{V_N} \subset Q \quad (52)$$

Podríamos pensar que el usar regiones en lugar de puntos no tiene relación, pero esto no es así debido a que al usar regiones se tomarán los puntos medio de estas, es decir P_0 es el punto medio de Q , $P_1 = \omega_{V_1}(P_0)$ es el punto medio de Q_{V_1} , $P_2 = \omega_{V_2} \circ \omega_{V_1}(P_0)$ es el punto medio de $Q_{V_2V_1}$, así sucesivamente $P_N = \omega_{V_N} \circ \dots \circ \omega_{V_1}(P_0)$ será el punto medio de $Q_{V_N \dots V_1}$.

De forma que $\mathbf{W}^N(Q)$ es la cubierta de Q , la cual estará conformada por 4^N cuadrados en los cuales habrá subsecuencias con sufijos que indican en que subcuadrado se encuentran. Cada subcuadrado o cubierta adosará aun solo punto el cual es el resultado de una subsecuencia de vértices $V_N(V_N = A, C, G, T)$ que estará dada por la secuencia genómica, es decir dado el trinucleótido $V_1V_2V_3$ este se encontrará en el subcuadrado $Q_{V_3V_2V_1}$, esto será discutido con más detalle en la siguiente sección.

3.1.3. Representación gráfica de secuencias genómicas usando IFS

Para la representación del juego del caos se traza un punto para cada sitio de la secuencia, los puntos se trazan a medio camino entre el punto previo y la esquina correspondiente a la base de cada subsecuencia sucesiva, es decir se tomarán las bases en el orden en el que aparezcan en la secuencia genómica, como se puede ver en la Figura 24. Los puntos en una CGR correspondientes a una base de la secuencia son trazados en el cuadrante etiquetado con dicha base. Es decir, como se mencionó anteriormente el diagrama CGR estará conformado por 4 cuadrantes principales, los cuales son el resultado de la

aplicación de una función contractiva aplicada al cuadrado unitario.

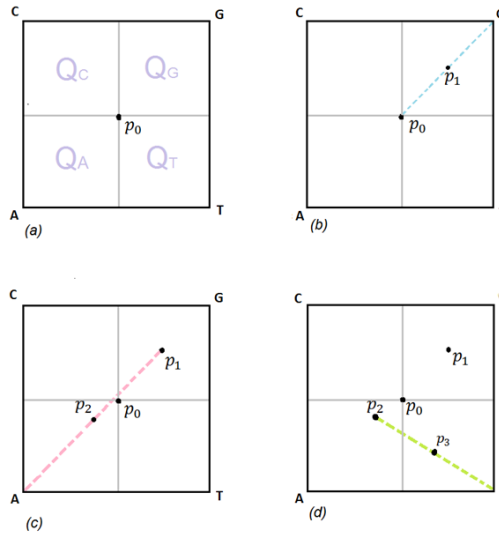


Figura 24: Primeros tres pasos de una secuencia $S = GAT$.

Cada cuadrante contiene todos los puntos que están a medio camino entre la esquina etiquetada con la base correspondiente al cuadrante y un punto anterior, por ejemplo si tenemos la secuencia genómica $S_g = AAGTCCTCCAGAG \dots$, la representación del juego del caos asocia una secuencia de puntos SP dentro del cuadrado unitario. En la Figura 25 es posible observar como cada punto correspondiente a una base dada estará en el cuadrante etiquetado por la misma base.

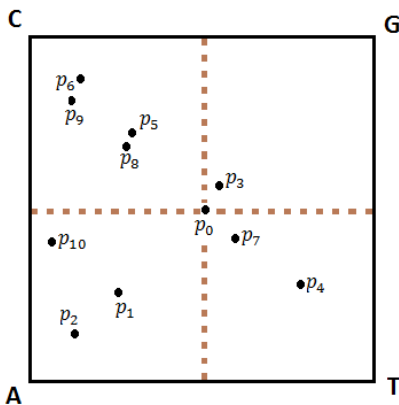


Figura 25: Visualización de puntos en cuadrantes.

Esta correspondencia entre puntos y subsecuencias continúa de manera recursiva a subcuadrantes, sub-subcuadrantes, etc. Por ejemplo la subsecuencia dinucleótida AT dará lugar a un punto en el sub-subcuadrante A que se encuentra en el cuadrante T (Figura 26 a), el trinucleótido TAG proporciona un punto en el sub-subcuadrante T que se encuentra en el subcuadrante A que se encuentra en el cuadrante G (Figura 26 b).

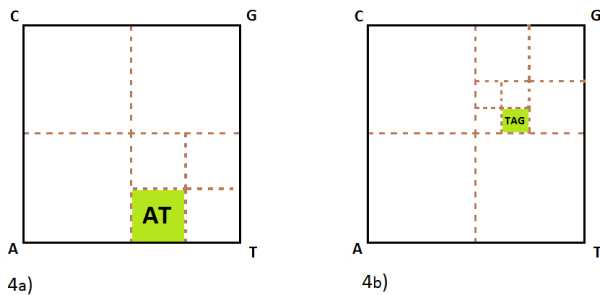


Figura 26: División de subcuadrantes

Al estudiar una secuencia de caracteres en este caso el ADN nos permite tener dos procesos sutilmente diferentes, uno de ellos es la *aplicación de funciones*, es decir la aplicación del IFS que caracteriza la secuencia; esta aplicación se da de acuerdo al orden de aparición de las bases en la secuencia. De esta forma la subsecuencia deberá aplicarse de izquierda a derecha, por otro lado el segundo proceso es de *localización de subcuadrantes*, donde

dado un subcuadrante podemos encontrarlo al usar la propiedad de las secuencias como las de ADN que genera una gran colección de subcuadrados. Esto significa que para el proceso de localización leeremos en orden inverso al proceso de aplicación de funciones es decir *leerá de derecha a izquierda*. La propiedad de localización es una propiedad de interés ya que una de las características más notorias en la CGR de la secuencia de la región beta globina humana (Jeffrey, 1992) es la región escasamente que se observa en dicho diagrama, esta región se repite en toda la CGR, presentando una propiedad de autosimilaridad. La mayor parte de esta región de escasez está localizada en el subcuadrante superior izquierdo, que corresponde al dinucleótido *CG*. En otras palabras, una escasez relativa de dinucleótidos *CG* está iniciada por el subcuadrante *CG* escasamente lleno.

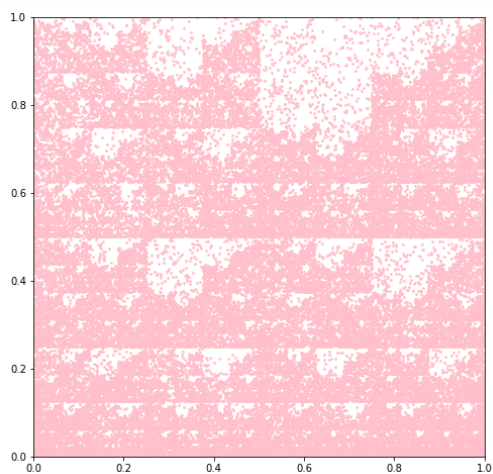


Figura 27: CGR de la secuencia de la región de beta globina humana (*HSHBB*, 73326pb).

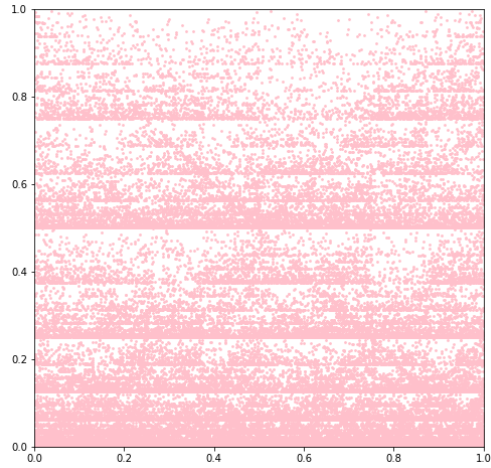


Figura 28: CGR de la *Amanita muscaria* mitochondrion.

La escasez de llenado en el subcuadrante GC indica la rareza del dinucleótido CG , esto a su vez significa que hay pocos trinucleótidos que contienen el dinucleótido CG , es decir los dinucleótidos $ACG, CCG, GCG, TCG, CGA, CGC, CGG$ y CGT se llenarán escasamente. Las primeras cuatro subsecuencias (XCG) con $X = A, C, G, T$, estarán dentro del subcuadrante GC , por lo que ya se encuentran en una región de escasez. Las otras cuatro subsecuencias, es decir, (CGX) se encontrarán fuera del subcuadrante CG , ya que el último carácter de estas subsecuencias indicará en que región (subcuadrante) estará dicha subsecuencia, similarmente para los 16 sub-sub-subcuadrantes correspondientes a las secuencias ($CGXY$) también se llenarán escasamente al igual que las 64 regiones ($CGXYZ$) o las 256 ($CGXYZW$) regiones, sus subcuadrantes serán cada vez más pequeños ($XTZW$ podrán tomar el valor de cualquiera de las bases de las secuencias: A, C, G o T).

Pero ¿Por qué son importantes estas zonas de escasas en la CGR?, Goldman (Goldman, 1993) propone que a través de estas regiones es posible recrear la CGR de una secuencia genómica en particular (en este caso la región beta globina humana), es decir es posible recrear los patrones de una secuencia mediante la unión de todas las regiones escasamente llenas, esto es posible a través de una representación matemática de una secuencia, como se muestra en la Figura 29.

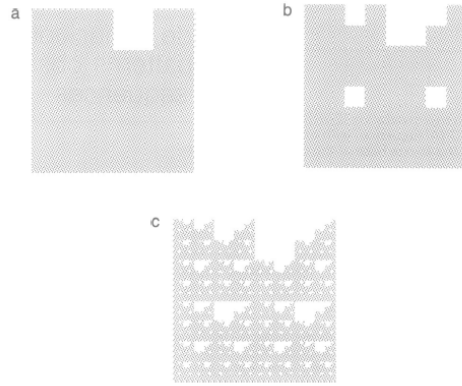


Figura 29: Recreación de la secuencia de la región de beta globina humana mediante la mezcla de las regiones escasamente llenas (Jeffrey, 1990).

La CGR de la Figura 29 también muestra una mayor densidad de puntos en sus diagonales $A-G$ y $C-T$. Estos puntos son causados por subsecuencias repetidas de dinucleótidos (AG) o (CT), por ejemplo $AGAGAGAGAGAGAGAG \dots$, en esta subsecuencia el primer par está dado por el dinucleótido AG el cual se encuentra en el subcuadrante GA , después tendremos el siguiente par el cual es GA , esto se debe a la forma en la que se aplican las funciones correspondientes a cada nucleótido de la secuencia, en la cual se recorre un lugar hacia la derecha en la secuencia, ahora este par estará en el subcuadrante AG , de esta forma después de muchas iteraciones se llenarán densamente las diagonales.

El k -ésimo punto trazado en el CGR de una secuencia corresponde a la primera subsecuencia inicial de la secuencia k -larga, y no a ninguna subsecuencia posterior hasta la resolución de la pantalla, por lo tanto existe una correspondencia uno a uno entre las subsecuencias (ancladas al inicio) de un gen y los puntos de la CGR, esto es descrito por el siguiente teorema:

Teorema 3.1

Existe un mapeo uno a uno entre las secuencias y el interior del cuadrado unitario, en el que el k -ésimo punto trazado en la CGR de una secuencia corresponde a la primera subsecuencia inicial de k de la secuencia, y ninguna otra subsecuencia (hasta la resolución de la pantalla) por lo tanto, existe una correspondencia entre las subsecuencias (ancladas al inicio) de una secuencia y los puntos de la CGR (Goldman)

Hasta el momento se ha hecho mención de secuencias ya determinadas como lo son las secuencias genómicas en las cuales se pueden visualizar patrones presentes en dichas secuencias, pero si en su lugar se tomara una secuencia aleatoria con las mismas probabilidades ¿Qué patrones observaremos? La respuesta es sencilla, se obtendrá un cuadrado uniformemente lleno (Figura 30), pero si por el contrario se asignaran diferentes probabilidades a los caracteres de la secuencia tomando las etiquetas A, C, G y T ¿Qué se obtendría? En este caso aparecerán bandas horizontales y verticales (Figura 31) en el diagrama, ya que dentro de cada subcuadrante se encuentran regiones correspondientes a todas las etiquetas las cuales tendrán su probabilidad.

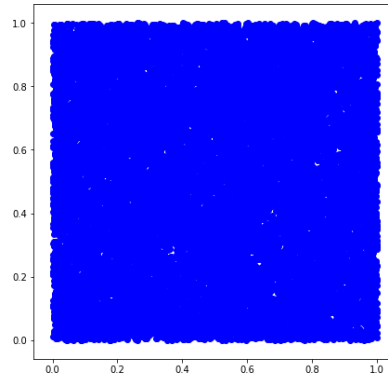


Figura 30: Secuencia de 20000 caracteres con las mismas probabilidades.

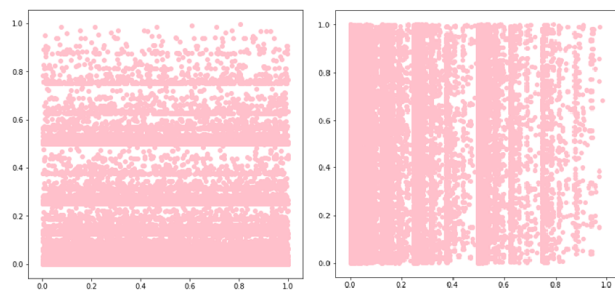


Figura 31: Secuencia de 20000 caracteres con diferentes probabilidades.

4. Cadenas de Markov

Una generalización directa de los *procesos estocásticos independientes* es la de los procesos Markovianos, que fueron estudiados por primera vez por el matemático ruso A. A. Markov, mejor conocidos como cadenas de Markov. Estas cadenas involucran transiciones entre valores de una variable estocástica discreta E a diferentes tiempos. Para definir las, supongamos que tenemos un experimento con k posibles resultados $(E_1^{(n)}, E_2^{(n)}, \dots, E_k^{(n)})$, teniendo una secuencia temporal de procesos en los cuales uno y solo uno de los k eventos pueden ocurrir a cada tiempo (el superíndice denota el número del experimento). Se define una cadena de Markov, cuando la probabilidad de que ocurra cada uno de los estados en la $(n + 1)$ -ésima repetición dependa solo del resultado del n -ésimo experimento.

Asumamos que el tiempo es medido en pasos discretos, es decir, $t = s\tau$ donde s es un entero y τ es un intervalo de tiempo fundamental. De tal forma que para un paso de $t = 0$ a $t = \tau$ la probabilidad p_{ij} de que para $t = 0$ ocurra E_i y al tiempo τ que ocurra E_j estará dada por:

$$p_{ij} = \sum_{i=1}^N p_{i|j} \cdot p_i \quad (53)$$

Donde p_{ij} es la *probabilidad conjunta* la cual denota la probabilidad de que ocurra el par ij , p_i es la probabilidad de que a $t = 0$ ocurra el estado E_i y $p_{i|j}$ es la *probabilidad condicional* la cual indica que dado que a $t = 0$ ocurrió el estado E_i al tiempo τ ocurrirá el estado E_j , esta cantidad también es conocida como la *probabilidad de transición* del estado E_i al estado E_j y contiene toda la información posible acerca del mecanismo de transición en un paso. Una condición necesaria para los procesos Markovianos es que las probabilidades de transición no cambien en el tiempo, así esta probabilidad no depende del instante n en el que ocurra la transición. Para ejemplificar un proceso markoviano consideremos los siguientes experimentos.

Ejemplo 4.1: Extracción de canicas.

Tenemos dos bolsas una roja (bolsa 1) y una azul (bolsa 2) con canicas en su interior. La bolsa 1 contiene 3 canicas rojas y 2 canicas azules, mientras que la bolsa 2 contiene 3 canicas rojas y 7 canicas azules. El experimento

consiste en sacar una canica de una de las bolsas, se registra *su color* para inmediatamente ser devuelta a la bolsa de la cual fue extraída. La siguiente canica será extraída de la bolsa que tiene *el mismo color* de la última canica extraída.

Para cada extracción se tienen 2 estados posibles, obtener una canica roja o una canica azul. La probabilidad de conseguir una canica roja en la $(n + 1)$ -ésima extracción depende solo de la canica que se obtuvo en la n -ésima extracción, ya que esto es lo que determinará de que bolsa se extraerá la siguiente canica (Sandefur, 1990).

Una de las características principales de los procesos markovianos es que sus probabilidades de transición no cambien, en este ejemplo cada canica que es extraída de las bolsas es regresada a la bolsa de donde fue extraída, por lo tanto la probabilidad de sacar una canica roja o azul de la bolsa 1 o de la bolsa 2 no cambia, de tal forma que denotaremos a $p(n)$ como la probabilidad de que en la n -ésima extracción la canica sea roja y $q(n) = 1 - p(n)$ será la probabilidad de que la n -ésima extracción la canica sea azul. Por lo tanto la probabilidad de que la primer canica extraída de la bolsa 1 sea roja será $p(1) = \frac{3}{5} = 0.6$ y la probabilidad de que sea azul será $q(1) = 0.4$. Para el siguiente caso se busca conocer cuál es la probabilidad de que la segunda canica extraída sea roja, es decir $p(2)$; Para esto se tienen dos posibilidades, en el primer caso cuando la primer canica extraída es roja y la segunda canica es roja, esto será denotado como rr y la probabilidad de que esto ocurra es p_{rr} , el segundo caso es que la primer canica sea azul y la segunda canica sea roja y estará denotada como ar con una probabilidad p_{ar} .

Para obtener $p(2)$ se tienen que calcular las probabilidades de cada uno de los casos y sumarlos, es decir:

$$p(2) = p_{rr} + p_{ar} \tag{54}$$

Para obtener la probabilidad conjunta rr tenemos que observar que este es un proceso de dos etapas, en la primera etapa se obtiene una canica roja en la primera extracción (dado que estamos en la bolsa 1) y la probabilidad de que esto ocurra es $p(1) = 0.6$, en la segunda etapa es obtener una canica roja en la segunda extracción dado que estamos en la bolsa 1 y la probabilidad

de que esto ocurra es de 0.6 la probabilidad de obtener una canica roja en la primera y en la segunda extracción (probabilidad conjunta) es el producto de las probabilidades individuales:

$$p_{rr} = p_{1|1}p_1(n) = (0.6)p(1) = (0.6)^2 = 0.36$$

Donde en la notación $p_{1|1}$ el primer sub-índice indica el proceso que ya pasó y el segundo sub- índice indica el proceso que va a ocurrir. Para obtener ahora la probabilidad conjunta ar nuevamente tendremos un proceso de dos etapas, donde en la primera etapa se obtiene una canica azul en la primera extracción (dado que estamos en la bolsa 1) con lo que tendríamos una probabilidad de $q(1) = 0.4$ y la segunda etapa es obtener una canica roja en la segunda extracción pero esta extracción sería de la bolsa 2 dado que la canica anterior fue azul, de tal forma que la probabilidad de que esto suceda es 0.3, nuevamente por el principio de multiplicación utilizado anteriormente la probabilidad conjunta del caso 2 es:

$$p_{ar} = p_{2|1}q(n) = (0.3)q(1) = (0.3)(0.4) = 0.12$$

De esta forma la probabilidad de que la segunda canica extraída sea roja es:

$$p_2 = 0.6p(1) + 0.3q(1) = 0.36 + 0.12 = 0.48$$

La misma lógica es usada para calcular la probabilidad de sacar una canica azul en la segunda extracción ($q(2)$), pero este caso es más simple debido a que ya conocemos $p(2)$, de modo que:

$$q_2 = 1 - p(2) = 0.52$$

Ahora bien ¿Cuál será la probabilidad de que la $(n + 1)$ -ésima canica extraída sea roja? Para responder esta pregunta nuevamente tendremos dos

casos, el caso 1 es que la n -ésima canica sea roja y la canica $(n + 1)$ sea roja, denotada por rr , mientras que el caso 2 es que la n -ésima canica sea azul y la canica $(n + 1)$ sea roja, denotada como ar , cabe mencionar que solo se numerarán las dos últimas canicas extraídas, así la probabilidad de que la $(n + 1)$ -ésima canica extraída sea roja estará descrita por la expresión (55).

$$p(n + 1) = p_{rr} + p_{ar} \quad (55)$$

Para calcular la probabilidad rr tendremos dos etapas, en la primera etapa en la n -ésima extracción obtenemos una canica roja con una probabilidad $p(n)$ y en la segunda etapa la extracción $(n + 1)$ será una canica roja, es decir se extraerá la siguiente canica de la bolsa roja, de tal forma que la probabilidad de que esto ocurra es $p_{1|1} = 0.6$, teniendo así:

$$p_{rr} = 0.6p(n)$$

De manera similar para obtener la probabilidad de ar tendremos:

$$p_{ar} = 0.3q(n)$$

Sumando estos dos casos:

$$p(n + 1) = 0.6p(n) + 0.3q(n) \quad (56)$$

Ya que debemos obtener una canica roja o una canica azul en la n -ésima extracción se deberá cumplir la siguiente condición [11]:

$$p(n) + q(n) = 1 \quad \text{o} \quad q(n) = 1 - p(n) \quad (57)$$

Sustituyendo entonces (57) en (56) obtendremos el siguiente sistema dinámico de primer orden:

$$p(n + 1) = 0.6p(n) + 0.3(1 - p(n)) = 0.3p(n) + 0.3 \quad (58)$$

Definamos ahora $p_1(n)$ como la probabilidad de que ocurra el estado E_1 en la n -ésima repetición del experimento, por lo tanto $p_2(n), \dots, p_m(n)$ serán las probabilidades de que ocurran los estados E_2, \dots, E_m en la n -ésima repetición respectivamente. Ya que uno de los m estados debe ocurrir en la n -ésima repetición, se deduce que:

$$p_1 + p_2 + \dots + p_m(n) = 1 \quad (59)$$

En la Teoría de cadenas de Markov se habla de ciertos sistemas físicos en los que a cada instante de tiempo pueden estar en uno de los estados E_1, E_2, \dots, E_k y alterar su estado solo a tiempos t_1, t_2, \dots, t_n , veamos un ejemplo.

Ejemplo 4.2: Caminata aleatoria.

Imagine que una partícula ubicada en una línea recta se mueve a lo largo de la línea a través de impactos aleatorios que ocurren a tiempos t_1, t_2, t_3, \dots . La partícula puede estar en puntos con coordenadas integrales $a, a + 1, a + 2, \dots, b$, donde en los puntos a y b hay barreras reflectantes. Cada impacto desplaza la partícula a la derecha con probabilidad p y a la izquierda con probabilidad $q = 1 - p$ siempre que la partícula no se encuentre en alguna de las barreras. Si la partícula está en una barrera cualquier impacto lo transferirá una unidad dentro del espacio entre las barreras. Como se observa en la Figura 4.1, en el caso (1) la partícula solo tendrá la opción de moverse una unidad a la derecha terminando así en la posición $a + 1$, si ahora la partícula se encuentra en la posición $a + 1$ (caso (2)) se tendrán dos posibilidades una será que la partícula se desplace una unidad a la derecha y la otra será que se desplace una unidad a la izquierda.

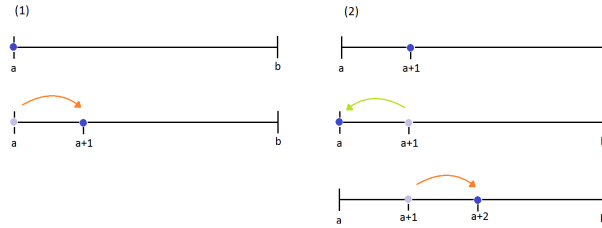


Figura 32: Partícula en una caminata aleatoria entre barreras reflectantes.

Esta descripción se aplicará a cada posición que la partícula pueda tomar a lo largo de la recta \overline{ab} . Como podemos ver las probabilidades de transición son una parte fundamental de los procesos markovianos, por lo tanto es necesario profundizar un poco más en el tema. Nos limitaremos a estudiar las características más elementales para cadenas de Markov homogéneas.

Hasta el momento hemos estudiado sistemas en los cuales existen solo dos estados, pero ¿Qué pasará cuando el sistema tenga más estados?. Imaginemos que tenemos un sistema con m estados y queremos calcular la probabilidad de que en la $(n+1)$ -ésima repetición del experimento el sistema se encuentre en el estado E_1 , dado que para este sistema hay m estados, existen m formas de que esto ocurra. El primer caso es que en la n -ésima repetición y en la $(n+1)$ -ésima repetición se obtenga E_1 , este caso estará expresado como E_1E_1 , el segundo caso es que en la n -ésima repetición obtengamos E_2 y en la siguiente repetición se obtenga E_1 , descrito como E_2E_1 y para el caso m se tendrá que en la n -ésima repetición el sistema se encuentre en el estado E_m y en la repetición $(n+1)$ el sistema pasará al estado E_1 , es decir E_mE_1 . Como podemos observar tendremos procesos de dos etapas (como en el ejemplo 1), donde en la primera etapa se tiene el estado E_i para la n -ésima repetición y E_j para la $(n+1)$ -ésima repetición con $i, j = 1, 2, \dots, m$. Por lo tanto la probabilidad conjunta estará dada por la expresión (53), para nuestro ejemplo para el caso 1 el sistema estará en el estado E_1 , el cual tendrá una probabilidad $p_1(n)$, pero dado que queremos que el sistema pase de este estado al estado E_1 tendremos una probabilidad de transición $p_{1|1}$, teniendo así que la probabilidad conjunta para el primer caso estará dada de la siguiente forma:

$$p(\text{caso1}) = p_{1|1}p_1(n)$$

De la misma forma para los m casos posibles tendremos:

$$p(\text{caso2}) = p_{2|1}p_2(n)$$

⋮

$$p(\text{casom}) = p_{m|1}p_m(n)$$

De tal forma que $p_{i|j}$ es la probabilidad condicional de $E_i E_j$, que indica la probabilidad de que ocurra E_j en la siguiente repetición dado que E_i ya ocurrió en la repetición anterior. Sumando las probabilidades de estos casos se tendrá la probabilidad de obtener E_1 en la $(n + 1)$ -ésima repetición:

$$p_1(n + 1) = p_{1|1}p_1(n) + p_{2|1}p_2(n) + \cdots + p_{m|1}p_m(n)$$

Este proceso se repetirá para los demás estados, es decir si queremos ahora que en la $(n + 1)$ -ésima repetición el sistema se encuentre en el estado E_j , con $j = 2, 3, \cdots, m$, tendremos el siguiente sistema de ecuaciones:

$$p_2(n + 1) = p_{1|2}p_1(n) + p_{2|2}p_2(n) + \cdots + p_{m|2}p_m(n)$$

⋮

$$p_m(n + 1) = p_{1|m}p_1(n) + p_{2|m}p_2(n) + \cdots + p_{m|m}p_m(n)$$

Esto nos dará un sistema dinámico de m ecuaciones, que podemos expresar en forma matricial como:

$$\mathbf{P}(n + 1) = \mathbf{QP}(n) \tag{60}$$

Donde $\mathbf{P}(n)$ es el **vector de probabilidad** asociado con la cadena de Markov y estará definido como:

$$\mathbf{P}(n+1) = \begin{pmatrix} p_1(n) \\ \vdots \\ p_m(n) \end{pmatrix} \quad (61)$$

Y \mathbf{Q} será la matriz de transición la cual se expresa en la ecuación (62).

$$\begin{pmatrix} Q_{11=p_{1|1}} & \cdots & Q_{1k=p_{1|k}} \\ \vdots & \ddots & \vdots \\ Q_{k1=p_{k|1}} & \cdots & Q_{kk=p_{k|k}} \end{pmatrix} \quad (62)$$

Ya que cada componente $p_{j(k)}$ de $\mathbf{P}(k)$ es la probabilidad de que el evento E_j ocurra en la k -ésima repetición, se deberá cumplir la siguiente condición:

$$0 \leq p_j(k) \leq 1 \quad (63)$$

Esto significa que cada componente del vector $\mathbf{P}(k)$ debe estar en el intervalo $[0, 1]$, por otro lado ya que alguno de los eventos E_j debe ocurrir, la suma de los componentes de $\mathbf{P}(k)$ debe ser igual a 1. Debido a que cada uno de los componentes de Q es una probabilidad, todos los valores en la matriz de transición estarán en el intervalo $[0, 1]$, por otro lado dado que en las transiciones el sistema debe pasar por uno y solo uno de los estados las columnas de la matriz cumplen la siguiente propiedad:

$$\sum_j^m p_{i|j} = 1, \quad (i = 1, 2, \dots, k)$$

Para comprender estos procesos con varios estados veamos algunos ejemplos.

Ejemplo 4.3: Matriz de transición para barreras reflectantes.

Vamos a escribir la matriz de transición para el caso descrito en el ejemplo 2 de la partícula en una caminata aleatoria entre dos barreras reflectantes. Si denotamos E_1 como el evento que indica que la partícula esté en el punto con coordenadas a , E_2 es el evento en el cual la partícula está en un punto con

coordenadas $a + 1$ y E_s con $s = b - a + 1$ estará en un punto con coordenadas b , de esta forma la matriz de transición estará dada como:

$$\mathbf{Q}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ q & 0 & p & 0 & \cdots & 0 \\ 0 & q & 0 & p & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

Ejemplo 4.4: Matriz de transición para barreras absorbentes.

Describamos ahora la matriz de transición de una partícula en una caminata aleatoria entre barreras absorbentes. La notación y condiciones permanecen igual que en el ejemplo anterior pero en este caso cuando la partícula pasa al estado E_1 o al estado E_s la partícula permanecerá en estos estados con probabilidad igual a 1.

$$\mathbf{Q}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ q & 0 & p & 0 & \cdots & 0 \\ 0 & q & 0 & p & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

4.1. Condiciones para la matriz de transición

Los elementos de la matriz de transición deben satisfacer algunas condiciones, una de estas es la siguiente condición:

$$0 \leq p_{ij} \leq 1 \tag{64}$$

Esto se debe a que los elementos de la matriz de transición son probabilidades. Además debido a que la transición del estado $E_i^{(s)}$ al estado $E_j^{(s+1)}$ para el proceso (s) y $(s + 1)$ respectivamente debe definitivamente pasar por uno y solo uno de los estados $E_j^{(s+1)}$, se cumple la siguiente condición:

$$\sum_{j=1}^k p_{i|j} = 1, \quad (i = 1, 2, \dots, k) \tag{65}$$

La cual indica que los estados son mutuamente exclusivos y colectivamente exhaustivos. Por otro lado la matriz de transición será estocástica puesto que la suma de los elementos de cada renglón será igual a 1.

Uno de los principales problemas en la Teoría de cadenas de Markov consiste en determinar las probabilidades de transición del estado $E_i^{(s)}$ (para el s -ésimo proceso) al estado $E_j^{(s+n)}$ después de n procesos. Para comprender más a fondo estas probabilidades se examinarán algunos procesos intermedios, para esto se tomará el proceso $(s + m)$ donde uno de los posibles estados es $E_r^{(s+m)}$ con $(1 \leq r \leq k)$, la probabilidad de dicha transición es igual a $P_{i|r}(m)$ y la probabilidad de dicha transición del estado $E_r^{(s+m)}$ al estado $E_j^{(s+n)}$ es $P_{r|j}(n - m)$. De la expresión (54) obtenemos una generalización de las probabilidades conjuntas, esta es la ecuación de Chapman-Kolmogorov (expresión 66).

$$P_{ij}(n) = \sum_{r=1}^k P_{i|r}(m) \cdot P_{r|j}(n - m) \quad (66)$$

Así la matriz de transición después de n procesos será:

$$\mathbf{Q}_n = \begin{pmatrix} P_{1|1}(n) & P_{1|2}(n) & \cdots & P_{1|k}(n) \\ \vdots & \vdots & \vdots & \vdots \\ P_{k|1}(n) & P_{k|2}(n) & \cdots & P_{k|k}(n) \end{pmatrix} \quad (67)$$

De acuerdo con (66) se tiene la siguiente expresión matricial:

$$\mathbf{Q}_n = \mathbf{Q}_m \cdot \mathbf{Q}_{n-m}, \quad (0 < m < n) \quad (68)$$

Y de esta relación podemos observar que para $m = 1$ y $n = 2$:

$$\mathbf{Q}_2 = \mathbf{Q}_1 \cdot \mathbf{Q}_1 = \mathbf{Q}_1^2$$

Para $n = 3$:

$$\mathbf{Q}_3 = \mathbf{Q}_1 \cdot \mathbf{Q}_2 = \mathbf{Q}_2 \cdot \mathbf{Q}_1 = \mathbf{Q}_1^3$$

Y en general para cualquier n :

$$\mathbf{Q}_n = \mathbf{Q}_1^n \quad (69)$$

Notemos un caso especial de la expresión (66), en el que el paso intermedio es $m = 1$:

$$P_{ij}(n) = \sum_{r=1}^k P_{i|r} \cdot P_{r|j}(n - m) \quad (70)$$

4.2. Clasificación de posibles estados

La clasificación de estados para cadenas de Markov se realiza en conjuntos contables y finitos de estados. Se tendrán dos tipos de estados para estas cadenas, uno de estos es llamado estado no esencial (o transitorio), es decir E_i es un estado no esencial si existe un estado E_j para un proceso n tal que $P_{i|j}(n) > 0$ pero $P_{i|j}(m) = 0$ para todo m , por lo tanto para un estado transitorio es posible pasar de un estado a otro con probabilidades positivas pero ya no es posible regresar al estado inicial.

Un ejemplo de estos estados se puede observar en el ejemplo 4.4 que analiza a una partícula dentro de barreras absorbentes, en este ejemplo todos los estados excepto E_1 y E_s son transitorios ya que sin importar en qué estado se encuentre la partícula esta puede alcanzar cualquiera de los estados, E_1 y E_s con probabilidades $P_{i|j}(n) > 0$ mediante un número finito de pasos pero una vez que la partícula llega a estos estados no puede regresar de ellos (las barreras reflectantes) a ningún otro estado, es decir $P_{i|j}(m) = 0$.

El segundo tipo de estados son los estados esenciales, un estado E_i es esencial si este transiciona a un estado E_j y eventualmente puede regresar al estado E_i después de haber salido de él, es decir la probabilidad de pasar de un estado a otro en el proceso n será $P_{i|j}(n) > 0$, mientras que la probabilidad de regresar al estado inicial en un proceso m será $P_{j|i}(m) > 0$. Si E_i y E_j son tales que para ambos se mantienen estas desigualdades. Dados ciertos m y n entonces se les llamará **comunicantes**. De este modo si E_i se comunica con E_j y E_j se comunica con E_k , entonces E_i también se comunicará con E_k , teniendo de este modo estados conexos, de tal forma que será posible ir de un estado a cualquier otro; Expresado de otra forma un estado E_j es recurrente si $p_{j|j} = 1$ y transitorio si $p_{j|j} < 1$. Si E_j es recurrente y la cadena comienza en E_j , entonces regresa a E_j con probabilidad 1. Si, en cambio, E_j es transitorio, hay una probabilidad positiva e igual a $1 - p_{j|j}$ de que si la cadena comienza en E_j , nunca regresará a ese estado. Si E_j es un estado

absorbente $p_j = 1$ y por lo tanto $p_{j|j} = 1$, de modo que un estado absorbente es necesariamente recurrente.

4.2.1. Probabilidades limitantes.

Como lo hemos mencionado anteriormente la parte fundamental de estos procesos Markovianos serán las probabilidades condicionales de transición. Dada la probabilidad inicial $P(0)$ lo que se buscará obtener será la probabilidad al siguiente paso $P(1)$ que estará dada por la siguiente expresión:

$$\mathbf{P}(1) = \mathbf{P}(0) \cdot \mathbf{Q} \quad (71)$$

De tal forma que para el siguiente paso se tiene que:

$$\mathbf{P}(2) = \mathbf{P}(1) \cdot \mathbf{Q} = \mathbf{P}(0) \cdot \mathbf{Q}^2 \quad (72)$$

Así después de n pasos la probabilidad será:

$$\mathbf{P}(n) = \mathbf{P}(0) \cdot \mathbf{Q}^n \quad (73)$$

Hasta ahora solo hemos analizado las probabilidades de transición para tiempos cortos, es decir para n pequeñas, pero ¿Qué pasará cuando n tiende a infinito?. Una pregunta importante en el desarrollo a tiempos largos es si el sistema tiene algo de memoria de su estado inicial o de transiciones de más estados a algún estado final, es decir, ¿El estado final depende del estado inicial a tiempos largos? Para responder esta pregunta analizaremos el siguiente teorema:

<i>Teorema 4.1</i>

<p><i>Si para algún $s > 0$ todos los elementos de la matriz de transición Q^s son positivos (matriz de transición regular), entonces existen números constantes $p_j (j = 1, 2, \dots, k)$ tales que independientemente del subíndice i se mantiene la igualdad:</i></p> $\lim_{n \rightarrow \infty} P_{i j}(n) = P_j \text{ (Gnedenko, 1997).}$

El Teorema 4.1 expresa que para cada columna de Q^s la diferencia entre el máximo elemento y el mínimo elemento de la matriz de transición tiende a cero conforme s tiende a infinito, esta demostración no se realizará en este trabajo pero para más información ir a la referencia (Gnedenko, 1997). Para

probar el Teorema 4.1 debemos tener en cuenta algunas propiedades importantes de las probabilidades de transición:

- I. La mayor de las probabilidades $P_{i|j}(n)$ no puede aumentar con el crecimiento de n y la menor no puede disminuir:

$$\lim_{n \rightarrow \infty} \min_{1 \leq l \leq k} P_{l|j}(n) = \bar{p}_j \quad (74)$$

$$\lim_{n \rightarrow \infty} \max_{1 \leq i \leq k} P_{i|j}(n) = \overline{\bar{p}}_j \quad (75)$$

- II. El máximo de la diferencia $P_{i|j}(n) - P_{l|j}(n)$ con $(i, l = 1, 2, \dots, k)$ tiende a cero cuando n tiende a infinito, es decir se tomará la diferencia entre la probabilidad de transición más grande y la más pequeña de la matriz \mathbf{Q}^s , como se expresa en la siguiente ecuación:

$$\lim_{n \rightarrow \infty} \max_{1 \leq i, l \leq k} |P_{i|j}(n) - P_{l|j}(n)| = 0 \quad (76)$$

De esta forma para tiempos grandes $\bar{p}_j - \overline{\bar{p}}_j = p_j$, es decir la probabilidad de transición del estado i al j para n tendiendo a infinito no dependerá del estado inicial i como se muestra en la Figura 33.

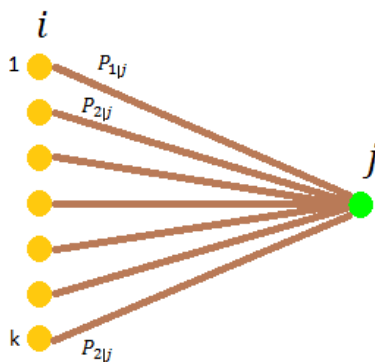


Figura 33: Transición de estados para tiempos grandes.

Una matriz estocástica Q es llamada **regular** si todos los elementos de alguna potencia de la matriz de transición Q^s son positivas. Si Q es regular

entonces el vector de probabilidad $P(s)$ tiende a un único vector de probabilidad fijo $\boldsymbol{\pi}$, de tal forma que se acerque a un estado estacionario, cabe mencionar que las componentes de $\boldsymbol{\pi}$ son todas positivas. La secuencia de potencias de Q, Q^2, Q^3, \dots tenderán a una única matriz de transición estacionaria como es mencionado en el Teorema 4.1, cuyas filas son cada una el vector estacionario $\boldsymbol{\pi}$:

$$\mathbf{P}(S) \rightarrow \boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_l) \quad (77)$$

Dónde:

$$\lim_{S \rightarrow \infty} \mathbf{Q}^S = \mathbf{M} = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_l \\ \vdots & \vdots & \vdots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_l \end{pmatrix} \quad (78)$$

Ahora bien si multiplicamos al vector de probabilidad inicial $\mathbf{P}(0)$ por la matriz \mathbf{M} , se obtendrá un **vector estacionario**:

$$\mathbf{P}_{st} = \mathbf{P}(0) \cdot \mathbf{M} \quad (79)$$

De tal forma que cada vez que se aplique la matriz de transición \mathbf{Q} al vector estacionario \mathbf{P}_{st} la matriz de transición no modificará al vector, siendo esta una propiedad importante del vector estacionario. Para demostrar esto partamos de la siguiente expresión:

$$\mathbf{P}_{st} \cdot \mathbf{Q} = \mathbf{P}(0) \cdot \mathbf{M} \cdot \mathbf{Q} \quad (80)$$

Y haciendo uso de la expresión (78) se tiene que:

$$\mathbf{M} \cdot \mathbf{Q} = (\lim_{n \rightarrow \infty} \mathbf{Q}^n) \cdot \mathbf{Q} = \lim_{n \rightarrow \infty} \mathbf{Q}^{n+1} = \mathbf{M} \quad (81)$$

Sustituyendo la expresión anterior en la expresión (80):

$$\mathbf{P}_{st} \cdot \mathbf{Q} = \mathbf{P}(0) \cdot \mathbf{M}$$

y usando la expresión (79) se llegará a la siguiente ecuación:

$$\mathbf{P}_{st} \cdot \mathbf{Q} = \mathbf{P}_{st} \quad (82)$$

Esta será la propiedad más importante de los sistemas estocásticos que estudiaremos aquí. Es posible notar que la expresión (82) es una **ecuación**

de eigenvalores donde el vector \mathbf{P}_{st} es eigenvector de la matriz Q con eigenvalor 1 debido a que \mathbf{P}_{st} no cambia, por lo que ahora estaremos interesados en encontrar el eigenvector correspondiente al eigenvalor 1 ya que este será el vector estacionario; Veamos un ejemplo que nos ayude comprender mejor estas definiciones.

Ejemplo 4.5: Bolsas de canicas

Supongamos que tenemos tres bolsas de canicas una bolsa roja, una azul y una amarilla. La bolsa roja contiene 2 canicas rojas, 3 azules y 5 amarilla, la bolsa azul contiene una roja, 5 azules y 4 amarillas y la bolsa amarilla contiene 3 rojas, 1 azul y 6 amarillas. Una canica es sacada de una de las bolsas, su color es registrado y la canica es regresada inmediatamente a la bolsa de donde fue extraída, la siguiente canica es extraída de la bolsa cuyo color sea igual al de la canica extraída anteriormente. Definiremos E_1 como el estado que corresponde a extraer una canica roja, E_2 como el estado que corresponde a extraer una canica azul y E_3 como el estado que corresponde a extraer una canica amarilla. De esta forma si se acaba de extraer una canica roja (E_1) sabemos que la siguiente canica se extraerá de la bolsa roja. Este experimento estará descrito por las siguientes probabilidades (Gnedenko, 1997):

$$\begin{aligned} p_{1|1} &= 0.2 & p_{2|1} &= 0.3 & p_{3|1} &= 0.5 \\ p_{1|2} &= 0.1 & p_{2|2} &= 0.5 & p_{3|2} &= 0.4 \\ p_{1|3} &= 0.3 & p_{2|3} &= 0.1 & p_{3|3} &= 0.6 \end{aligned} \tag{83}$$

Expresando las ecuaciones en (83) en su forma matricial tendremos:

$$\mathbf{Q} = \begin{pmatrix} 0.2 & 0.1 & 0.3 \\ 0.3 & 0.5 & 0.1 \\ 0.5 & 0.4 & 0.6 \end{pmatrix} \tag{84}$$

Podemos observar que los números de cada columna de \mathbf{Q} suman a 1 ya que los elementos de la matriz \mathbf{Q} son las proporciones de canicas rojas, azules y amarillas en cada bolsa (para cada columna). Ahora bien lo que buscamos será la solución general al sistema dinámico, es decir:

$$\mathbf{P}(n+1) = \mathbf{Q}\mathbf{P}(n) \tag{85}$$

Es fácil observar que la matriz \mathbf{Q} es una matriz regular, de tal forma que es posible encontrar su vector de probabilidad estacionaria, por lo tanto es

necesario resolver la ecuación de eigenvalores (82):

$$\mathbf{P}_{st} \cdot \mathbf{Q} - \mathbf{1} = 0$$

Como fue mencionado se tomará el eigenvalor $\lambda = 1$ para cumplir la condición necesaria de las matrices regulares, por lo que tenemos:

$$\mathbf{P}_{st} \cdot (\mathbf{Q} - \mathbf{1}) \begin{pmatrix} -0.8 & 0.1 & 0.3 \\ 0.3 & -0.5 & 0.1 \\ 0.5 & 0.4 & -0.6 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (86)$$

Resolviendo la ecuación anterior se obtendrá el siguiente vector de probabilidad estacionario:

$$\mathbf{P}_{st} = \left(\frac{16}{70} \quad \frac{17}{70} \quad \frac{37}{70} \right) \quad (87)$$

Así las potencias de una matriz estocástica regular \mathbf{Q} se acercarán a una matriz de estado estacionario \mathbf{M} , de tal forma que $\boldsymbol{\pi} = \mathbf{P}(0) \cdot \mathbf{M}$. Las cadenas de ADN son cadenas de Markov regulares por lo tanto contiene solamente estados esenciales, es decir en estas cadenas se puede ir de un estado a cualquier otro sin restricciones.

5. Revisión de algunos aspectos de Biología Molecular

Existen dos tipos de ácidos nucleicos los cuales son estructuralmente y químicamente distintos, estos son el ribonucleico (ARN) que se presenta cuando el azúcar es ribosa y el cual es conformado por una cadena simple y el otro tipo es el desoxirribonucleico (ADN) cuya estructura se conforma de una cadena doble y será aquel cuya azúcar es la desoxirribosa. El ARN y ADN son polímeros formados por largas cadenas de nucleótidos, siendo esta la pieza fundamental de los ácidos nucleicos. Los nucleótidos son moléculas orgánicas formadas por la unión covalente de una pentosa (azúcar), una base nitrogenada y un grupo fosfato.

Bases nitrogenadas: Las bases nitrogenadas son los componentes que contienen la información genética, son moléculas formadas de átomos de carbono y nitrógeno que crean anillos heterocíclicos aromáticos *Purina* y *Pirimidina*, de tal forma que las bases nitrogenadas se derivan en dos tipos:

- **Bases nitrogenadas purícas:** Estas son la Adenina (*A*) y la Guanina (*G*), ambas forman parte del ADN y del ARN.
- **Bases nitrogenadas pirimidínicas:** Estas son la Timina (*T*), la Citosina (*C*) y el Uracilo (*U*). La Timina y la Citosina intervienen en la formación del ADN, mientras que en el ARN aparecerán la Citosina y el Uracilo.

Pentosa: Azúcar formado por una cadena de cinco átomos de carbono; estos azúcares pueden ser ribosa que estará presente en el ARN o la desoxirribosa que se encuentra en el ADN.

Ácido fosfórico: Consiste en un átomo de oxígeno rodeado por cuatro átomos de fósforo y está unido al carbono 5' de la pentosa. Cada nucleótido puede contener uno, dos o tres grupos fosfatos. Si el nucleótido posee solo un grupo fosfato se dirá que este se encuentra estable y para cada grupo fosfato adicional el nucleótido se volverá más inestable, así el enlace del fósforo y fosfato libera energía al romperse por hidrólisis. El grupo fosfato le otorga al núcleo un enlace de alta energía, por lo que son tomados como fuentes para la transferencia energética por parte de las células.

Los nucleótidos esencialmente son ensamblados de uno en uno por la célula y después se agrupan de diferentes formas que dependen del ácido nucleico, en el proceso de replicación en el caso del ADN o en el llamado proceso de transcripción o producción del ARN. Los nucleótidos entonces tendrán la siguiente estructura:

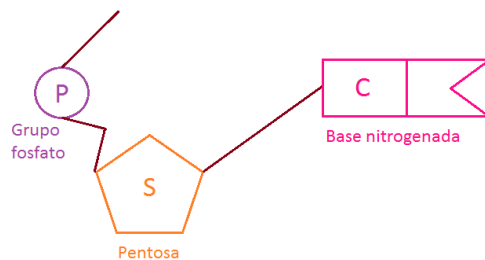


Figura 34: Estructura del Nucleótido

Los nucleótidos tienen características importantes, una de las más importantes es la forma en la que se unen las diferentes bases nitrogenadas, estas cumplen con una regla que especifica que la base Ademina (*A*) solamente se une con la base Timina (*T*) a través de un puente de hidrógeno doble y la base Guanina (*G*) se une a la base Citocina (*C*) mediante un puente triple de hidrógeno esta regla se cumple para el ADN mientras que para el caso del ARN la base Timina se cambia por la base Uracilo (*U*) y el resto de la regla de conexión permanece igual que en el ADN.

5.1. Función y estructura de los ácidos Nucleicos

Los ácidos nucleicos son las biomoléculas portadoras de la información genética y son los responsables de su transmisión hereditaria en todos los organismos. Son biopolímeros de elevado peso molecular los cuales estarán formados por otras subunidades estructurales (monómeros) que se repiten los cuales son llamados nucleótidos. Los ácidos nucleicos tienen dos funciones importantes una es almacenar la información genética para luego transformarla en proteínas y la otra es que sirven como almacén de energía, por lo tanto las secuencias de estas moléculas en el polímero puede transmitir órdenes como: “haz una proteína”, “replícame”, “trasládame al núcleo”, etc. Una de las cualidades de los ácidos nucleicos es que son moléculas muy estables, esto

es importante ya que para el proceso de transmisión de información genética de una célula a otra es necesario que la molécula no se deshaga por sí sola.

Los dos tipos importantes de ácidos nucleicos ADN y ARN se encuentran en todas las células procariotas, eucariotas y virus. El ácido desoxirribonucleico (ADN) codifica la información que la célula necesita para fabricar proteínas; El ADN funciona como el almacén de la información genética y se encuentra en los cromosomas del núcleo, las mitocondrias y los cloroplastos de las células eucariotas. En el caso de las células procariotas el ADN se encuentra en su único cromosoma, mientras que el ácido ribonucleico (ARN) presenta diversas formas moleculares y participan en la síntesis de las proteínas. El ARN interviene también en la transferencia de la información contenida en el ADN hacia los compartimientos celulares, este ácido nucleico se encuentra en el núcleo, el citoplasma, la matriz mitocondrial, el estroma de cloroplastos de células eucariotas y en el citosol de células procariotas.

Se sabe que los ácidos nucleicos constituyen el depósito de la información de todas las secuencias de aminoácidos de todas las proteínas de la célula, por lo que existe una relación colineal entre las secuencias de ácidos nucleicos y proteínas que es llamado ***Código genético***, la peculiaridad de esta relación establece que una secuencia de tres nucleótidos en un ácido nucleico corresponde a un *aminoácido* en una proteína.

5.1.1. Características del ADN y ARN

Tanto el ADN como el ARN son moléculas que se encuentran presentes en los organismos vivos para actividades esenciales, pero ¿Cuáles son las diferencias entre estos ácidos nucleicos? El ADN lleva la información necesaria para dirigir la *síntesis de proteínas* y la *replicación*. La mayoría de las moléculas de ADN poseen dos cadenas anti paralelas unidas entre sí mediante las bases nitrogenadas por medio de puentes de hidrógeno (bicatenaria), donde la Adenina enlaza con la Timina mediante dos puentes de hidrógeno, mientras que la Citocina enlaza con la Guanina mediante tres puentes de hidrógeno, dicha secuencia de nucleótidos en el ADN determina el orden de los aminoácidos en las proteínas. Por otro lado el ácido ribonucleico (ARN) es una molécula similar a la de ADN pero a diferencia del ADN el ARN es una cadena sencilla (monocatenaria); La hebra de ARN tiene un eje constituido por un azúcar (ribosa) y grupo de fosfato de forma alterna, unidos a

cada azúcar se encuentra una de las 4 bases nitrogenadas (Adenina, Uracilo, Citocina o Guanina) las cuales cumplen la siguiente regla de enlace entre estas bases nitrogenadas:



Hay diferentes tipos de ARN en la célula: **ARN mensajero (ARNm)**, **ARN ribosomal (ARNr)**, **ARN de transferencia (ARNt)** y **el ARN nucleolar (ARNn)**, estas biomoléculas son necesarias para el proceso de traducción de ADN a Proteína. Estos tipos de ARN son indispensables para procesos intermedios ya que estas moléculas llevan físicamente los aminoácidos al sitio donde se lleva a cabo la traducción y permiten que sean ensamblados en las cadenas de proteínas en dicho proceso. El ARN dirige la síntesis de aminoácidos, es decir, a diferencia del ADN no almacena directamente la información si no que funciona como puente para pasar de una cadena de ácidos nucleicos a una cadena aminoácidos.

5.1.2. Replicación del DNA y formación del RNA nucleolar

La replicación del ADN es un proceso mediante el cual se duplica una molécula de ADN, cuando una célula se divide debe duplicar su genoma para que cada célula hija contenga un juego completo de cromosomas. La replicación del ADN es probablemente uno de los procesos más impresionantes que realiza el ADN ya que cada célula contiene todo el ADN que necesita para fabricar las demás células, este proceso es tan impresionante que nosotros mismos comenzamos siendo solo una célula y terminamos con billones de ellas y durante ese proceso de división celular toda la información de una célula tiene que ser copiada a la perfección a las otras células, eso es sorprendente teniendo en cuenta que hay casi tres mil millones de pares de bases de ADN para ser copiadas. La replicación del ADN utiliza polimerasas, que son moléculas dedicadas específicamente solo a copiar ADN.

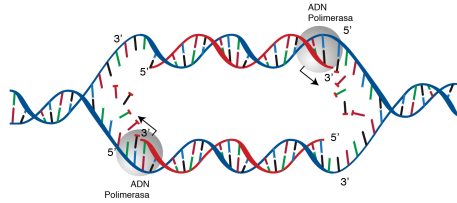


Figura 35: Replicación del ADN

El proceso de replicación del ADN (Figura 35) es el mecanismo que permite al ADN sintetizar una copia idéntica, esta duplicación del material genético se produce de acuerdo con un mecanismo *semiconservador* (se dice que es semiconservador por que en cada una de las moléculas hijas se conserva una de las cadenas originales), lo que indica que los dos polímeros complementarios del ADN original al separarse sirven de molde cada una para la síntesis de una nueva cadena complementaria de la cadena molde, de esta forma cada nueva doble hélice contiene una de las cadenas del ADN original. Gracias a la complementación de las bases que forman la secuencia de cada una de las cadenas de ADN, este tiene la importante propiedad de reproducirse idénticamente lo que permite que la información genética se transmita de una célula madre a las células hijas y es la base de la herencia del material genético.

En el proceso de replicación la molécula de ADN se abre como una cremallera por la ruptura de los puentes de hidrógeno entre las bases complementarias en puntos determinados llamados los orígenes de replicación. Las proteínas iniciadoras reconocen secuencias de nucleótidos específicas en esos puntos y facilitan la fijación de otras proteínas que permitirán la separación de las dos hebras de ADN formándose una horquilla de replicación. El **ADN polimerasa** añade los nucleótidos complementarios a cada cadena original, es decir a las plantillas que salen del desenrollamiento del ADN. En la mayoría de los casos de replicación es unidireccional, no obstante la replicación se puede considerar de forma general como bidireccional. En el proceso de replicación del ADN existe un mecanismo de corrección (función correctora) que modifica los errores producidos durante la copia del ADN los cuales darían lugar a mutaciones.

5.1.3. Tipos y funciones del ARN

Existen dos tipos de ARN que están implicados en la síntesis de proteínas, estos son los *codificantes* y los *no codificantes*, estos serán el ARN mensajero, el ARN de transferencia y el ARN ribosomal. El ARN codificante será el ARN mensajero (ARNm) el cual se encarga de llevar la información del ADN a los ribosomas, ya que la secuencia de nucleótidos del ARNm determinará la secuencia de aminoácidos de la proteína. Por otro lado los ARN no codificantes serán el ARN de transferencia y el ARN ribosomal, el ARN no codificante se origina a partir de genes propios, es decir intrones ⁵ rechazados durante el proceso de splicing ⁶. El ARN de transferencia (ARNt) y el ARN ribosómico (ARNr) son elementos fundamentales en el proceso de *traducción*. Ciertos ARN no codificantes denominados ribozimas son capaces de catalizar reacciones químicas como cortar y unir otras moléculas de ARN o formar enlaces peptídicos entre aminoácidos en el ribosoma durante la *síntesis de proteínas*.

Como fue mencionado estudiaremos cuatro tipos de ARN, estos son:

ARN mensajero (ARNm): Consiste en una molécula lineal de nucleótidos cuya secuencia de bases es complementaria a una porción de la secuencia de bases del ADN, el ARNm dicta con exactitud la secuencia de aminoácidos en una cadena polipeptídica, las instrucciones para la construcción de dichas cadenas polipeptídicas residen en tripletes de bases a las que llaman codones. Estas cadenas de ribonucleótidos están encargadas de llevar la información sobre la secuencia de aminoácidos de la proteína desde el ADN hasta el ribosoma, lugar en que se sintetizan las proteínas de la célula, es por eso que recibe el apelativo de “mensajero”. Esta molécula también determinará el orden en que se unirán los aminoácidos.

ARN de transferencia (ARNt): Esta molécula tiene aproximadamente 75 nucleótidos en su cadena y se pliega en una forma particular llamada

⁵Un intrón es una región del ADN que forma parte de la transcripción primaria de ARN, pero a diferencia de los exones, son eliminados del transcrito maduro, previamente a su traducción.

⁶Proceso mediante el cual los intrones, es decir, las regiones no codificadoras de los genes son separados del transcrito de ARN mensajero primario y los exones (es decir las regiones codificadoras) se unen para generar ARN mensajero maduro.

hoja de trébol plegada. El ARNt se encarga de transportar los aminoácidos libres del citoplasma al lugar de síntesis proteica. Para trasladar estos aminoácidos a su posición específica es necesario que el ARNt presente un triplete de nucleótidos llamado anticodón, el cual se une a su codón complementario que estará presente en el ARNm, esta conexión será mediante puentes de hidrógeno como se observa en la Figura 36.

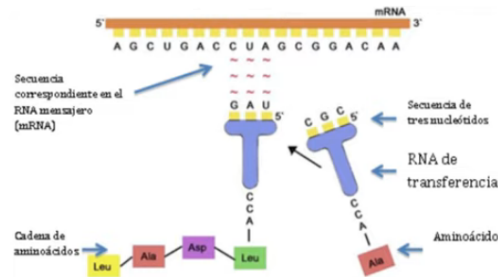


Figura 36: ARN de transferencia.

ARN ribosomal (ARNr): El ARNr es el componente catalítico de los ribosomas, se encarga de crear los enlaces peptídicos entre los aminoácidos del polipéptido en formación durante la síntesis de proteínas (actúan como ribozimas). Este tipo de ARN una vez transcrito pasa al núcleo donde se une a proteínas, de esta manera se forman las subunidades de los ribosomas. Para el proceso de síntesis del ARN ribosómico es necesario una molécula de ácido ribonucleico llamado ARN nucleolar.

ARN nucleolar (ARNn): Esta es una molécula larga formada por una secuencia de unos 13,000 nucleótidos que estará localizada y sintetizada en el núcleo de las células eucariotas a partir de la *transcripción* del ADN y que es indispensable para la síntesis de la mayor parte del ARNr. El ARNn se asocia a proteínas procedentes del citoplasma para formar las subunidades de los ribosomas.

5.2. Síntesis de Proteínas

Uno de los procesos más importantes en los seres vivos es la transferencia de información genética, el proceso por el cual esta información es transferida del ADN a las proteínas se representa en la Figura 37 y es llamado *Síntesis*

de Proteínas. Se conoce como síntesis de proteínas al proceso por el cual se componen nuevas proteínas a partir de los veinte aminoácidos esenciales. En este proceso se transcribe el ADN en ARN y posteriormente en aminoácidos, que son las unidad más pequeña de las proteínas, las cuales al agruparse formaran proteínas. En el proceso de síntesis los aminoácidos son trasportados por el ARNt correspondiente para cada aminoácido hasta el ARNm donde se unen en la posición adecuada para formar las nuevas proteínas. La síntesis de proteínas se realiza en los ribosomas situados en el citoplasma celular. Al finalizar la síntesis de una proteína se libera el ARNm y este puede volver a ser leído, incluso antes de que la síntesis de la proteína anterior termine ya puede comenzar la siguiente síntesis, por lo cual el mismo ARNm puede utilizarse por varios ribosomas.



Figura 37: Esquema del proceso de la síntesis de proteínas.

Como se observará en secciones posteriores este proceso se realiza a través de cambios de lenguajes, es decir se tendrá una decodificación de mensajes genéticos ya que la información que sale del ADN solo cambiará de lenguaje pero no se perderá la información en estos cambios. Dichos cambios de lenguaje se presentan en dos etapas: la *Transcripción* y la *Traducción*, los cuales se describirán con más detalle a continuación.

5.2.1. Transcripción Genética

El ADN se encuentra en el núcleo pero la mayoría de su información debe de salir del núcleo para ser expresada y esto sucederá gracias al ARN ya que una de sus funciones es ser el mensajero que lleve esta información fuera del núcleo para ser traducido a una proteína, este proceso es llamado *Transcripción*. La Transcripción del ADN es el primer proceso de la expresión genética mediante el cual se transfiere la información contenida en las secuencias de ADN hacia las secuencias de proteínas utilizando diversos ARN intermedarios. Durante la transcripción genética la secuencia de ADN es copiada sin pérdida de información a una secuencia de ARN mediante una enzima llamada **ARN polimerasa (ARNp)**, la cual utiliza un molde de ADN de cadena

sencilla para sintetizar una cadena complementaria de ARN agregando un nuevo nucleótido al extremo 3' de la cadena (Figura 38).

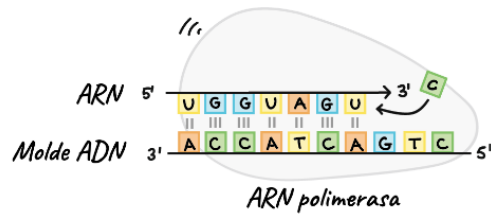


Figura 38: ARN polimerasa.

La ARN polimerasa sintetiza una cadena de ARN complementaria a la cadena molde de ADN, esta enzima sintetiza la cadena de ARN en dirección 5' a 3', mientras que lee la cadena molde de ADN en dirección 3' a 5'. La cadena molde de ADN y la cadena de ARN son anti paralelas. La enzima avanza a lo largo de la cadena molde en dirección 3' a 5' y al avanzar abre la doble hélice del ADN, así el ARN sintetizado solo se mantiene unido a la cadena molde por un corto tiempo y luego sale de la polimerasa como una cadena colgante para permitir que el ADN se vuelva a cerrar y formar una doble hélice.

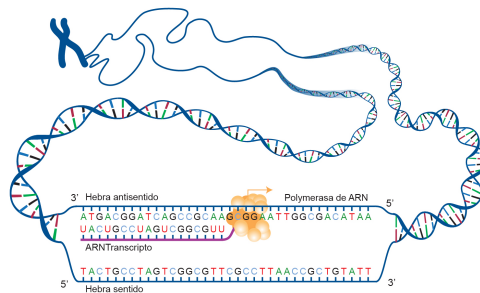


Figura 39: Cadenas de ADN con su respectivo ARN Transcrito o ARNm.

En el proceso de transcripción una región de ADN se abre y una sola cadena (cadena molde) sirve como plantilla para la síntesis de un transcrito complementario de ARN (ARNm), esta cadena será idéntica a la cadena de

ADN excepto que el ARN tiene base de Uracilo (U) en lugar de base Timina (T), este proceso se ilustra en la Figura 39. Para comprender mejor este proceso veamos un ejemplo.

Ejemplo 5.1: Tomemos una secuencia de ADN la cual está conformada de dos cadenas a las cuales llamaremos codificante y molde. En la transcripción lo que se hace es tomar una hebra de la doble hélice (como se observa en la Figura 40).

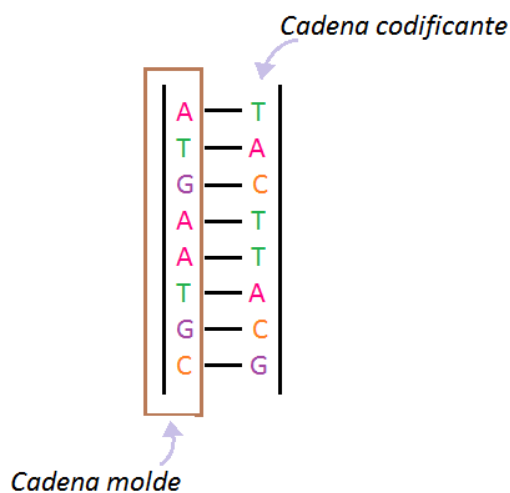


Figura 40: Segmento de una cadena de ADN.

En el proceso de transcripción se toma una cadena del ADN y se construye una secuencia de ARN, donde este ARN será el ARN mensajero ya que este extraerá la información del núcleo. Este proceso es muy parecido al proceso de replicación en el cual el nucleótido Uracilo se cambia por el nucleótido Timina como se observa en la Figura 41.

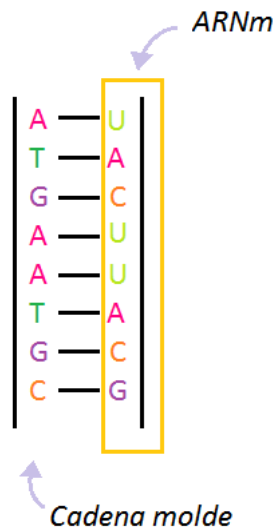


Figura 41: Secuencia de ARNm.

De tal forma que esta molécula puede separarse para así tener una cadena simple de ARNm que tendrá toda la información de la molécula de ADN. En el caso de un gen codificante el transcripto de ARN contiene la información necesaria para sintetizar un *polipéptido*⁷ con una secuencia de aminoácidos en particular. En este caso el transcripto de ARN que actúa como ARNm estará relacionado con el polipéptido: *Tyr – Leu – Arg*. Cabe mencionar que las células regulan cuidadosamente la transcripción de forma que solo se transcriben los genes cuyos productos son necesarios en un momento determinado. Cuando se transcriben diferentes genes no necesariamente se transcriben en cantidades iguales, es decir, se produce un número diferente de moléculas de ARN de cada gen si así lo necesita el organismo.

5.2.2. Traducción Genética

La traducción es el segundo proceso de la síntesis proteica el cual ocurre en todos los seres vivos. En la traducción el ARN mensajero se decodifica para generar una cadena específica de aminoácidos que formarán un polipéptido o una proteína, el cual se formará de acuerdo con las reglas específicas por el código genético, es necesario que la traducción venga precedida de un

⁷Un polipéptido es una subunidad de la proteína.

primer proceso de transcripción.

Las fases de la traducción son tres: iniciación, elongación y terminación durante los cuales se va dando el crecimiento del polipéptido o proteína. El código genético describe la relación entre la secuencia de pares de bases en un gen y la secuencia correspondiente de aminoácidos que codifica. En el citoplasma de la célula el ribosoma lee la secuencia del ARNm en grupos de tres bases para ensamblar la proteína, dichos grupos de tres bases son llamados codones los cuales corresponden a un aminoácido.

Cabe mencionar que se debe de ser muy cuidadoso de la posición donde se empieza a tomar el codón ya que si se toma en un punto incorrecto en la secuencia de ARNm el aminoácido correspondiente al codón en cuestión será erróneo. Una vez que se tienen los codones estos son emparejados con sus respectivos aminoácidos por un tipo específico de ARN, este es el ARN de transferencia (ARNt) el cual acerca estas dos moléculas (Figura 42).

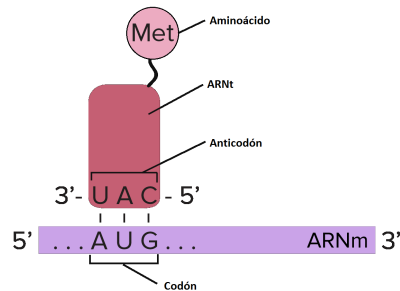


Figura 42: Unión del aminoácido con su codón correspondiente.

Cabe mencionar que cada codón estará unido a su anti codón el cual será la secuencia presente en el ARNt como se observa en la Figura 42. De tal forma que el aminoácido se va a pegar bajo el ARNt al extremo que tiene las bases complementarias (en este caso el codón *AUG* se unirá al anti codón *UAC*). Este proceso se realiza para cada codón de tal forma que conforme se agrupan los aminoácidos se formará una proteína. Una característica importante de mencionar es que el orden en el que se unen los aminoácidos determina la forma, propiedades y función de una proteína.

6. Extensión del Juego del Caos para Proteínas

Desde la primera aplicación del Juego del caos para secuencias genómicas en 1990, diversas investigaciones se han centrado en extraer características de las imágenes CGR y mostrar que dichas características pueden jugar un papel importante en el estudio de secuencias genómicas. Debido al gran potencial que este método presentaba para mostrar patrones subyacentes o sesgos hacia algunos nucleótidos en secuencias de ADN, en 1994 Fiser, Tusnády & Simon (Fiser y Tusnády, 1994) plantearon una extensión de la Representación del Juego del Caos de Jeffrey por primera vez, en este artículo se demuestra que la CGR puede ser extendida de tal forma que puede ser aplicada para visualizar y analizar las estructuras primarias y secundarias de proteínas.

La CGR puede ser generalizada para ser aplicable a secuencias de cualquier número de elementos base (es decir el alfabeto de la secuencia), de tal forma que es necesario cambiar el tablero de juego que fue presentado en la Sección 3.1, el cual era un cuadrado debido a que las secuencias de ADN o ARN tienen 4 bases ($A, C, G, T/U$); Para secuencias que tengan más bases el tablero del Juego del caos será un polígono regular de n -lados, donde n es el número de elementos que conforman el alfabeto de la secuencia.

El primer desarrollo de esta extensión planteaba un polígono regular de 20 lados donde cada vértice representaba un aminoácido (Fiser y Tusnády, 1994), pero una aplicación tan generalizada sufre de serias limitaciones, una de ellas es visualizar los patrones característicos de la CGR de diferentes Familias de proteínas, otra limitación se presenta en secuencias de aminoácidos homólogas pertenecientes a una familia en particular, ya que los residuos de aminoácidos en diferentes posiciones son a menudo remplazados por ***sustituciones conservativas*** manteniendo sus funciones invariantes, pero el uso de la CGR de 20 vértices no permite analizar de forma adecuada una secuencia de aminoácidos que presente sustituciones conservativas ya que la imagen resultante no mostrará las características específicas de la secuencia con dichas sustituciones, por el contrario mostrará características de secuencias de aminoácidos similares, es decir al tener secuencias homólogas las cuales presentarán sustituciones conservativas la imagen resultante podría no ser de la secuencia que se está estudiando, por lo cual fue necesario

modificar el método desarrollado por Fiser, Tusnady & Simon a un polgono de 12 lados y esto fue realizado por Basu, Pan Dutta & Das en su artculo *Chaos game representation of protein structure* (Basu et al., 1997), de esta forma secuencias de aminocidos homlogas pueden exhibir patrones teniendo entonces substituciones conservativas en las secuencias. Para el estudio de Familias de Protenas son usadas secuencias concatenadas de aminocidos de miembros pertenecientes a dicha familia.

En bioinformtica la CGR de 12 vrtices es usada para comparar secuencias de aminocidos usado el mtodo de *Alineacin de secuencias*, en el cual son remplazados aminocidos por otros que tengan propiedades bioqumicas similares como: carga, hidrofobicidad, tamano, etc., es decir, dado el pptido **IDEAL** (Isolucina, Aspartato, Glutamato Alanina y Leucina) puede ser remplazado por el pptido **MEEGL** (Metionina, Glutamato, Glutamato, Glicina y Leucina), pero no por el pptido **SREYL** (Serina, Arginina, Glutamato, Tirosina y Leucina). En el primer caso se ha hecho una *sustitucin conservativa* mientras que en el segundo caso se reemplazaron residuos por otro diferentes. Los cuales no tienen las mismas caractersticas.

Para poder entonces estudiar secuencias que presentan substituciones conservativas es necesario reducir el nmero de vrtices del polgono a 12 lados agrupando aminocidos ⁸ similares, por ejemplo: Alanina (*A*) y Glicina (*G*) estarn representados en un mismo vrtice, mientras que la Isoleucina (*I*), Leucina (*L*), Valina (*V*) y Metionina (*M*) estarn representados en otro vrtice, es decir, cada una de estas agrupaciones sern representadas por un vrtice como se observa en la Figura 43.

⁸En algunos artculos estos aminocidos son llamados *residuos*.

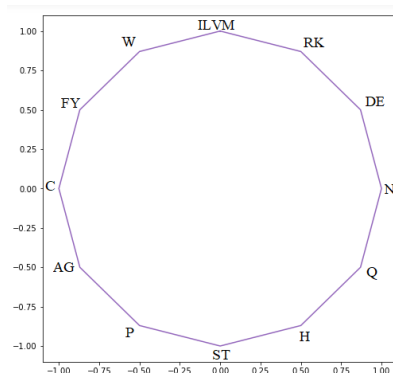


Figura 43: Tablero de juego para la CGR de 12 vértices.

La siguiente tabla muestra todos los aminoácidos y sus respectivos códigos de una letra los cuales son usados para etiquetar la CGR de 12 lados.

Tabla 4. Aminoácidos.

CLASE	AMINOÁCIDOS	CODIGO DE UNA LETRA
Alifático	Glicina, Alanina, Valina, Leucina, Isoleucina	<i>G, A, V, L, I</i>
Contenido de Hidroxilo o Azufre/Selenio	Serina, Cisteína, Selenocisteína, Treonina, Metionina	<i>S, C, U, T, M</i>
Cíclico	Prolina	<i>P</i>
Aromático	Fenilalanina, Tirosina, Triptófano	<i>F, Y, W</i>
Básico	Histidina, Lisina, Arginina	<i>H, K, R</i>
Ácido y sus amidas	Aspartato, Glutamato, Aspargina, Glutamina	<i>D, E, N, Q</i>

En el nuevo tablero de juego cada vértice del polígono representará a un grupo particular de *sustituciones conservativas* de residuos de aminoácidos permitiendo una representación pictórica de los patrones característicos de

familias de proteínas. Este etiquetado de los vértices del polígono son ordenados con respecto a la *hidrofobicidad normalizada decreciente* de los grupos de residuos en dirección anti horaria empezando con el grupo *ILVM* como se observa en la Figura 43. Este etiquetado permite la derivación de información sobre la relativa densidad de residuos hidrófobicos e hidrófilos en la secuencia primaria de proteínas de la CGR.

6.1. Algoritmo CGR de Proteínas

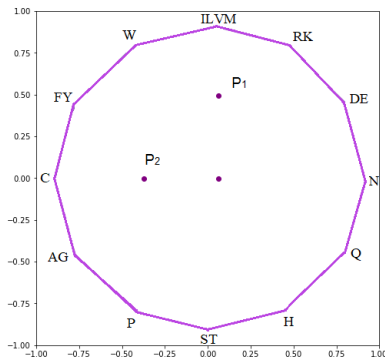
Como fue mencionado anteriormente este método es una extensión del método estudiado en la Sección 3 en el cual nuestro tablero será un polígono regular de 12 vértices, los cuales estarán representados matemáticamente por la expresión (88), donde $V_1(1, 0)$ es el primer vértice del polígono.

$$\begin{aligned} V_k(x) &= \cos \frac{k-1}{6} \pi \\ V_k(y) &= \sen \frac{k-1}{6} \pi \end{aligned} \quad \text{con } k = 2, 3, \dots, 12 \quad (88)$$

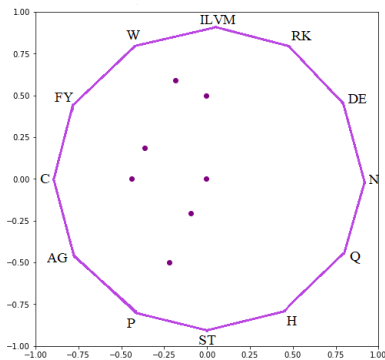
El algoritmo usado para el juego del caos es el mismo que se presentó en la Sección 3.1 en el cual se trazará el primer residuo de la secuencia a medio camino entre el centro del polígono y el vértice etiquetado con la primer letra (residuo) de la secuencia, el siguiente punto será trazado a medio camino entre el punto anterior y el vértice etiquetado con el siguiente residuo de la secuencia, este proceso se repetirá L veces para una secuencia de longitud L . Representando el IFS del Juego del Caos como fue expresado en sección 3.1.

$$\omega_i = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} \frac{V_{ix}}{2} \\ \frac{V_{iy}}{2} \end{pmatrix} \quad \text{con } i = 1, 2, 3, \dots, L \quad (89)$$

Veamos un ejemplo pictórico de este método, para esto graficaremos la CGR de la secuencia *MATWLS*. El primer residuo (*M*) de la secuencia debe ser graficado a medio camino entre el centro del polígono y el vértice etiquetado como “*ILVM*” como se ve en la Figura 44(a). El segundo residuo (*A*) será entonces graficado a medio camino entre el primer punto y el vértice etiquetado (*AG*), este proceso deberá repetirse hasta el último residuo de la secuencia (Figura 44 (b)).



(a)



(b)

Figura 44: Ejemplo de CGR para una secuencia corta.

Las Figuras (63 - 65) muestra la CGR de algunas proteínas en la cuales se pueden observar zonas de escasez y agrupación de puntos ⁹.

⁹Para ver más referencias dirigirse al Apéndice D

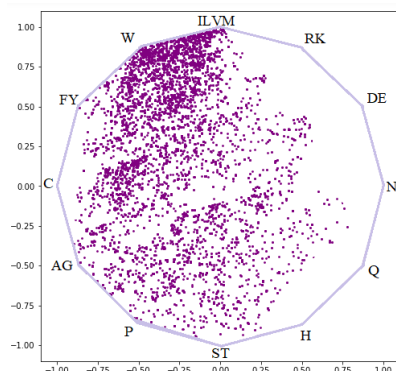


Figura 45: CGR de la Familia de Proteínas Hemoglobina.

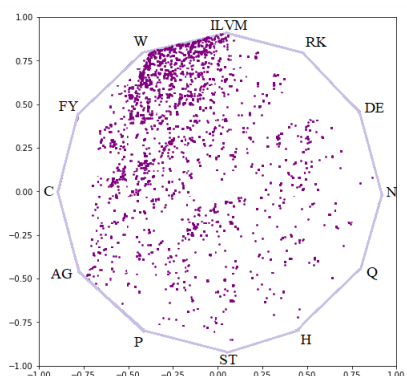


Figura 46: CGR de la Familia de Proteínas Lysozyme.

La distribución de puntos en la CGR vista en la Figura 63 no es aleatoria, en esta imagen es posible observar que se presenta un agrupamiento de puntos en algunas regiones del polígono, lo cual implica que en estas clases de proteínas la frecuencia de aparición del dipeptido mn (con $m, n = I, L, V, M, R, K, D$ y E) son mucho más altas que las de otros dipeptidos, por ejemplo en el caso de la Hemoglobina se presenta una agrupación a lo largo de la región que une los vértices $ILVM - W$.

Como fue mencionado un problema importante en la extensión de la técnica CGR para secuencias de proteínas individuales se refiere al número mínimo de residuos que podrían ser requeridos para generar patrones identificables. Se ha reportado que para secuencias de nucleótidos al menos se requieren

2000 bases para generar patrones (Basu et al., 1997). Pero en el caso del número de aminoácidos de la mayoría de proteínas sus secuencias son más pequeñas por lo que para solucionar este problema se han *concatenado secuencias de aminoácidos* que pertenezcan a una familia en particular y de esta forma el número de residuos será mayor de tal forma que se podrán generar patrones. Es importante mencionar que los patrones característicos de una familia de proteínas no dependen del número de secuencias sumadas (concatenadas) o del orden en que se concatenan. Además una vez generado el patrón, agregar más secuencias de proteínas de la familia no altera el patrón (Basu et al., 1997).

6.2. Conteo de rejilla de la CGR de Proteínas

El algoritmo para la CGR de proteínas permite una representación gráfica de patrones en secuencias de aminoácidos, sin embargo, para una correcta interpretación de tales patrones sería útil una caracterización matemática de estos, para lo cual dividimos todo el polígono en 24 segmentos que estarán etiquetados con los números del 1 al 24 en serie como se muestra en la Figura 47, esta cuantificación se le llama *Conteo de rejilla (Grid Counts)*. Esta es una estimación del porcentaje de puntos dibujados en diferentes regiones de la CGR (grid points), permitiendo así la cuantificación de la no aleatoriedad de los patrones.

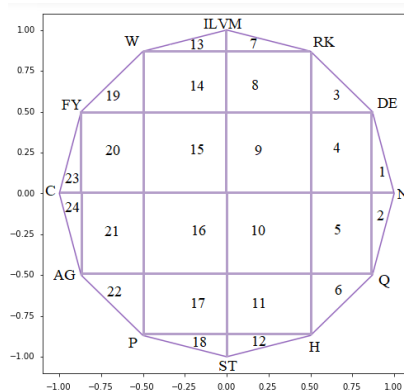


Figura 47: Rejilla de la CGR de 12 vértices

El grid counts de la CGR de cualquier familia de proteínas depende solo de la posición relativa de diferentes grupos de residuos a lo largo de sus

vértices y no del número u orden de las proteínas concatenadas (Basu et al., 1997), de esta forma se describe una medida cuantitativa de las preferencias si las hay en la CGR. En la Figura 47 se muestran algunos ejemplos.

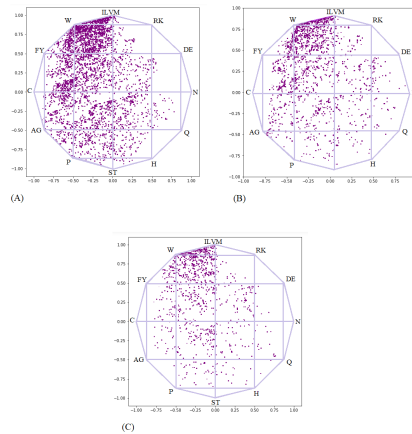


Figura 48: Ejemplos de Grid counts.

Para cada segmento un contador (C_i para el j -ésimo segmento) se pone en cero y se inicia en cero. Como cada punto es graficado dentro de la CGR el contravalor del segmento en que el punto vive incrementará en 1 manteniendo todos los demás contravalores sin cambios. Por ejemplo si un punto es graficado en el k -ésimo segmento, entonces $c_k = c_k + 1$ y $c_j = c_j (j \neq k)$. El porcentaje de puntos que caen en diferentes segmentos de la cuadrícula son calculados tal que:

$$G_j = \left(\frac{C_j}{N}\right) \times 100, \quad (j = 1, 2, 3, \dots, 24)$$

Donde N es el número total de residuos en la secuencia graficada y G_j representa el porcentaje de puntos graficados dentro o en cualquier límite de la j -ésima región.

6.3. Aplicación del método CGR de 12 vértices

La representación del Juego del Caos para Proteínas es un campo nuevo de investigación por lo que las aplicaciones de este método son hoy en día

un campo de investigación en crecimiento. Algunas de las principales aplicaciones de este método es en la comparación o clasificación de familias de proteínas. Pero una de las aplicaciones más interesantes hoy en día es la aplicación de este método para la predicción de interacciones ARN-Proteína (RPIs) la cual juega un papel importante en una amplia gama de regulaciones postranscripcionales tal como el empalme de ARN, transporte de ARN, replicación de ARN y traducción de ARNm. Identificar si un determinado par de ARN-Proteína puede o no formar interacciones es un requisito previo vital para diseccionar los mecanismos reguladores de los *ARN funcionales* [14].

Esta es un área brevemente estudiada debido a que no se cuentan con suficientes herramientas computacionales. En el artículo *Prediction of RNA-protein interactions using conjoint triad feature and chaos game representation* de Hongchu, Wang & Pengfe es usado este método extendido del Juego del Caos de Jeffrey junto al llamado Conjoint Triad Feature (CTF), los cuales proporcionan un grupo fundamental de características las cuales son usadas para determinar la presencia de interacción ARN-Proteína.

Para la $CGR_{proteínas}$ el conjunto de características contiene 24 características las cuales están representadas por la cuadrícula presenta en la sección anterior, mientras la CGR_{ARN} proporciona 16 características las cuales están representadas por la cuadrícula de la CGR para secuencias de ARN, estas características se observan en la siguiente figura.

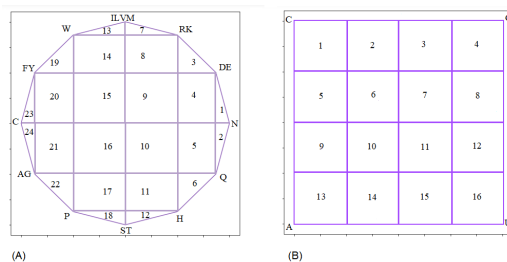


Figura 49: Tableros donde se representan las características de las CGR de proteínas y de ARN.

7. Fractales estrictamente autosimilares compuestos por polígonos estelares

El uso de polígonos para generar imágenes fractales es un algoritmo muy bien conocido (Sección 2), algunos ejemplos son el conjunto de Cantor, el cual está conformado por segmentos de recta, otro ejemplo es el triángulo de Sierpinski el cual está formado por un número infinito de triángulos o el Helecho de Barnsley el cual está formado por rectángulos, pero ¿qué pasará si tomamos un número arbitrario de vértices $n \in \mathbb{N}$, $n \geq 2$? En esta sección se discutirá un poco sobre este caso.

Existen diferentes algoritmos que pueden generar objetos fractales, en este trabajo se presentaron algunos de estos algoritmos como lo son la Máquina Copiadora de Reducción Múltiple (MRCM) y el Juego del Caos, pero existe un algoritmo con el cual se pueden generar fractales estrictamente autosimilares que estarán conformados por **polígonos estrella**, estos son polígonos no convexos que estarán caracterizados por $\{n/m\}$ con $m, n \in \mathbb{N}$, donde, $n \geq 2$ es el número de vértices del polígono y $m \leq \frac{n}{2}$ indica cada cuantos vértices se unirá el vértice anterior con el siguiente.

Veamos un ejemplo de polígono estrella, para esto tomemos un círculo unitario y dividámoslo en n segmentos iguales, en este caso $n = 7$, y se unirá el primer punto con el segundo inmediato, de tal forma que al repetir este proceso el polígono se cerrará por completo (Figura 50 (a)), pero también es posible unir el primer punto con el tercero inmediato (Figura 50(b)), de esta forma podemos ver que hay diferentes formas de conectar los vértices entre sí, los cuales generan diferentes polígonos estrella.

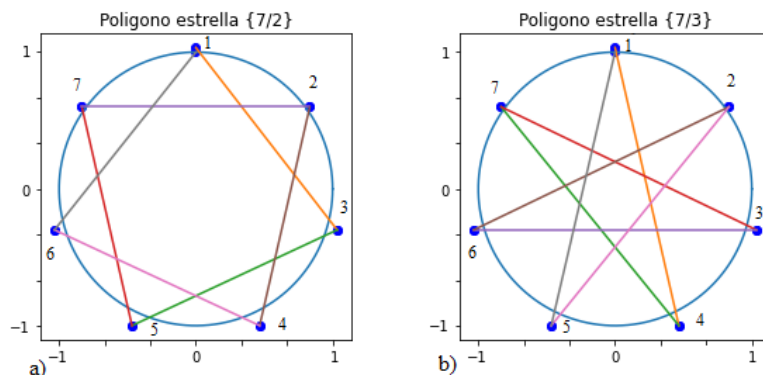


Figura 50: Polígono estrella $n = 7$.

7.1. Algoritmo Tzanov

En este trabajo fueron estudiados dos algoritmos importantes para la generación de fractales, en los cuales el factor de contracción P no es muy estudiado, pero este es un elemento muy importante en la generación de fractales al igual que el valor m en cual indica vértices con el que se conectará el vértice anterior.

Una vez que se ha elegido un polígono- $\{n/m\}$ inicial este puede ser escalado por un factor $P \in (0, 1)$, este nuevo polígono se copiará n veces y se colocará en cada uno de los vértices del polígono semilla. Si repetimos este proceso para cada uno de estos nuevos polígonos tendremos n^2 polígonos los cuales serán P^2 veces más pequeños que el polígono- $\{n/m\}$ inicial circunscrito en el círculo unitario.

El valor de P debe ser elegido cuidadosamente para que después de infinitas aplicaciones el resultado sea un fractal estrictamente auto similar conformado por n polígonos estrella que NO se intersectan. Dado que P es un factor de escalamiento se cumple el siguiente teorema.

Teorema 7.1

Dados los mapeos $S_1, \dots, S_n: \|S_i(x) - S_i(y)\| \leq c_i \|x - y\|$, entonces \exists un único conjunto no vacío $F : F = \cup_{i=1}^n S_i(F)$, por lo tanto invariante para el mapa S y $F = \cap_{k=1}^{\infty} S^k(E)$ (Tzanov, 2015)

Este teorema habla sobre el atractor del proceso de iteración, este atractor poligonal es un fractal producido por infinitas contracciones (n^i , cuando $i \rightarrow \infty$) del polígono inicial, es decir, estará compuesto por infinitos polígonos similares al inicial; pero una de las características fundamentales de estos fractales es que las copias reducidas del polígono inicial no deberán sobreponerse pero pueden tocarse entre si, esta condición restringe el valor de P . Este parámetro de escalamiento será deducido de los valores n y m y no dependerá del diámetro del círculo que contenga al polígono semilla.

7.1.1. Parámetros P y m

Para obtener la relación para $P(n, m)$ construyamos un polígono estrella $\{n/m\}$, tomaremos $m = 3$, la Figura 51 esboza este polígono con O_a el centro del círculo S_a , M es el punto de intersección de la secante $\overline{A_1A_4}$ y $\overline{A_3A_6}$, H es la proyección ortogonal de O_a en $\overline{A_1A_4}$ y L es la proyección ortogonal de O_a en $\overline{A_3A_4}$. En la imagen es posible observar que el segmento de línea $\overline{MA_4}$ será un segmento de línea del polígono estrella resultante después del escalamiento del polígono inicial con respecto al punto A_4 por el factor P .

Para conocer este valor de P es necesario analizar la Figura 51 (a). Dado que el polígono inicial está circunscrito en S_a , $\angle A_3O_aA_4 = 2\pi/n$ entonces $\angle LO_aA_4 = \frac{\pi}{2}$ ya que $\overline{A_3O_a} = \overline{A_4O_a}$. Por otro lado $\angle A_1O_aA_3 = \frac{4\pi}{n}$ entonces $\angle A_1A_4A_3 = \frac{2\pi}{n}$ y $\angle A_1O_aH = \frac{3\pi}{n}$, ya que S_a con el centro en O_a circunscribe A_1 , A_3 y A_4 . Ahora podemos deducir $\overline{A_1A_4}$ y $\overline{A_4M}$ por el radio r de S_a , ya que $r = \overline{O_aA_i}$ para $i = 1, \dots, n$, entonces $\overline{A_1A_4} = 2r \operatorname{sen}(\frac{3\pi}{n})$. Para deducir $\overline{A_4M}$ encontraremos primero $\overline{A_4L}$, donde $\overline{A_4L} = r \operatorname{sen}(\frac{\pi}{n})$ entonces $\overline{A_4M} = \frac{\overline{A_4L}}{\cos(\frac{2\pi}{n})} = \frac{r \operatorname{sen}(\frac{\pi}{n})}{\cos(\frac{2\pi}{n})}$. De la Figura 51 es posible observar que $P = \frac{\overline{MA_4}}{\overline{A_1A_4}}$, de esta forma sustituyendo los valores de $\overline{A_1A_4}$ y $\overline{MA_4}$ obtenemos el valor del factor de escalamiento para $m = 3$, el cual se muestra en la ecuación (90).

$$P(n, 3) = \frac{\overline{MA_4}}{\overline{A_1A_4}} = \frac{\operatorname{sen}(\frac{\pi}{n})}{2\cos(\frac{2\pi}{n})\operatorname{sen}(\frac{3\pi}{n})} \quad (90)$$

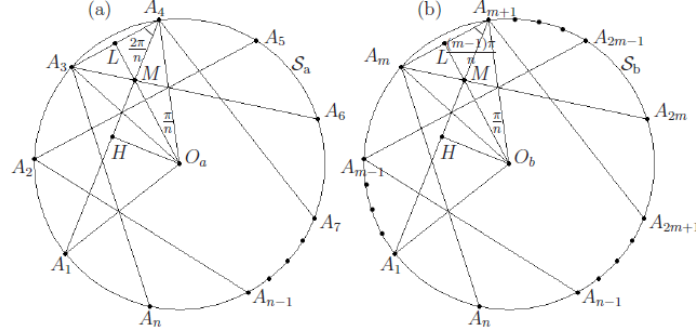


Figura 51: Bocetos de polígonos estrella $\{n/3\}$ y $\{n/m\}$ circunscritos en S_a y S_b respectivamente (Tzanov, 2015).

Este factor de escalamiento fue encontrado para un m específico, pero este procedimiento puede ser generalizado para cualquier m que cumpla la condición $1 \leq m \leq \frac{n}{2}$. En la Figura 51 (b) se observa el boceto de un polígono entrella- $\{n/m\}$ donde los puntos A_i (para $i = 1, \dots, n$) son los vértices del polígono estrella y O_b es el centro del círculo circunscrito S_b , con radio r , de tal manera que M es el punto de intersección de la secante $\overline{A_1A_{m+1}}$ y $\overline{A_mA_{2m}}$, H es la proyección ortogonal de O_b en $\overline{A_1A_{m+1}}$ y L es la proyección ortogonal de O_b en $\overline{A_mA_{m+1}}$. En la Figura 51 (b) observamos nuevamente que $P = \frac{\overline{MA_{m+1}}}{\overline{A_1A_{m+1}}}$.

Dando el mismo tratamiento que en el ejemplo anterior $\angle LO_bA_{m+1} = \frac{\pi}{n}$ ya que $\overline{A_mO_b} = \overline{A_{m+1}O_b}$, por otro lado podemos obtener que $\angle A_1O_bA_m = \frac{(2m-2)\pi}{n}$ entonces $\angle A_1A_{m+1}A_m = \frac{(m-1)\pi}{n}$ y $\angle A_1O_bH = \frac{m\pi}{n}$ ya que S_b circunscribe a A_1, A_m y A_{m+1} . Con esta información es posible deducir entonces los segmentos $\overline{A_1A_{m+1}}$ y $\overline{A_{m+1}M}$, obteniendo que $\overline{A_1A_{m+1}} = 2r \operatorname{sen}\left(\frac{m\pi}{n}\right)$.

Para deducir $\overline{A_{m+1}M}$ es necesario primero encontrar el el valor para el segmento $\overline{A_{m+1}L}$. La Figura 51 (b) muestra que $\overline{A_{m+1}L} = r \operatorname{sen}\left(\frac{\pi}{n}\right)$, de esta forma encontramos que $\overline{A_{m+1}M} = \frac{\overline{A_{m+1}L}}{\cos\left(\frac{(m-1)\pi}{n}\right)} = \frac{r \operatorname{sen}\left(\frac{\pi}{n}\right)}{\cos\left(\frac{(m-1)\pi}{n}\right)}$. Sustituyendo los valores de $\overline{A_1A_{m+1}}$ y $\overline{MA_{m+1}}$ podemos encontrar el valor de contracción P para cualquier n y m , como se observa en la siguiente expresión.

$$P(n, m) = \frac{\overline{MA_{m+1}}}{\overline{A_1A_{m+1}}} = \frac{\operatorname{sen}\left(\frac{\pi}{n}\right)}{2\cos\left(\frac{(m-1)\pi}{n}\right)\operatorname{sen}\left(\frac{m\pi}{n}\right)} \quad (91)$$

Ya que deseamos evitar la autointersección de los conjuntos resultantes, se enunciará el siguiente Teorema:

Teorema 7.2

Dado un conjunto fractal estrictamente auto similar obtenido como un atractor IFS, donde n puntos de atracción se encuentran en S^1 , siendo estos los vértices de un polígono estrella- $\{n/m\}$, con un factor de atracción $P = P(n, m)$ (ecuación 91), entonces este conjunto fractal no es autointersectante si y solo si $m \in [\frac{n}{4}, \frac{n}{4+1}]$, lo que define de manera única a P para un n dado (Tzanov, 2015).

El teorema 7.2 previene la autointersección de los conjuntos resultantes especificando una condición para m , $\frac{n}{4} \leq m \leq \frac{n}{4} + 1$ ya que este valor juega un papel importante, este parámetro deberá tener un valor único a menos que n sea divisible entre 4 sin residuos. En este caso cuando n es divisible entre 4 sin residuos con $m \in [\frac{n}{4}, \frac{n}{4} + 1]$ implicará que la ecuación (91) producirá dos valores para P , para saber como serán estos valores tomemos $n = 4a$ para algún $a > 0, a \in \mathbb{N}$, usando la expresión (91):

$$\begin{aligned}
 P_1(n, m) : n = 4a, m = a & & P_2(n, m) : n = 4a, m = a + 1 \\
 P_1 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{2\cos(\frac{(a-1)\pi}{4a})\operatorname{sen}(\frac{a\pi}{4a})} & & P_2 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{2\cos(\frac{a\pi}{4a})\operatorname{sen}(\frac{(a+1)\pi}{4a})} \\
 P_1 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{\sqrt{2}\cos(\frac{(a-1)\pi}{4a})} & & P_2 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{\sqrt{2}\operatorname{sen}(\frac{(a+1)\pi}{4a})} \\
 P_1 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{\sqrt{2}\cos(\frac{a\pi}{4a})\cos(\frac{\pi}{4a}) + \sqrt{2}\operatorname{sen}(\frac{a\pi}{4a})\operatorname{sen}(\frac{\pi}{4a})} & & P_2 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{\sqrt{2}\operatorname{sen}(\frac{a\pi}{4a})\cos(\frac{\pi}{4a}) + \sqrt{2}\cos(\frac{a\pi}{4a})\operatorname{sen}(\frac{\pi}{4a})} \\
 P_1 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{\cos(\frac{\pi}{4a}) + \operatorname{sen}(\frac{\pi}{4a})} & & P_2 = \frac{\operatorname{sen}(\frac{\pi}{4a})}{\cos(\frac{\pi}{4a}) + \operatorname{sen}(\frac{\pi}{4a})}
 \end{aligned} \tag{92}$$

Con estas condiciones podemos entonces generar fractales estrictamente autosimilares que no se intersectan así mismos.

7.2. Generación de fractales usando IFS

El factor de contracción y el factor m generarán atractores de IFS's, estos atractores pueden obtenerse usando la metáfora de la Máquina Copiadora de Reducción Múltiple (MRCM) o mediante el método del Juego del Caos, analizados en secciones anteriores, para esto primero definiremos una matriz

que especifique el número de puntos (n) a lo largo del círculo unitario, estos serán los vértices del polígono semilla. Se analizará primero el método usado en la MRCM, en la cual se toman conjuntos para generar los fractales, para comprender mejor este proceso analicemos un ejemplo.

Ejemplo 7.1: Polígono estrella de 5 vértices.

Tabla 5. Vértices del polígono $\{5/2\}$ circunscritos en el círculo unitario.

X	Y
0	1
0.9	0.3
0.6	-0.8
-0.6	-0.8
-0.9	0.3

Dada la condición $1 \leq m \leq \frac{n}{2}$, en este caso $m = 2$ de esta forma el primer punto se conectará con el segundo punto inmediato y este punto con el segundo inmediato, este proceso se repetirá hasta cerrar por completo el polígono estrella, como se observa en la siguiente figura.

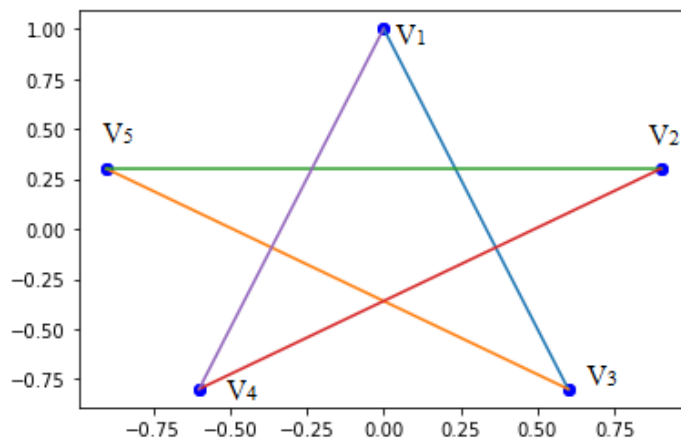


Figura 52: Polígonos semilla de 5 vértices.

Este polígono será el polígono semilla, para este caso el valor del factor de contracción será $P(5, 2) = 0.382$. Dado este factor de contracción las copias reducidas del polígono semilla serán colocadas en cada uno de los vértices del polígono semilla, después esta imagen resultante será reducida por el factor de contracción P , esta imagen resultante será copiada nuevamente y se colocará en los vértices del polígono semilla, este proceso se repetirá las veces que deseemos para generar el fractal, en la Figura (53) podemos visualizar la primera, segunda y cuarta iteración respectivamente de este proceso.

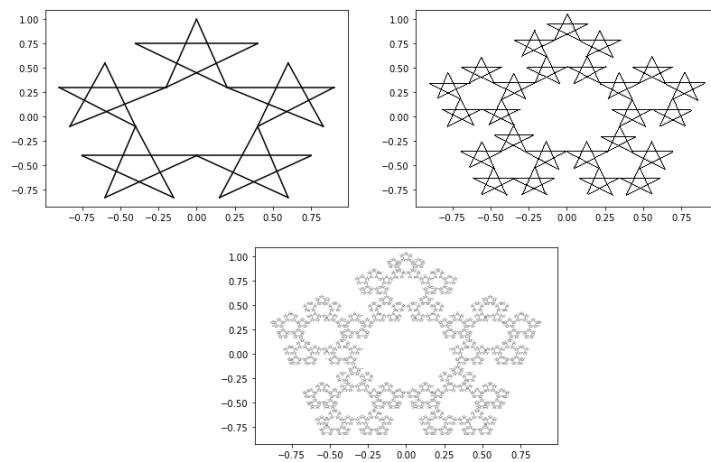


Figura 53: Primera, segunda y cuarta iteración del IFS con polígono estrella inicial $\{5/2\}$.

El Juego del Caos también es un algoritmo descrito por un Sistema de Funciones Iteradas al igual que el método de la MRCM, este método genera un subconjunto matemático en el plano que estará definido como el atractor. Como fue mencionado en la sección 3, para este algoritmo es necesario tener un tablero de juego, en este caso el tablero de juego será el polígono estrella semilla.

El algoritmo nos dice que partiendo del centro del tablero tomaremos un dado cuyas caras corresponderán a cada uno de los vértices del polígono, dicho de otra forma tendremos un generador aleatorio de vértices, los cuales tendrán las mismas probabilidades. Una vez que un vértice se obtiene se dibujará una línea imaginaria entre el centro del tablero y el vértice que se obtuvo y trazaremos un punto, pero en lugar de ser el punto medio como

se estudió en la sección 3 este punto se trazará con proporción $P(n, m)$, el cual dependerá del valor de contracción del polígono estrella que se use como iniciador del proceso, este punto será ahora nuestro punto inicial, entonces nuevamente se generará un vértice aleatorio y se trazara un punto entre el punto anterior y el vértice aleatorio, pero este punto dependerá del valor de P . Tomemos el ejemplo anterior, donde $n = 5$, $m = 2$ y el factor de contracción es $P(5, 2) = 0.38$. En la Tabla 5 podemos observar los vértices del polígono estrella, este algoritmo puede escribirse como un conjunto de transformaciones afines, como se observa en las siguientes expresiones.

$$\begin{aligned}
\omega_1 = \omega_{V_1}(p_0) &= \begin{pmatrix} 0.38 & 0 \\ 0 & 0.38 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
\omega_2 = \omega_{V_2}(p_0) &= \begin{pmatrix} 0.38 & 0 \\ 0 & 0.38 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.9 \\ 0.3 \end{pmatrix} \\
\omega_3 = \omega_{V_3}(p_0) &= \begin{pmatrix} 0.38 & 0 \\ 0 & 0.38 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.6 \\ -0.8 \end{pmatrix} \\
\omega_4 = \omega_{V_4}(p_0) &= \begin{pmatrix} 0.38 & 0 \\ 0 & 0.38 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -0.6 \\ -0.8 \end{pmatrix} \\
\omega_5 = \omega_{V_5}(p_0) &= \begin{pmatrix} 0.38 & 0 \\ 0 & 0.38 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -0.9 \\ 0.3 \end{pmatrix} \tag{93}
\end{aligned}$$

La unión de estas transformaciones después de un número considerable de iteraciones nos dará como resultado un conjunto fractal (Figura 54).

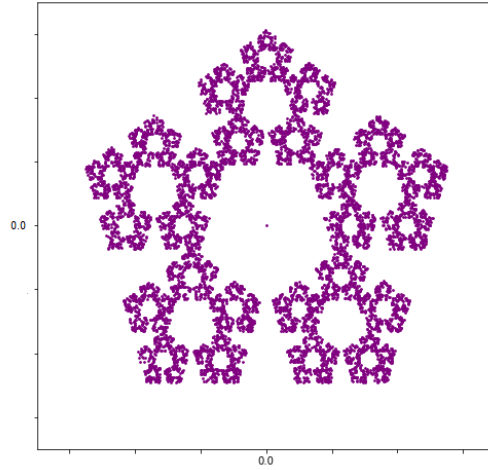


Figura 54: Conjunto fractal para un polígono inicial $\{5/2\}$ con 10000 puntos.

7.3. Dimensión fractal

La condición de no autointersección en estos conjuntos fractales $F_{\{n/m\}}$ es una condición indispensable que nos permitirá calcular la *Dimensión de Hausdorff* ($Dim_H F_{\{n/m\}}$) resolviendo la expresión (94):

$$\sum_{i=1}^n c_i^{dim_H F_{\{n/m\}}} = 1 \quad (94)$$

Donde n indica el número de mapeos de similitud S_i (Teorema 7.1) y c_i es la relación de contracción, estas relaciones serán del tipo $0 < c_i < 1$. Pero por otro lado sabemos que $\sum_{i=1}^n c_i = nP(n, m)$, de esta forma al sustituir esta expresión en (94) y despejando la dimensión de Hausdorff, obtenemos la siguiente expresión:

$$dim_H F_{\{n/m\}} = \frac{-\ln(n)}{\ln(P(n, m))} \quad (95)$$

Para comprobar esta expresión podríamos obtener la dimensión de Hausdorff para $n = 2$ y $m = 1$, estos valores definirán a una recta y cuya dimensión de Hausdorff será 1, como se esperaba. Como se ha mencionado en esta sección los valores de n y m juegan un papel importante en el estudio de estos

conjuntos fractales, el caso de la dimensión de Hausdorff no será la excepción, ya que conforme n tienda a infinito, $dim_H F_{\{n/m\}}$ tiende a 1. Dado que $F_{\{n/m\}}$ está inscrito en el mismo círculo en que el polígono- $\{n/m\}$ inicial está inscrito, conforme $n \rightarrow \infty$ el $F_{\{n/m\}}$ estará arbitrariamente cercano al círculo en que el polígono- $\{n/m\}$ inicial está inscrito.

7.3.1. Algoritmo Tzanov para Secuencias Proteicas

Hasta el momento la generación de estos conjuntos fractales se da debido a secuencias aleatorias, las cuales indican el orden en el que se aplicarán los mapeos S_i , pero nosotros estamos interesados en usar secuencias no aleatorias como lo son las secuencias proteicas. En la Sección 6 se presentó la extensión del Juego del Caos en la cual el tablero de juego era un polígono regular de 12 lados, en el que cada vértice representaba a un grupo, en los cuales los miembros de dichos grupos presentaban el mismo grado de *hidrofobicidad*, pero en las imágenes resultantes es posible ver que los puntos dentro de la CGR se sobreponen lo que impide un análisis completo de las secuencias usadas. Al hacer uso de los polígonos estrella este problema es eliminado ya que la condición fundamental del algoritmo Tzanov especifica que las copias reducidas del polígono semilla no se intercepten. La Figura (??) muestra el conjunto fractal para un polígono estrella con $n = 12$ y $m = 4$ para una secuencia aleatoria, en la cual todos los vértices del polígono inicial tienen las mismas probabilidades.

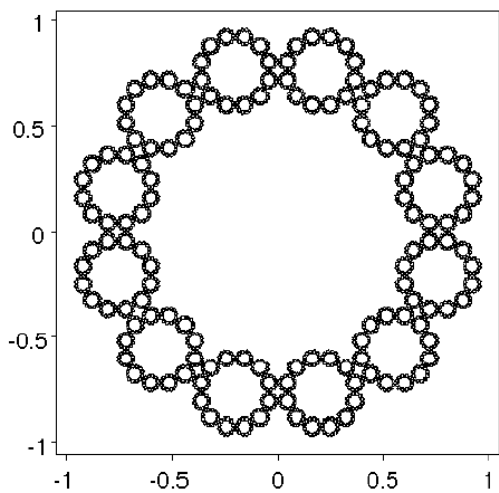


Figura 55: Conjunto fractal generado por IFS hecho de puntos que se encuentran en $F_{\{12/4\}}$.

En las siguientes figuras se observan algunas familias de proteínas, en las cuales se observa más claramente las proporciones de los aminoácidos en dichas familias. En este caso el factor de contracción sera $P(12, 4) = 0.2113$.

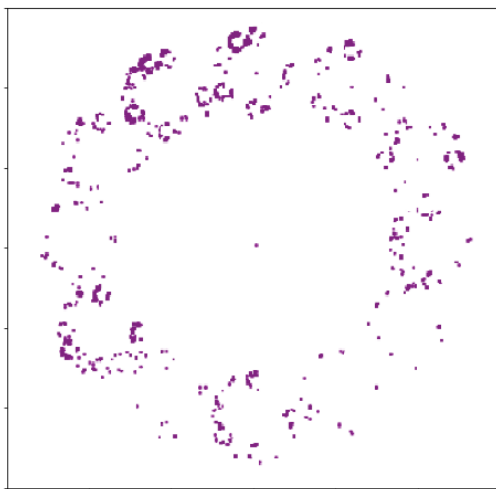


Figura 56: Conjunto fractal de la Familia de proteínas Lysozyme.

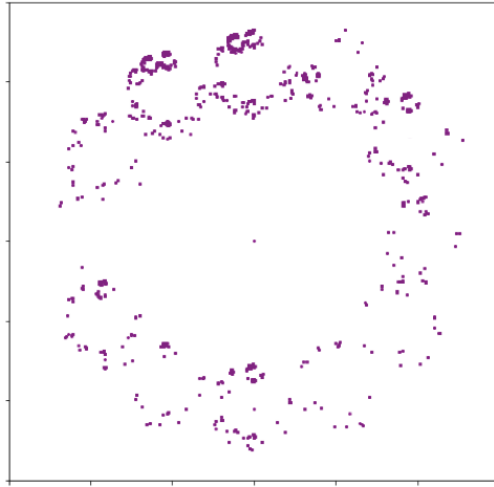


Figura 57: Conjunto fractal de la Familia de proteínas Rab.

En estas figuras es posible observar con mayor facilidad las zonas de escasez o sobrepoblación en las diferentes regiones del polígono estrella, de esta forma es posible observar cuales son los aminoácidos de mayor población en alguna familia de proteínas.

8. Conclusiones

La Representación del Juego del Caos es un método que puede ser aplicado en diversos campos de la ciencia siendo uno de ellos la Biología Molecular, donde este método muestra un gran potencial en la obtención de características subyacentes en secuencias de aminoácidos. En este trabajo se observaron algunas aplicaciones del Juego del Caos y como este algoritmo puede ser usado en secuencias genómicas, pero también fue posible observar las limitaciones que este método presenta al ser usado en secuencias proteicas, es por esto que fue implementado un nuevo algoritmo que generara firmas genómicas en las que los puntos de la grafica de dispersión no se sobrepongan, de tal forma que se puedan estudiar las características intrínsecas de dichas secuencias como la hidrofobicidad o los sesgos que presentan las familias de proteínas sobre algunos aminoácidos. Como pudimos observar el Algoritmo de Tzanov soluciona el problema de superposición presentado por el Algoritmo del Juego del Caos clásico, ya que las imágenes CGR resultantes de este método solo muestran cúmulos de puntos y al hacer una caracterización estadística, la rejilla no es la adecuada ya que algunas regiones son más grandes que otras. Al usar polígonos estrella como cubiertas del fractal resultante de la iteración del IFS presentado en la Sección 7 con un factor de contracción $P(12, 4) \approx 0.211324865$, nos ayuda a que los puntos de la secuencia proteica no se sobrepongan debido a la condición de no autointersección de las cubiertas, esta condición no solo nos ayuda a observar los patrones presentes en las secuencias de proteínas sino que nos ayudará a asignarle una dimensión fractal a los conjuntos resultantes de cada familia de proteínas, pero este trabajo será presentado en trabajos futuros.

La importancia de este trabajo es mostrar que este es un nuevo campo de investigación en el cual las características obtenidas de las CGR's pueden ayudar al estudio de procesos biológicos como la replicación de ARN, empalme de ARN o traducción de ARNm los cuales son procesos importantes en los organismos o puede ser usado para estudiar las interacciones ARN-Proteína (Wang y Wu, 2018).

Por otro lado cabe mencionar que aunque el algoritmo de Tzanov elimina una de las limitaciones presentadas por el Algoritmo del juego del caos para proteínas, aun se presentan limitaciones en este algoritmo, uno de ellos es que este método solo puede ser aplicado a familias de proteínas y no a proteínas sola, debido al número de puntos que debe tener una firma genómica para presentar patrones visibles. Esta será una de las incógnitas que se desea

abordar en trabajos futuros.

9. Perspectivas a futuro

Debido a que este campo de investigación se encuentra en crecimiento las proyecciones a trabajos futuros son muy amplias, debido al proceso de investigación realizado para este trabajo se presentan algunas proyecciones de trabajos futuros que muestran gran potencial debido a las nuevas herramientas computacionales, las cuales son usadas en trabajos de actualidad.

1. Caracterización de familias de proteínas por su dimensión fractal.
2. Clasificación de proteínas mediante sus imágenes CGR y el uso de redes neuronales.
3. Predicción de interacciones ARN-Proteína.

10. Apéndice A

10.1. Dimensión del conjunto de Cantor

Partiendo de la construcción del conjunto de Cantor, el paso $n + 1$ describe el comportamiento del tamaño del diámetro de las esferas necesarias para adoquinar cada paso y el número de esferas necesarias para ello, esto puede ser visto en la siguiente tabla para algunos pasos.

Tabla 1.

Número de paso	Imagen	Diámetro	Número de esferas
1	—————	$\varepsilon = \left(\frac{1}{3}\right)^0$	$N(\varepsilon) = (2)^0$
2	— — —	$\varepsilon = \left(\frac{1}{3}\right)^1$	$N(\varepsilon) = (2)^1$
3	- - - - -	$\varepsilon = \left(\frac{1}{3}\right)^2$	$N(\varepsilon) = (2)^2$
n+1	- - - - -	$\varepsilon = \left(\frac{1}{3}\right)^n$	$N(\varepsilon) = (2)^n$

Tomaremos ahora el logaritmo natural del Diámetro y del Número de esferas para el n -ésimo paso:

$$\ln(\varepsilon_n) = \ln\left(\frac{1}{3}\right)^n \quad \ln N(\varepsilon_n) = \ln(2)^n$$

$$\ln(\varepsilon_n) = \ln n\left(\frac{1}{3}\right) \quad \ln N(\varepsilon_n) = \ln n(2)$$

Dividiendo estas expresiones:

$$\frac{\ln N(\varepsilon_n)}{\ln(\varepsilon_n)} = \frac{\ln(2)}{\ln\left(\frac{1}{3}\right)}$$

$$\ln N(\varepsilon_n) = \frac{\ln(2)}{\ln\left(\frac{1}{3}\right)} \ln(\varepsilon_n)$$

$$\ln N(\varepsilon_n) = -\frac{\ln(2)}{\ln(3)} \ln(\varepsilon_n)$$

Si nombramos $D = \frac{\ln 2}{\ln 3}$, entonces la expresión anterior puede ser escrita como:

$$\ln N(\varepsilon_n) = -D \ln(\varepsilon_n)$$

$$N\varepsilon_n = \varepsilon_n^{-D} \tag{96}$$

Ahora de la expresión para la medida de Hausdorff:

$$H_\alpha = \lim_{\varepsilon \rightarrow 0} N(\varepsilon) \varepsilon^\alpha \tag{97}$$

Y como podemos ver regresamos a la expresión (7) para la Medida de Hausdorff, teniendo entonces tres casos posibles:

$$\alpha > D \quad \rightarrow \quad H_\alpha = 0$$

$$\alpha < D \quad \rightarrow \quad H_\alpha = \infty$$

$$\alpha = D \quad \rightarrow \quad H_\alpha = A$$

Obteniendo así la dimensión fractal del Conjunto de Cantor ya que $D = \frac{\ln 2}{\ln 3}$, entonces $\alpha = \frac{\ln 2}{\ln 3} = 0.63093$.

11. Apéndice B

11.1. Dimensión del conjunto de Koch

Se tomará el mismo procedimiento que se aplicó en el conjunto de Cantor, para esta curva el diámetro de las esferas que adoquinan al conjunto será $\varepsilon_n = (\frac{1}{3})^n$ y el número de esferas para adoquinar será $N(\varepsilon_n) = (4)^n$ tomando el logaritmo natural de ambas expresiones:

$$\ln(\varepsilon_n) = n \ln\left(\frac{1}{3}\right)$$

$$\ln N(\varepsilon_n) = n \ln(4)$$

Y dividiéndolas:

$$\frac{\ln N(\varepsilon_n)}{\ln(\varepsilon_n)} = -\frac{\ln(4)}{\ln(3)}$$

$$\ln N(\varepsilon_n) = -\frac{\ln(4)}{\ln(3)} \ln(\varepsilon_n)$$

Entonces:

$$N\varepsilon_n = \varepsilon_n^{-D}$$

Obteniendo la dimensión del conjunto para la cual la medida de Hausdorff es finita.

$$\alpha = D = \frac{\ln(4)}{\ln(3)} = 1.2618$$

12. Apéndice C

12.1. Imágenes Fractales aplicando el Operador de Hutchinson

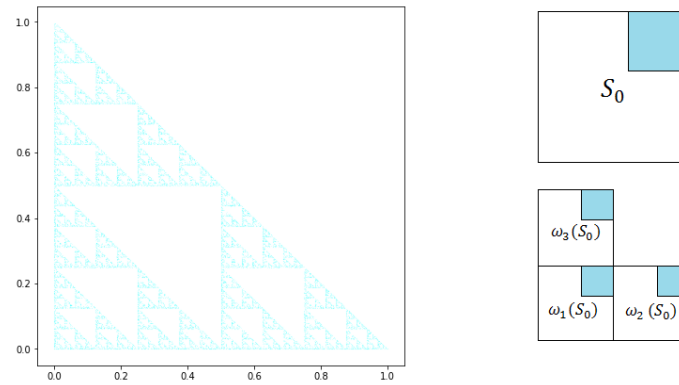


Figura 58: Fractal 1.

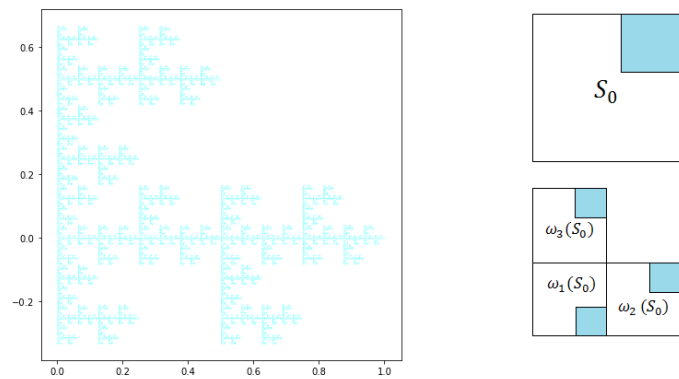


Figura 59: Fractal 2.

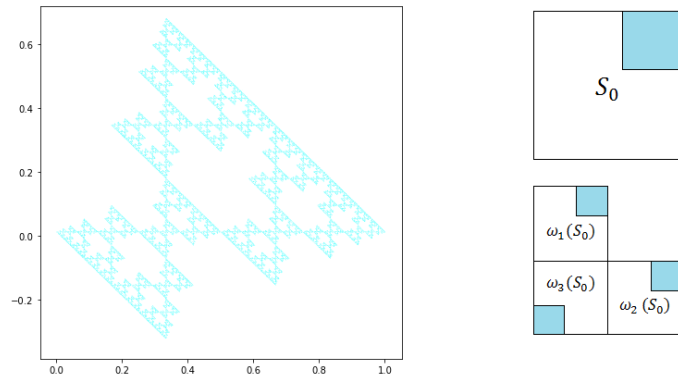


Figura 60: Fractal 3.

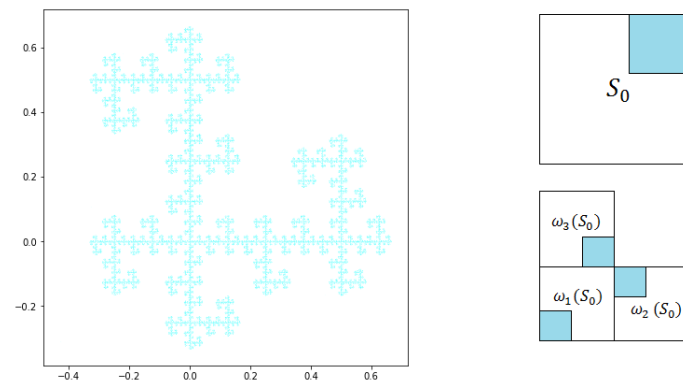


Figura 61: Fractal 4.

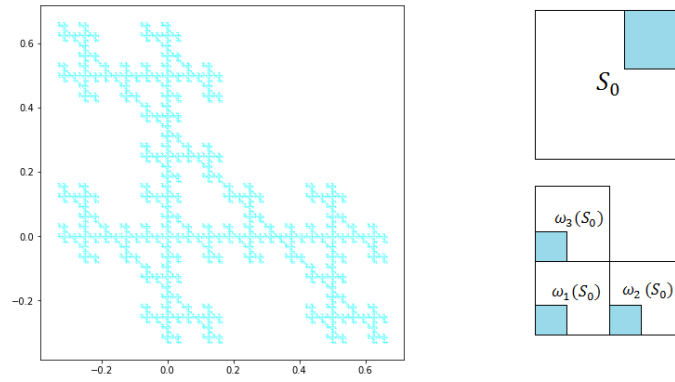


Figura 62: Fractal 5.

13. Apéndice D

13.1. CGR de Familias de Proteínas

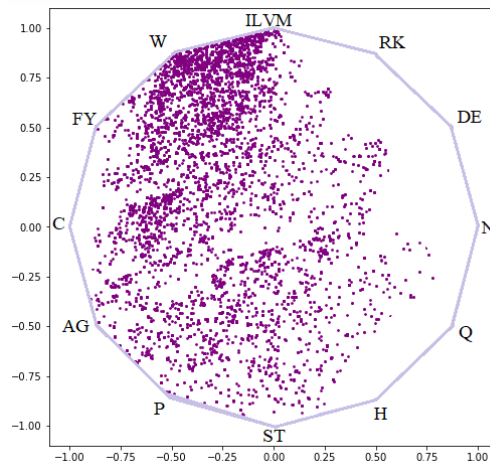


Figura 63: CGR de la Familia de Proteínas Hemoglobina.

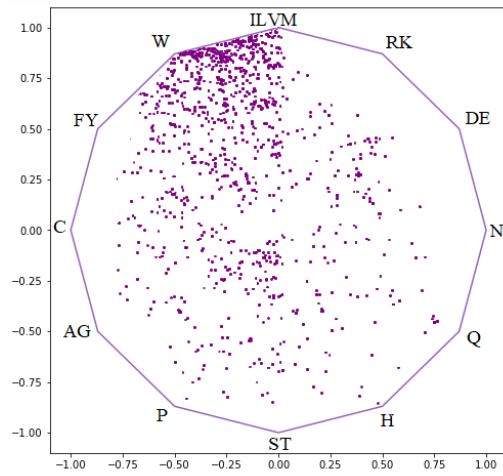


Figura 64: CGR de la Familia de Proteínas Rab.

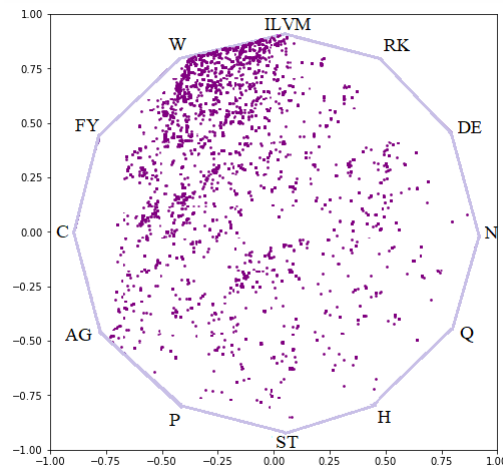


Figura 65: CGR de la Familia de Proteínas Lysozyme.

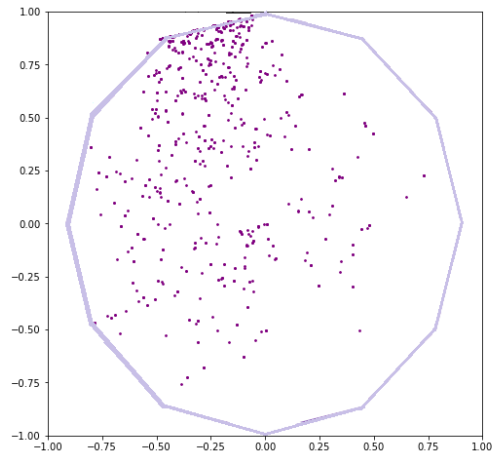


Figura 66: CGR de la Familia de Proteínas Familia Hsp90.

14. Referencias

- [1.] Jeffrey, H. J. (1990). Chaos Game Representation of Gene Structure. *Nucleic Acid Research*, 18(8), 2163-2170.
- [2.] Jeffrey, H. J. (1992). Chaos Game Visualization of Sequences. *Comput & Graphics*, 16(1), 25-33.
- [3.] Durán, G., López, J., and Del Río, J. L.(2019).The self-similarity properties and multifractal analysis of DNA sequences.*Applied Mathematics and Nonlinear Sciences*, 4(1), 267-278.
- [4.] T. Hoang, et al., Genomics (20169, <http://dx.doi.org/10.1016/j.ygeno.2016.08.002>
- [5.] Goldman, N. (1993). Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences. *Nucleic Acid Research*, 21(10), 2487-2491.
- [6.] Del Río, J. L, y Durán, G. (2017). Análisis Multifractal de secuencias del DNA. *Memorias XXII Reunión Nacional Académica de Física y Matemática*, 22(1), 261-267.
- [7.] Del Río, J. L. (2017). Análisis Multifractal de secuencias del DNA. *Bol. Soc. Mex. Fís*, 31(1), 17-31.

- [8.] Peitgen, H. O., Jürgens, H., and Saupe, D. (2002). Encoding Images by Simple Transformations, , *Chaos and Fractals New Frontiers of Science* (215-267). Springer.
- [9.] Barnsley, M. (1993). *Fractals Everywhere*. Morgan Kaufmann.
- [10.] Gnedenko, B. V. (1997). Markov Chains. Second Edition, *Theory of Probability* (110-119). Gordon and Breach Science Publishers.
- [11.] Sandefur, J. T. (1990). An introduction to Markov chains, Regular Markov chains, *Discrete Dynamical Systems: Theory and Applications* (86-110, 311-330). Oxford Univ. Pr. (Sd.)
- [12.] Fiser, A., Tusnády, G. E., and Simon, I. (1994). Chaos game representation of protein structures. *J. Mol. Graphics*, 12, 302-304.
- [13.] Basu, S., Pan, A., Dutta, C., and Das, J. (1997). Chaos game representation of proteins. *Journal of Molecular Graphics and Modelling*, 15, 279-289.
- [14.] Wang, H., and Wu, P. (2018). Prediction of RNA-protein interactions using conjoint triad feature and chaos game representation. *Bioengineered*, 9(1), 242-251
- [15.] Yang, J. Y., Peng, Z. L., Yu, Z. G., Zhang, R. J., Anh, V., and Wang, D. (2009). Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. *Journal of Theoretical Biology*, 257, 618-626.
- [16.] Deschavanne, P. J., Giron, A., Vilain, J., Fagot, G., and Fertil, B. (1999). Genomic Signature: Characterization and Classification of Species Assessed by Chaos Game Representation of Sequences. *Molecular Biology and Evolution*, 16(10), 1391-1399.
- [17.] National Human Genome Research Institute (NIH).
<https://www.genome.gov/>
- [18.] Snustad, D. P., and Simmons, M. J (2012). *Principles of Genetics*. John Wiley & Sons, Inc.
- [19.] Klug, W. S., Cumming, M. R., Spencer, C. A., and Palladino, M. A (2020). *Concepts of Genetics Twelfth Edition*. Pearson.
- [20.] arXiv:1502.01384 [math.DS]



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE EXAMEN DE GRADO

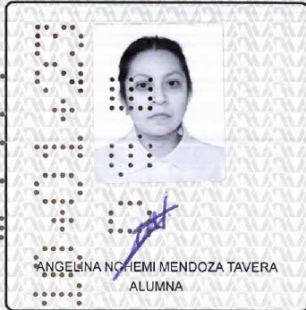
No. 00123

Matrícula: 2182800711

Aplicación del Juego del Caos para Secuencias de Aminoácidos.

En la Ciudad de México, se presentaron a las 12:00 horas del día 21 del mes de diciembre del año 2022 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. JOSE LUIS DEL RIO CORREA
DR. JOSE RAFAEL GODINEZ FERNANDEZ
DR. ALEJANDRO MUÑOZ DIOSDADO



ANGELINA NOHEMI MENDOZA TAVERA
ALUMNA

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron para proceder al Examen de Grado cuya denominación aparece al margen, para la obtención del grado de:


MAESTRA EN CIENCIAS (FISICA)

DE: ANGELINA NOHEMI MENDOZA TAVERA

y de acuerdo con el artículo 78 fracción III del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

Aprobar

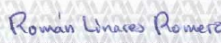
REVISÓ



MTRA. ROSALIA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

Acto continuo, el presidente del jurado comunicó a la interesada el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

DIRECTOR DE LA DIVISION DE CBI



DR. ROMAN LINARES ROMERO

PRESIDENTE



DR. JOSE LUIS DEL RIO CORREA

VOCAL



DR. JOSE RAFAEL GODINEZ FERNANDEZ

SECRETARIO



DR. ALEJANDRO MUÑOZ DIOSDADO