



INSTITUTO POLITÉCNICO NACIONAL

LA TÉCNICA AL SERVICIO DE LA PATRIA

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
UNIDAD ZACATENCO

MODELOS ASOCIATIVOS PARA LA IDENTIFICACIÓN DE ADN

## TRABAJO DE INVESTIGACIÓN

PARA OBTENER EL TÍTULO DE  
ING. EN COMUNICACIONES Y ELECTRÓNICA

PRESENTAN

STEPHANIE ANAID LOREDO SANTIAGO  
JOSÉ ANDRÉS OLMEDO WOODER

ASESORES:

DRA. MARÍA ELENA ACEVEDO MOSQUEDA  
DR. MARCO ANTONIO ACEVEDO MOSQUEDA



MÉXICO D. F.

JULIO 2015



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA**  
**UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”**

**TEMA DE TESIS**

QUE PARA OBTENER EL TÍTULO DE  
POR LA OPCIÓN DE TITULACIÓN  
DEBERA (N) DESARROLLAR

INGENIERO EN COMUNICACIONES Y ELECTRÓNICA  
TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL  
C. STEPHANIE ANAID LOREDO SANTIAGO  
C. JOSE ANDRES OLMEDO WOODER

“MODELOS ASOCIATIVOS PARA LA IDENTIFICACIÓN DEL ADN”

RECONOCER PATRONES DE ADN CON MEMORIAS ASOCIATIVAS Y SEÑALAR SI DICHA MUESRA ESTÁ ASOCIADA CON UN LINAJE EN ESPECÍFICO

- ❖ INVESTIGAR LOS TIPOS DE MEMORIAS ASOCIATIVAS PARA DETERMINAR CUAL ES LA MEJOR OPCIÓN PARA EL RECONOCIMIENTO DE PATRONES DE ADN
- ❖ IMPLEMENTAR EL TIPO DE MEMORIA ASOCIATIVA ELEGIDA EN UNA HERRAMIENTA DE SOFTWARE MATEMÁTICO QUE NOS PERMITA HACER LAS PRUEBAS EXPERIMENTALES QUE PERMITE LA CREACIÓN DE INTERFACES DE USUARIO
- ❖ REALIZAR LAS PRUEBAS DE RENDIMIENTO DE LA IMPLEMENTACIÓN DEL MÉTODO ELEGIDO
- ❖ COMPARAR LOS RESULTADOS DE LA IMPLEMENTACIÓN CON LOS RESULTADOS DE OTROS MÉTODOS

MÉXICO D.F, A 22 DE ENERO DE 2016.

ASESOR



M. EN C. MARÍA ELENA  
ACEVEDO MOSQUEDA



M. EN C. MARCO ANTONIO  
ACEVEDO MOSQUEDA



ING. PATRICIA LORENA RAMÍREZ RANGEL  
JEFA DEL DEPARTAMENTO DE INGENIERÍA  
EN COMUNICACIONES Y ELECTRÓNICA



DEPARTAMENTO





## ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS.....	V
ÍNDICE DE FIGURAS.....	VII
AGRADECIMIENTOS .....	IX
RESUMEN. ....	XI
<b>CAPÍTULO I. INTRODUCCIÓN.....</b>	<b>1</b>
1.1. PLANTEAMIENTO DEL PROBLEMA .....	1
1.2. JUSTIFICACIÓN.....	2
1.3. HIPÓTESIS .....	2
1.4. OBJETIVOS.....	3
a. <i>Objetivo general</i> .....	3
b. <i>Objetivos particulares</i> .....	3
1.5. ESTRUCTURA DEL TRABAJO.....	3
1.6. ALCANCE DE INVESTIGACIÓN .....	4
1.7. TRABAJO A FUTURO .....	4
<b>CAPÍTULO II. ESTADO DEL ARTE .....</b>	<b>5</b>
2.1. RECONOCIMIENTO DE PATRONES. ....	5
2.1.1. <i>Reconocimiento sintáctico de patrones</i> .....	5
2.1.1.1. Reconocimiento con cadenas y lenguajes formales .....	6
2.1.1.2. Árboles de decisión.....	7
2.1.1.3. Trabajos destacados en este enfoque .....	8
2.1.2. <i>Redes neuronales artificiales</i> .....	9
2.1.2.1. Aprendizaje por corrección de error.....	11
2.1.2.2. Aprendizaje competitivo.....	11
2.1.2.3. Aprendizaje asociativo .....	12
2.1.2.4. Aprendizaje estocástico o de Boltzmann .....	12
2.1.2.5. Trabajos destacados en este enfoque .....	12
2.1.3. <i>Reconocimiento lógico combinatorio de patrones</i> .....	14
2.1.3.1. Modelo ALVOT.....	15
2.1.3.2. Modelo KORA .....	16
2.1.3.3. Trabajos destacados en este enfoque .....	16
2.1.4. <i>Redes Neuronales aplicadas al Reconocimiento de Patrones</i> .....	17
2.2. RECONOCIMIENTO DE ADN.....	18
2.2.1. <i>Programa FÉNIX</i> .....	19
2.2.2. <i>Iniciativa DNA-PROKIDS</i> .....	20
2.2.3. <i>Identificación biológica de secuencias de ADN del genoma humano</i> .....	23
2.2.4. <i>BLAST</i> .....	24
<b>CAPÍTULO III. MARCO TEÓRICO.....</b>	<b>25</b>
3.1. MEMORIAS ASOCIATIVAS .....	25
3.1.1. <i>Lernmatrix de Steinbuch</i> .....	28
3.1.2. <i>Correlograph de Willshaw, Buneman y Longuet-Higgins</i> .....	30



3.1.3. <i>Linear Associator de Anderson-Kohonen</i> .....	31
3.1.4. <i>La memoria asociativa Hopfield</i> .....	32
3.1.5. <i>Memorias Asociativas Morfológicas</i> .....	34
3.1.5.1. <i>Memorias Heteroasociativas Morfológicas</i> .....	35
3.1.5.2. <i>Memorias Autoasociativas Morfológicas</i> .....	36
3.1.6. <i>Memorias Asociativas Alfa-Beta</i> .....	37
3.2.7. <i>Memorias Asociativas Media</i> .....	38
3.3. <i>ÁCIDO DESOXIRRIBONUCLEICO</i> .....	40
<b>CAPÍTULO IV. DISEÑO DEL SISTEMA</b> .....	<b>43</b>
4.1. <i>FASE DE APRENDIZAJE</i> .....	43
4.2. <i>RUIDO</i> .....	46
4.3. <i>FASE DE RECUPERACIÓN</i> .....	48
4.4. <i>INTERFAZ DE USUARIO</i> .....	51
<b>CAPÍTULO V. PRUEBAS Y RESULTADOS</b> .....	<b>57</b>
<b>CONCLUSIONES</b> .....	<b>81</b>
<b>GLOSARIO</b> .....	<b>LXXXIII</b>
<b>REFERENCIAS</b> .....	<b>LXXXV</b>



## ÍNDICE DE TABLAS

TABLA 3.1 DEFINICIÓN DE B	37
TABLA 3.2 DEFINICIÓN DE A	37
TABLA 4.1 CONVERSIÓN DE PATRÓN	44
TABLA 4.2 EJEMPLO DE MEMORIA MORFOLOGICA HETEROASOCIATIVA MAX (M)	46
TABLA 4.3 EJEMPLO DE MEMORIA MORFOLOGICA HETEROASOCIATIVA MIN (W)	46
TABLA 4.4 EJEMPLO DE PATRÓN CONVERTIDO A NÚMEROS ENTEROS	47
TABLA 4.5 PATRÓN CON UN 10% DE RUIDO ADITIVO	48
TABLA 4.6 PATRÓN CON UN 10% DE RUIDO SUSTRACTIVO	48
TABLA 4.7 EJEMPLO DE PATRÓN CONVERTIDO A NÚMEROS ENTEROS	49
TABLA 5.1 PRIMEROS 5 PATRONES DE LA BASE DE DATOS	57
TABLA 5.2 PATRÓN NÚMERO UNO CONVERTIDO A NÚMEROS ENTEROS	58
TABLA 5.3 PATRÓN NÚMERO DOS CONVERTIDO A NÚMEROS ENTEROS	58
SEGUNDO PATRÓN	58
TABLA 5.4 PATRÓN NÚMERO TRES CONVERTIDO A NÚMEROS ENTEROS	58
TERCER PATRÓN	58
TABLA 5.5 PATRÓN NÚMERO CUATRO CONVERTIDO A NÚMEROS ENTEROS	59
TABLA 5.6 PATRÓN NÚMERO CINCO CONVERTIDO A NÚMEROS ENTEROS	59
TABLA 5.7 MEMORIA MAX (M) DEL PRIMER PATRÓN	59
TABLA 5.8 MEMORIA MAX (M) DEL SEGUNDO PATRÓN	60
TABLA 5.9 MEMORIA MAX (M) DEL TERCER PATRÓN	60
TABLA 5.10 MEMORIA MAX (M) DEL CUARTO PATRÓN	60
TABLA 5.11 MEMORIA MAX (M) DEL QUINTO PATRÓN	60
TABLA 5.12 MEMORIA MIN (W) DEL PRIMER PATRÓN	61
TABLA 5.13 MEMORIA MIN (W) DEL SEGUNDO PATRÓN	61
TABLA 5.14 MEMORIA MIN (W) DEL TERCER PATRÓN	61
TABLA 5.15 MEMORIA MIN (W) DEL CUARTO PATRÓN	61
TABLA 5.16 MEMORIA MIN (W) DEL QUINTO PATRÓN	62
TABLA 5.17 TABLA DE TIEMPOS DE RECUPERACIÓN DE PATRONES SIN RUIDO	72



<b>TABLA 5.18 TIEMPO PROMEDIO DE RECUPERACIÓN DE TODOS LOS PATRONES, PARA CADA PORCENTAJE DE RUIDO ADITIVO</b> .....	<b>74</b>
<b>TABLA 5.19 TIEMPO PROMEDIO DE RECUPERACIÓN DE TODOS LOS PATRONES, PARA CADA PORCENTAJE DE RUIDO SUSTRATIVO</b> .....	<b>75</b>
<b>TABLA 5.20 PORCENTAJE DE RECUPERACIÓN DE PATRONES CON RUIDO ADITIVO.</b> .....	<b>77</b>
<b>TABLA 5.21 PORCENTAJE DE RECUPERACIÓN DE PATRONES CON RUIDO SUSTRATIVO.</b> .....	<b>78</b>



## ÍNDICE DE FIGURAS

FIGURA 3.1 SISTEMA DE UNA MEMORIA ASOCIATIVA	26
FIGURA 3.2 ESTRUCTURA DE UNA CÉLULA	42
FIGURA 4.1 UBICAR LA CARPETA DE ORIGEN	51
FIGURA 4.2 ABRIR LA INTERFAZ GRÁFICA DE USUARIO	51
FIGURA 4.3 SELECCIONAR LA UBICACIÓN	52
FIGURA 4.4 PLANTILLA DE INTERFAZ DE USUARIO	52
FIGURA 4.5 INTERFAZ DE USUARIO	53
FIGURA 4.6 SELECCIÓN DE PATRÓN EN LA INTERFAZ DE USUARIO	53
FIGURA 4.7 PROBANDO EL BOTÓN IDENTIFICAR	54
FIGURA 4.8 IDENTIFICACIÓN CON 0% DE RUIDO	54
FIGURA 4.9 IDENTIFICACIÓN CON 10% DE RUIDO ADITIVO	55
FIGURA 4.10 IDENTIFICACIÓN CON 10% DE RUIDO SUSTRACTIVO	55
FIGURA 5.1 INTERFAZ DE USUARIO PARA EL PATRÓN UNO CON 0% DE RUIDO	64
FIGURA 5.2 INTERFAZ DE USUARIO PARA EL PATRÓN DOS CON 25% DE RUIDO ADITIVO	66
FIGURA 5.3 INTERFAZ DE USUARIO PARA EL PATRÓN TRES CON 50% DE RUIDO ADITIVO	67
FIGURA 5.4 INTERFAZ DE USUARIO PARA EL PATRÓN CUATRO CON 25% DE RUIDO SUSTRACTIVO	69
FIGURA 5.5 INTERFAZ DE USUARIO PARA EL QUINTO PATRÓN CON 50% DE RUIDO SUSTRACTIVO	71
FIGURA 5.6 PORCENTAJE DE PATRONES RECUPERADOS CON RUIDO ADITIVO	78
FIGURA 5.7 PORCENTAJE DE PATRONES RECUPERADOS CON RUIDO SUSTRACTIVO	79





## **AGRADECIMIENTOS**

No es fácil llegar, se necesita lucha y deseo, pero sobre todo el apoyo como el que recibo día con día, toda mi vida.

A mis padres que siempre estuvieron ahí, y sin cuyo apoyo no hubiera podido concluir este trabajo de tesis.

A mi familia por sus consejos y apoyo moral.

A mis compañeros y amigos que de una u otra manera me soportaron y alentaron durante la carrera.

A mi director de tesis por compartirme sus conocimientos, su amistad y por su interés en la dirección y término de ésta.

A mis sinodales por la valiosa aportación a este trabajo, así como también a mis maestros de la carrera a los que debo mi formación académica y demás profesores amigos.

A todos, muchas gracias.





## RESUMEN.

En este trabajo se presenta un método de detección de patrones enfocado en ADN utilizando memorias asociativas morfológicas y un banco de datos de varias secuencias de ADN, en la que cada secuencia de ADN representa un patrón para la memoria y el cual pertenece a un linaje.

Se eligieron las memorias asociativas morfológicas por su mejora en la capacidad de almacenamiento con respecto a las memorias clásicas, así como el hecho de que la memoria morfológica no requiere convergencia, es decir, es capaz de responder en forma perfecta en un solo paso.

Para el caso de las memorias clásicas, es un hecho que aun sin distorsión en la entrada, no es aceptable esperar una respuesta perfecta. Para las memorias morfológicas, en cambio, es válida la existencia de ruidos sin perder la propiedad de respuesta perfecta.

El ADN que se utiliza para la base de datos es de tipo mitocondrial debido a que este tipo de ADN resiste las condiciones adversas sin ser degradado y se sigue conservando aún con el paso del tiempo.

Para lograr el desarrollo y funcionamiento del método de detección de ADN se implementa la fase de aprendizaje de las memorias asociativas morfológicas con los operadores Max y Min; para después implementar la fase de recuperación de las memorias asociativas morfológicas con los operadores Max y Min haciendo uso de una interfaz de usuario que facilite manipular en forma simple la detección de ADN, permitiendo visualizar los resultados obtenidos.

Dichos resultados se despliegan en una interfaz de usuario mostrando el linaje al que pertenece el individuo.





# CAPÍTULO I. Introducción

Los progresos científicos y tecnológicos en las herramientas utilizadas en genética han permitido nuevas aplicaciones en otras disciplinas. En este trabajo se pretenden abordar el reconocimiento de patrones de ADN el cual se puede aplicar en la identificación de cadáveres y restos humanos, reunión de víctimas de tráfico humano con sus familias o el reconocimiento de restos de personajes históricos. Esto tiene gran importancia en aplicaciones como Biomedicina, ya que cada día es más frecuente la colaboración entre grupos forenses y otras especialidades médicas.

El trabajo presentado a continuación fue implementado en el entorno de programación de Matlab 2014.

## 1.1. Planteamiento del problema

Actualmente el reconocimiento de ADN se hace sólo de muestras que no tienen ruido, o si contienen ruido, la muestra se somete a largos procesos biológicos que tardan tiempo y cuestan dinero, para tratar de reducir el ruido que se tiene en la muestra, sin embargo en algunos casos el ruido no se puede quitar completamente.



En la detección de ADN existen varios métodos que resuelven de manera eficaz dicha detección utilizando ADN nuclear, ya que este es único para cada persona, además sirve para determinar el sexo de la persona de la que proviene la muestra. Sin embargo en algunos casos este tipo de ADN resulta insuficiente como cuando se tiene escasa cantidad de muestra o la muestra se encuentra en mal estado.

## 1.2. Justificación

No en todos los casos el tipo de ADN nuclear funciona, mientras que el ADN mitocondrial funciona aun cuando existe una muestra en malas condiciones de conservación, con mucha antigüedad o se tenga una escasa cantidad.

Para la realización del método de detección de ADN se utilizaron las memorias asociativas morfológicas, debido a su mejora en la capacidad de almacenamiento con respecto a las memorias clásicas, así como el hecho de que la memoria morfológica no requiere convergencia; es decir, es capaz de responder en forma perfecta en un solo paso.

Para el caso de las memorias clásicas, es un hecho que aun sin distorsión en la entrada, no es aceptable esperar una respuesta perfecta. Para las memorias morfológicas, en cambio, es válida la existencia de ruidos sin perder la propiedad de respuesta perfecta.

## 1.3. Hipótesis

El problema planteado puede resolverse utilizando el tipo de ADN mitocondrial porque funciona aun cuando existe una muestra en malas condiciones de conservación, dado que el número de copias de ADN mitocondrial por célula es muy elevado y algunas pueden llegar a resistir bajo condiciones adversas, su conservación en el tiempo es mayor a la del ADN nuclear debido a su pequeño tamaño, es decir, que al ser más grande el ADN nuclear la probabilidad de verse afectado es mayor, y a pesar de tener una escasa cantidad de ADN mitocondrial es posible realizar el reconocimiento. Además se propone utilizar para la detección de ADN el tipo de memorias asociativas morfológicas, porque este tipo de memorias pueden trabajar con la existencia de ruido y además puede realizar tiene un alto porcentaje de recuperación sin ruido.



## 1.4. Objetivos

### a. Objetivo general

Reconocer patrones de ADN con memorias asociativas y señalar si dicha muestra está asociada con un linaje en específico.

### b. Objetivos particulares

- i. Investigar los tipos de memorias asociativas para determinar cual es la mejor opción para el reconocimiento de patrones de ADN.
- ii. Implementar el tipo de memoria asociativa elegida en una herramienta de software matemático que nos permita hacer las pruebas experimentales que permite la creación de interfaces de usuario.
- iii. Realizar las pruebas de rendimiento de la implementación del método elegido.
- iv. Comparar los resultados de la implementación con los resultados de otros métodos.

## 1.5. Estructura del trabajo

En este trabajo se presenta una metodología para la detección de ADN utilizando memorias asociativas morfológicas.

El trabajo está organizado de la siguiente manera:

El capítulo 1 describe el planteamiento del problema, hipótesis, los objetivos generales y particulares, así como la justificación del trabajo.

El capítulo 2 reseña otros métodos que se utilizan en la detección de ADN con anterioridad y sus resultados.

El capítulo 3 para un mejor entendimiento expone algunos conceptos básicos de lo que es el ADN, el tipo de ADN mitocondrial, tipos de memorias asociativas, memorias asociativas morfológicas.

El capítulo 4 explica el funcionamiento del sistema y las fases que se llevaron a cabo para su elaboración, además del modelo que se usó en la solución del problema.

En el capítulo 5 se muestran las pruebas y los resultados obtenidos.



## **1.6. Alcance de investigación**

Este trabajo se delimita a reconocimiento de patrones de ADN utilizando el tipo de ADN mitocondrial y las memorias asociativas morfológicas, por ende no se utiliza otro tipo de ADN, ni ningún otro tipo de memoria asociativa, esta detección de patrones sólo contempla los tipos de ruido aditivo o sustractivo.

## **1.7. Trabajo a futuro**

No limitar el reconocimiento de patrones de ADN al ADN mitocondrial y lograr la detección de los patrones aunque presenten otros tipos de ruido, como el ruido mixto o mezclado.



## CAPÍTULO II. Estado del Arte

En este capítulo se mencionan algunos trabajos e investigaciones relacionados con el reconocimiento de patrones realizados en los últimos años, los cuales nos permiten comparar y ver diferentes posibilidades de comprensión del tema.

### 2.1. Reconocimiento de patrones.

El reconocimiento de patrones tiene 3 enfoques: reconocimiento sintáctico de patrones, redes neuronales artificiales y reconocimiento lógico combinatorio de patrones [URL1].

#### 2.1.1. Reconocimiento sintáctico de patrones

Este enfoque se basa en encontrar las relaciones estructurales que guardan los objetos de estudio, utilizando la teoría de lenguajes formales. El objetivo es construir una gramática que describa la estructura del universo de objetos.

En el reconocimiento sintáctico, cada patrón se describe en términos de sus elementos básicos, llamados primitivas y representados por P, y unas reglas sintácticas, llamadas gramática y representadas por G; en conjunto, especifican cómo se pueden generar patrones válidos de una



determinada clase. El problema se reduce entonces a responder si un determinado patrón pertenece al lenguaje generado por una gramática.

### **2.1.1.1. Reconocimiento con cadenas y lenguajes formales**

#### **a. Lenguajes formales.**

Se basan en la definición de una gramática que describe las estructuras sintácticas válidas capaces de identificar. De esta forma, un objeto viene definido por una palabra del lenguaje. El reconocimiento de un objeto equivale a identificar si la palabra asociada al objeto pertenece a la gramática. La gramática del lenguaje depende del ámbito al que va dirigida la aplicación. El algoritmo general de reconocimiento con base en lenguajes formales consta de dos pasos principales:

1. La gramática generativa crea las cadenas de símbolos terminales usando las primitivas.
2. De modo analítico, dada una secuencia y una especificación de gramática, se determina si:
  - a. La secuencia fue generada por la gramática.
  - b. Si es así, determina la estructura de la secuencia.

#### **b. Reconocimiento con cadenas.**

Los patrones son representados por las secuencias ordenadas o cadenas de elementos discretos como una secuencia de letras en una palabra, la clasificación de patrones basada sobre dichas cadenas de símbolos discretos, difiere en el número de caminos de las técnicas más comúnmente usadas, debido a que los elementos de las cadenas (letras, caracteres o símbolos) son nominales; no hay una noción obvia de la distancia entre cadenas. Una dificultad se presenta del hecho que las cadenas necesitan ser de longitud diferente. Dichas cadenas no son vectores. Existe un gran número de problemas referentes a cadenas en el área de computación; los únicos que son de importancia en el reconocimiento de patrones son los siguientes:



- Emparejamiento de cadenas: dada una cadena  $x$ , determinar si  $x$  es una subcadena de la cadena  $X$ , si es así establecer su posición.
- Corrección de distancias: dadas dos cadenas  $x$  y  $y$ , calcular el número mínimo de operaciones básicas, como inserción, borrado e intercambio de caracteres, necesarias para transformar  $x$  en  $y$ .

### 2.1.1.2. Árboles de decisión

Es uno de los métodos de aprendizaje inductivo supervisado no paramétrico más utilizado. Su dominio de aplicación no está restringido a un ámbito concreto sino que pueden utilizarse en diversas áreas: diagnóstico médico, juegos, predicción meteorológica, control de calidad, etc. Puede verse como la estructura resultante de la partición recursiva del espacio de representación a partir de un numeroso conjunto de prototipos. Esta partición recursiva se traduce en una organización jerárquica del espacio de representación que puede modelarse mediante una estructura de tipo árbol. Cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada nodo hoja se refiere a una decisión (clasificación) [1], [2], [3].

Entre los clasificadores basados en árboles descritos en la literatura se encuentran ID3, C4, C4.5, árboles Bayesianos y CART (*Classification And Regression Trees*). Las diferencias principales entre los distintos algoritmos de construcción de árboles de decisión radican en las estrategias de poda y en la regla adoptada para dividir nodos.

### Introducción a los algoritmos de biología computacional

Se utiliza el algoritmo de reconocimiento de patrones de árboles de sufijos para la búsqueda en base de datos de secuencias de ADN.

El algoritmo puede encontrar todas las apariciones de un patrón  $P$  (de longitud  $m$ ) en una cadena madre  $S$  (de longitud  $n$ ) en un tiempo  $O(n)$  en el caso peor. Examina cada carácter de  $S$  al menos una vez, luego realiza un mínimo de  $n$  comparaciones. BM es sublineal, no examina necesariamente todos los caracteres de  $S$  y el número de comparaciones es, a menudo, inferior



a  $n$ . Además, tiende a ser más eficiente cuando  $m$  crece. En el mejor caso, encuentra todas las apariciones de  $P$  en  $S$  en un tiempo  $O(m+n/m)$ .

Se utilizan dos reglas para decidir el desplazamiento que hay que hacer después de una discrepancia:

Regla A. Después de un desplazamiento, se compara  $p_m$  con un carácter  $c$  de  $S$  y son distintos:

Si  $c$  aparece más a la izquierda en el patrón, se desplaza éste para alinear la última aparición de  $c$  en el patrón con el carácter  $c$  de  $S$ .

Si  $c$  no está en  $P$ , se coloca éste inmediatamente detrás de la aparición de  $c$  en  $S$ .

Regla B. Si un cierto  $n^\circ$  de caracteres al final de  $P$  se corresponde con caracteres de  $S$ , se aprovecha este conocimiento parcial de  $S$  para desplazar  $P$  a una nueva posición compatible con la información que se posee.

Se recuperan todos los patrones, pero no acepta ningún tipo de ruido, y lo hace en un tiempo promedio, con la misma base de datos que utilizamos, de 10 minutos. [4]

### 2.1.1.3. Trabajos destacados en este enfoque

En [5], se presenta un método para la localización automática de acontecimientos fonéticos importantes en señales de discurso continuas. Estos acontecimientos se han elegido para emparejar los requisitos de un sistema que realiza estas regiones para hacer la señal más robusta para acanalar la degradación. La técnica utilizada es un acercamiento sintáctico para reconocimiento de patrones distinguido del reconocimiento automático de discurso o de acercamientos basados en el conocimiento. El funcionamiento en discurso leído de muchos transmisores es aproximadamente del 20% de errores y 20% de falsas alarmas.

En [6], se analizan fundamentos teóricos del reconocimiento sintáctico de patrones y de sus relaciones con matemática lingüística, el reconocimiento de patrones estructural, y el reconocimiento de patrones estadístico. Finalmente, observan que los tres acercamientos



principales al reconocimiento de patrones (estructural, estadístico y sintáctico) se pueden combinar en un método de gran alcance. Y se propone un sistema basado en estos principios y previsto para ser utilizado para el reconocimiento de escritura a mano.

Martí en [7] realiza un reconocimiento estructural mediante la estructura de grafos en documentos de tipo técnico (arquitectura e ingeniería) que contienen bloques gráficos, porque en estos es importante realizar un análisis de las líneas que usualmente interconectan los símbolos para obtener una mejor descripción y significado del diseño, con el objetivo de extraer información contextual que en algunos casos se utiliza para verificar si el documento gráfico cumple las reglas de diseño propias del ámbito técnico al que hace referencia.

En [8] se usa el reconocimiento sintáctico de patrones para la interpretación automática de formas reflejadas, las pruebas preliminares se realizan sobre formas geométricas simples. Los lenguajes usados para describir patrones ruidosos y distorsionados son a menudo ambiguos, de tal manera que, una secuencia o patrón se puede generar por más de un lenguaje, así que los patrones que pertenecen a diversas clases pueden tener la misma descripción, pero con diversas probabilidades de ocurrencia.

### **2.1.2. Redes neuronales artificiales**

Este enfoque supone que tiene una estructura de neuronas interconectadas que se estimulan unas a otras, las cuales pueden ser entrenadas para dar una cierta respuesta cuando se le presentan determinados valores. Adquieren, almacenan y utilizan conocimiento experimental, obtenido a partir de ejemplos por ajuste de parámetros de las neuronas mediante un algoritmo de aprendizaje. Dependiendo de la manera por la cual una red neuronal se relaciona con el ambiente, existen los siguientes paradigmas de aprendizaje [9], [10], [URL2]:

- **Aprendizaje Supervisado:** se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la



salida obtenida se aproxime a la deseada. Este tipo de aprendizaje es utilizado en las redes neuronales *BackPropagation*, Perceptron y ADALINE.

- Aprendizaje por Refuerzo: es una variante de aprendizaje supervisado, que se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. En el aprendizaje por refuerzo la función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito = +1 o fracaso = -1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades. Se podría decir que en este tipo de aprendizaje la función del supervisor se asemeja más a la de un crítico, que opina sobre la respuesta de la red; que a la de un maestro, que indica a la red la respuesta concreta que debe generar.
- Aprendizaje No Supervisado (auto-organización): no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en su entrada. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado en el pasado. En otro caso, podría realizar un agrupamiento o establecimiento de categorías (*clustering*), indicando la red a la salida a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas. Este tipo de aprendizaje es utilizado en los modelos de Mapas de Kohonen [11], [12] y redes de tipo ART (*Adaptive Resonance Theory*).



Existen muchos algoritmos de aprendizaje, entre los principales se tienen: aprendizaje por corrección de error, aprendizaje competitivo, aprendizaje asociativo y aprendizaje estocástico o de Boltzmann [13].

#### **2.1.2.1. Aprendizaje por corrección de error**

Algoritmo muy conocido basado en la regla Delta o regla del mínimo error cuadrado LMS, que busca minimizar la función de error usando un gradiente descendente. Este es el principio *BackPropagation*, muy utilizado para el entrenamiento de redes de múltiples capas. Este método de entrenamiento es de gran alcance, útil, y relativamente fácil de entender [2].

A este tipo también pertenece la regla de aprendizaje del Perceptron, utilizada en el entrenamiento de la red del mismo nombre; consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos a la salida de la red, es decir, en función del error cometido en la salida [URL2].

#### **2.1.2.2. Aprendizaje competitivo**

En las redes con aprendizaje competitivo, suele decirse que las neuronas compiten y cooperan unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, sólo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se active, es decir, alcance su valor de respuesta máximo. Por tanto las neuronas compiten para activarse quedando finalmente una, o una por grupo, como neurona vencedora; el resto quedan anuladas y siendo forzadas a sus valores de respuesta mínimos.

El objetivo de este aprendizaje es categorizar los datos que se introducen en la red, de esta forma las informaciones similares son clasificadas formando parte de la misma categoría y por tanto deben activar la misma neurona de salida. Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado a través de las correlaciones entre los datos de entrada. [1], [14]. Este tipo de aprendizaje es usado en redes ART, redes *Hamming* y en mapas de *Kohonen* con sus dos variantes denominadas LVQ (*Learning Vector Quantization*) y TPM (*Topology Preserving Map*) o SOM (*Self Organizing Map*) [15].



### 2.1.2.3. Aprendizaje asociativo

Esta regla de aprendizaje es la base de muchas otras, la cual pretende medir la familiaridad o extraer características de los datos de entrada.

El fundamento es una suposición bastante simple conocido como principio de *Hebb*: si dos neuronas  $N_i$  y  $N_j$  toman el mismo estado simultáneamente, ambas activas o ambas inactivas, el peso de la conexión entre ambas se incrementa, en el caso contrario será debilitada. Las redes asociativas se utilizan principalmente para filtrado de información en la reconstrucción de datos, eliminando distorsiones o ruido, también se emplean para explorar relaciones entre informaciones similares, para facilitar la búsqueda por contenido en bases de datos y para resolver problemas de optimización [16]. Se utiliza en el Modelo de *Hopfield* [17], [18].

### 2.1.2.4. Aprendizaje estocástico o de Boltzmann

Consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

El objetivo del aprendizaje de Boltzmann es ajustar los pesos de conexión de tal forma que el estado de las unidades visibles satisfaga una distribución de probabilidades deseada en particular. Según lo anterior, el aprendizaje consistiría en realizar un cambio aleatorio de los valores de los pesos y determinar la energía de la red. Si la energía es menor después del cambio, es decir, si el comportamiento de la red se acerca al deseado, se acepta el cambio; si, por el contrario, la energía es mayor, se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades [1], [URL2].

### 2.1.2.5. Trabajos destacados en este enfoque

El algoritmo descrito en [19] aprovecha las características de color y forma de las señales de tráfico para llevar a cabo su reconocimiento. Éste lo realiza en 2 fases: primera, para la detección, utiliza una técnica de umbral, *thresholding*, aplicada al color del objeto para dividir el análisis de la imagen en segmentos y de la forma para detectar las muestras. Segundo, para la clasificación, utiliza una red neuronal Perceptron multicapa pudiendo manipular 3 diferentes



tamaños de imágenes. El algoritmo fue implementado en una PC 486 a 33 MHz con bus local. La velocidad de detección de fase es 220 ms para una imagen de 256 X 256 bits. La red neuronal en la CPU toma 1.2 s. La puesta en práctica de la red neuronal en un procesador digital de señales (DSP) se está experimentando en la investigación, y la velocidad prevista está entre 30 y 40 ms.

En [20], para el reconocimiento de patrones en imágenes digitales realiza en un programa computacional la implementación y comparación de dos redes neuronales, una red ART2 y una *Backpropagation*. Las imágenes utilizadas para probar las redes son figuras geométricas, letras y rostros; y para poder extraer las características de éstas y posteriormente reconocerlas, se utilizaron los momentos invariantes y la transformada rápida de Fourier discreta.

En general, se obtienen mejores resultados con la red ART2 usando menos tiempo para las pruebas, ya que no depende del azar. Pero la información corre el riesgo de degradarse si no se tiene cuidado al introducir nuevas imágenes que estén en valores más bajos que la mínima similitud requerida entre imágenes de la misma clase; ya que la red clasificará mal las imágenes. En oposición, la red *Backpropagation* requiere más experimentos durante su fase de aprendizaje para encontrar la solución a un problema dado, pero no corre el riesgo de perder ninguna porción de su información durante su funcionamiento.

En [21], el algoritmo propuesto para la detección de señales de tráfico parte de un análisis del color, debido a los problemas de iluminación el espacio de colores utilizado es el HSI. Se utilizan algoritmos genéticos para realizar la búsqueda de los valores de la transformación que mejor se ajustan a la señal presente en la imagen. La función que mide lo bien que el modelo concreto se ajusta a la imagen se basa en la distancia de *Hausdorff*, que mide la separación entre dos conjuntos de puntos. Para el reconocimiento utilizan una red neuronal de tipo ART1 ya que es capaz de almacenar el conocimiento y no necesita repetir el proceso de entrenamiento si se presentan nuevos tipos, por lo que el funcionamiento de la red le sirve a su vez de entrenamiento.

En [22], utilizan un algoritmo genético para la tarea de la detección, permitiendo una localización invariante a los cambios en la posición, la escala, la rotación, condiciones atmosféricas, la obstrucción parcial, y la presencia de otros objetos del mismo color. Para el reconocimiento de las señales se usan las redes neuronales ART1 para entradas binarias. Estas



redes son capaces de desarrollar grupos estables de secuencias arbitrarias de los patrones de entrada. Tienen dos capas con trayectorias abajo-arriba y arriba-abajo entre ellas. La capa de salida está diseñada como una red competitiva, sólo el nodo que recibe el total más grande de entrada se activa. Este nodo genera un patrón arriba-abajo que se compara con la entrada. Si la diferencia es bastante grande el nodo se inhibe y el ciclo empieza de nuevo. Cuando la similitud es alta hay un estado resonante de la red: una categoría de vector prototipo iguala el actual vector de entrada bastante cerca. Si no, una señal es generada significando que un patrón diferente se presenta y se crean nuevas neuronas. Por lo tanto, no es necesario volver a entrenar la red si nuevos patrones tienen que ser agregados. Además, la red refina y enriquece su conocimiento con el uso.

### **2.1.3. Reconocimiento lógico combinatorio de patrones**

Este enfoque se basa en la idea de que la modelación del problema debe ser lo más cercana posible a la realidad del mismo, sin hacer suposiciones que no estén fundamentadas. Uno de los aspectos esenciales de este enfoque es que las características utilizadas para describir a los objetos de estudio deben ser tratadas cuidadosamente.

El objetivo en este enfoque es modelar problemas donde los patrones puedan estar formados por cualquier combinación de características tanto cualitativas como cuantitativas. Los problemas de clasificación se modelan según la Teoría clásica de conjuntos o la Teoría de subconjuntos difusos. Típicamente se aplica la Teoría Combinatoria para considerar las posibilidades de representación y análisis de los patrones. Es común en este enfoque estudiar y modificar algoritmos generados originalmente en otros enfoques para remover algunas de sus limitaciones y extender su capacidad de aplicación [23].

La clasificación con el enfoque lógico combinatorio se aplica a problemas donde el universo de objetos se agrupa en un número dado de clases, de las cuales se tiene información de cada una de ellas. El problema consiste en poder establecer las relaciones de un nuevo objeto dado con cada una de dichas clases, basada en semejanzas parciales entre los objetos.



El paradigma de aprendizaje utilizado en este enfoque es el aprendizaje supervisado. Los principales algoritmos de clasificación son: el modelo ALVOT o algoritmo de votación y el modelo KORA, basado en analogías parciales.

### **2.1.3.1. Modelo ALVOT**

El modelo de los algoritmos de votación se describe mediante 6 etapas [24]:

- Sistema de conjunto de apoyo: por un conjunto de apoyo se entiende un subconjunto no vacío de rasgos en términos de los cuales se analizarán los objetos. Cada conjunto da idea de puntos de vistas o representación de un subespacio. El sistema es el conjunto de los conjuntos de apoyo.
- Función de semejanza: establece los criterios de comparación de los valores para cada rasgo y para cada conjunto de apoyo.
- Evaluación por fila dado un conjunto de apoyo fijo: proceso de conteo de votación en cuanto a la medida de semejanza entre las diferentes partes de la subdescripción de los objetos ya clasificados y el que se desea clasificar.
- Evaluación por clase dado un conjunto fijo: aquí se totalizan las evaluaciones obtenidas para cada uno de los objetos de la matriz de aprendizaje respecto al objeto que se quiere clasificar.
- Evaluación por clase para todo el sistema de conjuntos de apoyo: se totalizan los mismos cálculos hechos para un conjunto de apoyo fijo ahora para todo el sistema seleccionado.
- Regla de solución: se trata ahora de establecer un criterio, para que sobre la base de cada una de las votaciones obtenidas en la etapa precedente dar una respuesta en cuanto a las relaciones del objeto a clasificar con cada una de las clases del problema, tomando en consideración la evaluación por clase para todo el sistema de conjuntos de apoyo.



### 2.1.3.2. Modelo KORA

Este algoritmo es también un algoritmo de precedencia parcial, es decir, analiza la experiencia acumulada. Existen dos variantes del modelo, el KORA 3 que es el modelo inicial y la extensión de la variable inicial o KORA  $\Omega$ .

El algoritmo KORA 3 consta de 3 etapas [24]:

- Etapa de aprendizaje: define los parámetros iniciales de suficiencia a aplicar en el modelo y depende en gran medida de los especialistas y del problema.
- Etapa de re-aprendizaje: etapa de cálculo de los parámetros y rasgos óptimos del modelo.
- Etapa de clasificación: dado un nuevo objeto realiza la clasificación mediante la aplicación de una regla de solución.

El modelo KORA 3 tiene algunas restricciones para su correcta aplicación, entre ellas se pueden mencionar las siguientes: sólo admite dos clases, la descripción de los objetos es en rasgos booleanos y no admite ausencia de información en la descripción de los objetos.

KORA  $\Omega$  consiste en permitir las siguientes características: uso de cualquier sistema de conjunto de apoyo, rasgos de cualquier naturaleza y cualquier cantidad de clases.

### 2.1.3.3. Trabajos destacados en este enfoque

En [24] se presenta un sistema de clasificación supervisada, basado en 2 diferentes algoritmos para reconocimiento de patrones: el modelo ALVOT y el modelo KORA extendida haciéndose un análisis crítico y experimental de diferentes criterios de comparación. Este estudio permite analizar las ventajas y limitaciones de un algoritmo respecto al otro, así como analizar la propuesta de clasificación que brindan los mismos.

En [25] se exponen las posibilidades de las herramientas del reconocimiento lógico combinatorio de patrones para la minería de datos de grandes grupos de datos mezclados incompletos. A partir de la existencia verdadera de muchas estructuras complejas de grandes grupos de datos, trabajan en la solución del problema de clasificación supervisado y no supervisado pero con estos tipos de grupos de datos.



En [26], se proponen con una matriz de representación del conocimiento, efectivos algoritmos lógico-combinatorio-probabilístico de reconocimiento de patrones en un sistema inteligente. Los 3 algoritmos se construyen en base a pruebas de diagnóstico mínimas, incondicionales, o mezclas de pruebas de diagnóstico. El reconocimiento de patrones finalmente se realiza según el principio de la votación sobre el grupo de todas las pruebas y algoritmos. Los algoritmos son recomendados para una pequeña cantidad de realizaciones de cada patrón y de un grupo grande de características. Los algoritmos fueron puestos en ejecución y probados parcialmente en el sistema IMSLOG en la solución de problemas reales.

El sistema es lo suficientemente flexible para que se le puedan incorporar nuevos algoritmos de clasificación y modelar diferentes aplicaciones que requieran de la clasificación, permitiendo definir sus rasgos, clases y especificaciones propias.

En [23] se aborda el problema de diseñar una metodología unificada para evaluar algoritmos de clasificación, tanto para problemas Supervisados como para problemas No-Supervisados. La metodología propuesta se basa en el enfoque de reconocimiento lógico combinatorio de patrones unificando los criterios de evaluación para ambos tipos de problemas y logra superar las debilidades de los métodos actuales. Para colaborar en la solución de este problema se diseña un lenguaje formal para especificación de problemas en Reconocimiento de patrones (LCARS) y se propone una arquitectura general para herramientas de software que hagan uso del lenguaje propuesto. La arquitectura propuesta se valida mediante la construcción de un prototipo de software (SHELL y STUDIO).

#### **2.1.4. Redes Neuronales aplicadas al Reconocimiento de Patrones**

Esta línea de investigación se centra en la aplicación de redes neuronales al reconocimiento de patrones. En esta dirección, se han resuelto problemas de *clustering* y clasificación. Actualmente, se estudia la posibilidad de optimizar las soluciones propuestas incorporando estrategias evolutivas sobre la arquitectura y el aprendizaje de la red.

Algunos resultados obtenidos en los trabajos experimentales:



**A) Reconocimiento de patrones.** Se ha desarrollado una nueva red neuronal para *clustering* y segmentación de los datos o patrones de entrada que, a diferencia de los modelos existentes, no requiere la indicación de parámetros de aceptación dependientes del problema [27]. Su comparación con las técnicas clásicas de reconocimiento de patrones ha dado resultados favorables.

Esta red fue pensada para la clasificación de píxeles de imágenes a 256 colores y forma parte de un proceso para el reconocimiento de los elementos de una muestra de tejido. Reducir la cantidad de parámetros de entrada implica un costo adicional que está dado por la evaluación del entorno de cada patrón. Esta evaluación puede tener un costo computacional alto por lo que resulta de interés analizar la paralelización de la red propuesta.

**B) Paralelización de la arquitectura CBE.** Resultados experimentales sobre distintos tipos de imágenes han demostrado la efectividad de la arquitectura CBE para resolver el *clustering* de un conjunto de patrones de entrada. En la solución monoprocesador, el tiempo de procesamiento es del orden  $n \times n \times p$  donde  $n \times n$  es la resolución de la imagen y  $p$  es la cantidad de patrones diferentes utilizados.

Luego de analizar la solución no paralela, se ha concluido en que el tiempo más significativo es el empleado en realizar la transformación del espacio de píxeles al espacio de patrones. Esta tarea es del orden  $n^2 \times p$ . El segundo factor es el análisis de similitud entre patrones de orden  $p^2$ .

Utilizando estos resultados se trabajó sobre una arquitectura paralela basada en procesadores homogéneos y memoria distribuida que permitió mejorar el *speed-up* de la solución [28].

## 2.2. Reconocimiento de ADN

Los avances en identificación genética permiten la aplicación de técnicas de reconocimiento de patrones a áreas muy diferentes. Los progresos científicos y tecnológicos en las herramientas utilizadas en genética forense han permitido nuevas aplicaciones de este campo en otras disciplinas. Algunas de las alternativas son el programa FENIX y DNA-PROKIDS



### 2.2.1. Programa FÉNIX

Se han hecho grandes esfuerzos en la identificación de cadáveres y restos humanos tras guerras, problemas socio-políticos y desastres en masa. En muchos de estos casos, el uso de la identificación a través de ADN ha ofrecido una respuesta definitiva en la identificación de las víctimas. La creación de una base de datos civil de personas desaparecidas no compromete los derechos de los individuos ya que la donación de las muestras incluidas en ella es voluntaria. España fue el primer país en establecer una base de datos forenses civil. Debido a la dinámica de las sociedades modernas, constantemente se presentan denuncias de personas desaparecidas así como se encuentran cadáveres y restos humanos. Se trata de un problema universal, una situación común que no solo está relacionada con guerras o catástrofes naturales.

En 1998, el grupo de investigación del Departamento de Medicina Legal y Toxicología de la Universidad de Granada ideó un programa de identificación genética de personas desaparecidas (Programa FÉNIX) para la identificación de cadáveres o restos cadavéricos que no se pudiesen identificar empleando métodos tradicionales. Su finalidad era prestar ayuda a los familiares de las personas desaparecidas. Éste programa fue pionero en el mundo entero ya que anteriormente no existía ni existe ninguna base de datos con estos fines. Ésta iniciativa fue apoyada por la Guardia Civil y el Ministerio del Interior, lo que permitió la creación de un programa nacional de identificación de cadáveres y restos de personas desaparecidas.

El programa FÉNIX contiene dos bases de datos independientes donde se pueden comparar automáticamente secuencias de ADN para identificar perfiles relacionados entre sí. Una se conoce como Base de Datos de Referencia que contiene información de micro satélites de ADN 4nuclear y cromosoma “Y” así como de secuencias de ADN mitocondrial de parientes que provienen de la línea materna. Debido a que esta base de datos es una base no criminal, sólo se encuentran en ella aquellos individuos que donan su ADN de forma voluntaria y siempre tras firmar un documento de consentimiento informado. La otra base de datos se denomina Base de Datos Dubitados y contiene todos los perfiles de restos encontrados que no han podido ser identificados por métodos clásicos como son, identificación de huellas dactilares, parámetros antropológicos, odontología, rayos “X”, etc [31].



Actualmente, 800 familias de personas desaparecidas participan en el programa FÉNIX y se han logrado 350 identificaciones de restos encontrados.

Estrategias similares se están implantando cada vez más en todo el mundo. En Brasil, donde en 2007 se determinó que existían 7,287 cadáveres y restos humanos sin identificar, se ha desarrollado una herramienta bioinformática para registrar y comparar la información genética y no genética de las personas desaparecidas, sus familiares y cadáveres no identificados, *Missing person Database-gen4*. Tras la guerra de la antigua Yugoslavia, el ICMP (*International Commission of Missing Persons*) desarrolló un programa de recogida de muestras de familiares de personas desaparecidas para la identificación de todos aquellos cadáveres y restos desconocidos.

Desde los primeros pasos del programa FÉNIX diversos países del área iberoamericana se mostraron interesados en su desarrollo. Algunos, incluso, pretenden poner en marcha un programa similar, dándose el caso de que, como en Colombia, se pretende que se denomine "FÉNIX-Colombia". En este país, responsables del Laboratorio de Genética del CTI (*Centro Técnico de Investigación*), adscrito a la Fiscalía General de la Nación, pretenden instaurar un sistema de identificación genética de personas desaparecidas para atacar el grave problema que en este sentido genera el elevado índice de violencia producido tanto por la actuación de la guerrilla como por otras bandas de delincuencia organizada fundamentalmente relacionadas con el narcotráfico.

En Chile, el Servicio Médico Legal, por orden del Parlamento de este país, pondrá en marcha una base de datos de muestras de detenidos-desaparecidos, habiéndose solicitado la colaboración del Programa FÉNIX.

### **2.2.2. Iniciativa DNA-PROKIDS**

Esta iniciativa, quiere ser una herramienta para luchar contra el tráfico de seres humanos, en especial de los más débiles, los niños. Según la UNODC, por 'trata de personas' se entenderá la captación, el transporte, el traslado, la acogida o la recepción de personas, recurriendo a la amenaza o al uso de la fuerza u otras formas de coacción, al rapto, al fraude, al engaño, al abuso de poder o de una situación de vulnerabilidad o a la concesión o recepción de pagos o beneficios



para obtener el consentimiento de una persona que tenga autoridad sobre otra, con fines de explotación.

Los niños, los individuos más inocentes y frágiles de la sociedad, pueden ser sujeto de muchos tipos de abusos. Uno de éstos es el tráfico humano que se ha convertido en una actividad criminal lucrativa. Según UNICEF, 300 millones de niños en todo el mundo sufren violencia, explotación y abusos incluyendo las peores formas de explotación laboral infantil, conflictos armados y prácticas dañinas. En los Estados Unidos se denuncian 800,000 desapariciones de niños al año. De éstos, aproximadamente 360,000 se escapan, 340,000 se clasifican como "desapariciones sin explicación plausible" y unos 100,000 son secuestrados por miembros de su familia u otros individuos conocidos o han sufrido algún daño [29].

Los gobiernos son conscientes de que el problema del tráfico humano es real y que se han de tomar medidas inmediatas. Sin embargo, es conveniente llegar a acuerdos internacionales para identificar las posibles víctimas y los responsables de estos delitos. Pero las dificultades legales y sociales que existen ralentizan el proceso de implementar un sistema contra el tráfico humano. Esto puede ser muy frustrante para los científicos que quieran ayudar desde una perspectiva humanitaria y disponen de las aplicaciones técnicas necesarias para asistir en la identificación de personas desaparecidas. Aunque las infraestructuras legales se desarrollen de forma lenta, no hay razón alguna para no implementar las herramientas analíticas y las bases de datos para la identificación de los niños desaparecidos [30].

DNA-PROKIDS es un proyecto internacional centrado en la lucha contra el tráfico de seres humanos a través de la identificación genética de las víctimas y sus familiares, especialmente en niños. Su meta no se centra únicamente en la resolución de crímenes domésticos sino que se pretende extender los resultados a nivel mundial con el objetivo de estimular la lucha contra el tráfico de seres humanos y ayudar a prevenir la misma [URL3].

El proyecto DNA-PROKIDS se inició en 2004 a través del Laboratorio de Identificación Genética de la Universidad de Granada. Tras un estudio piloto con países de Centroamérica y Asia llevado a cabo entre 2006 y 2008, se decidió la ampliación definitiva con alcance mundial. En 2009, con la colaboración de los Profs. Bruce Budowle y Arthur Eisenberg del Centro de Identificación Humana de la Universidad del Norte de Texas (UNT-HIC) y con la financiación



de BBVA, Caja Granada, Fundación Marcelino Botín y *Life Technologies* (EEUU) comenzó la expansión internacional del proyecto. Actualmente se ha ampliado la colaboración de la expansión mundial a China, India, Indonesia, Filipinas, Nepal, Sri Lanka, Tailandia, Brasil, Guatemala y México.

La principal misión de DNA-PROKIDS es identificar a las víctimas y devolverlas a su familia; dificultar el tráfico de seres humanos, incluyendo las adopciones ilegales, gracias a la identificación de las víctimas; y obtener información sobre los orígenes, las rutas y los medios de comisión del delito, claves para el trabajo de las fuerzas policiales y judiciales. Por otro lado, los objetivos de la iniciativa DNA-PROKIDS son:

- Promover una colaboración internacional sistemática y automática a través de la integración en una misma base de datos mundial de datos genéticos para lograr:
  - Reintegrar a estos niños desaparecidos en sus familias gracias a la comparación de su ADN con el de los familiares de personas desaparecidas susceptibles de haber sido víctimas de tráfico de seres humanos.
  - Luchar contra las adopciones ilegales al comprobar si las personas que entregan a un menor en adopción son los verdaderos familiares biológicos (madre, padre, abuelos), para evitar que menores procedentes de raptos, robos o tráfico de personas puedan ser dados en adopciones presuntamente legales.
- Estudiar y aplicar mejoras a los sistemas policiales y judiciales de todo el mundo para que luchen más eficazmente contra el tráfico de seres humanos, especialmente de niños y mujeres.
- Analizar y proponer una base legislativa común para la resolución de este problema.
- Evaluar e intentar solventar los problemas sociales y de comunicación que impiden una lucha más eficaz contra el tráfico de seres humanos.
- Ofrecer colaboración para la coordinación y la formación de especialistas en identificación de los diferentes países.



DNA-PROKIDS emplea métodos acreditados y aceptados por la comunidad científica internacional para la identificación genética de las personas desaparecidas. Los marcadores genéticos de rutina que se estudian son aquellos STR incluidos en sistemas como CODIS, ADN mitocondrial para confirmar la línea materna y STR del cromosoma Y para establecer relaciones de tipo paterno-filial. Actualmente, se están desarrollando paneles de SNPs (*Single Nucleotide Polymorphism*) que nos van a permitir determinar características fenotípicas de los individuos así como su procedencia geográfica y étnica.

### **2.2.3. Identificación biológica de secuencias de ADN del genoma humano**

El principal aporte del trabajo, consistió en la identificación biológica de las repeticiones detectadas por *findpat* al aplicarse a cromosomas enteros del hombre. Dicha identificación, permitió no solo confirmar el buen funcionamiento de la herramienta, sino también despertar el interés del estudio particular de las salidas generadas por el programa, aplicado a diversas fuentes de datos. A partir de la identificación lograda, se realizó un análisis comparativo entre las secuencias encontradas entre los cromosomas de la especie humana y algunas bases de datos de secuencias conocidas.

El grupo KAPOW (grupo de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires, que tiene como tema principal de investigación la resolución de problemas sobre secuencias de caracteres) creó el algoritmo *findpat*, el cual fue aplicado al genoma humano. La meta del trabajo fue poder validar el algoritmo *findpat*, como una herramienta útil para encontrar de secuencias de ADN.

Como hecho destacable de los experimentos que realizaron, y apoyándose en el resultado más fuerte, se pudo observar que aproximadamente el 54% de los patrones fueron biológicamente identificados; mientras que el 46% no pudo ser clasificado dentro del conjunto de repeticiones utilizado, no se permite inserción, modificación, o eliminación de caracteres para la búsqueda de los patrones, esto es sin ruido en las muestras [31].



#### **2.2.4. BLAST**

Es un algoritmo que permite a un investigador comparar una secuencia de ADN contra una, o un conjunto de bases de datos, e identificar secuencias dentro de estas con las cuales se asemeja.

El mismo introduce varios refinamientos a la búsqueda en bases de datos, que mejoran el tiempo de búsqueda. BLAST hace énfasis en la velocidad por sobre la sensibilidad. Esto es fundamental para que el algoritmo sea práctico a la hora de buscar en las bases de datos gigantes que están disponibles hoy día.

BLAST no está basado en un algoritmo que garantiza el alineamiento óptimo, sino que usa una heurística que funciona la mayoría de las veces en la práctica, así que, podría fallar con algunas secuencias poco relacionadas entre sí. BLAST es alrededor de 50 veces más rápido que otros algoritmos que garantizan el alineamiento local de secuencias óptimo y usan programación dinámica.

Existen varias implementaciones de la idea que propone BLAST. Se utilizó el BLAST implementado por el National Center for Biotechnology Information (NCBI). Una de las ventajas de esta implementación, es que puede comparar secuencias contra la bases de datos GenBank.

Se pudo constatar que con BLAST se recuperan el 100% de los patrones, sin ninguna clase de ruido, y lo hace en un tiempo promedio de 30 segundos, pero no soporta ningún tipo de ruido, y por ende, con cualquier clase de ruido no recupera nada [URL4].



## CAPÍTULO III. Marco Teórico

Este capítulo trata acerca de las bases teóricas de las técnicas más importantes.

Los conceptos de memoria asociativa son básicos para la elaboración de este trabajo, y se describe brevemente la evolución de los modelos propuestos más importantes relacionados con este tema. Las memorias asociativas que se presentarán en esta sección, son los modelos más representativos.

Se incluyen cuatro modelos clásicos basados en el anillo de los números racionales con las operaciones de multiplicación y adición: Lernmatrix, Correlograph, Linear Associator y Memoria Hopfield, además de tres modelos basados en paradigmas diferentes a la suma de productos, a saber: memorias asociativas Morfológicas, memorias asociativas Alfa-Beta y memorias asociativas Media.

Adicionalmente se habla un poco de ADN para mayor comprensión de la misma.

### 3.1. Memorias asociativas

Una memoria asociativa tiene como propósito principal recuperar correctamente patrones completos a partir de patrones de entrada, los cuales pueden tener una alteración por ruido aditivo, sustractivo o combinados [32], [33], [34].



Una Memoria Asociativa consta de un sistema de entrada y salida, idea que se esquematiza a continuación:

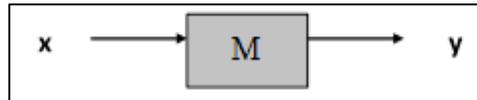


FIGURA 3.1 Sistema de una Memoria Asociativa

En la figura 3.1, los patrones de entrada y salida están representados por vectores columna denotados por  $x$  y  $y$ , respectivamente. Cada uno de los patrones de entrada forma una asociación con el correspondiente patrón de salida, la cual es similar a la una pareja ordenada; por ejemplo, los patrones  $x$  y  $y$  del esquema anterior forman la asociación  $(x,y)$ .

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}; y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \dots \text{ Ec.1}$$

Los patrones de entrada y salida se denotarán con las letras negrillas,  $x$  y  $y$ , agregándoles números naturales como superíndices para efectos de discriminación simbólica. Por ejemplo, a un patrón de entrada  $x_1$  le corresponderá el patrón de salida  $y_1$ , y ambos formarán la asociación  $(x_1,y_1)$ ; del mismo modo, para un número entero positivo  $k$  específico, la asociación correspondiente será  $(x_k,y_k)$ .

El conjunto fundamental de asociaciones queda representado de la siguiente manera:

$$\{(x^\mu,y^\mu) | \mu=1,2,\dots,p \} \quad \dots \text{ Ec.2}$$

donde  $p$  es un número entero positivo que representa la cardinalidad del conjunto fundamental. A los patrones que conforman las asociaciones del conjunto fundamental se les llama patrones fundamentales. La naturaleza del conjunto fundamental proporciona un importante criterio para clasificar las memorias asociativas:



- Una memoria es Autoasociativa si se cumple que  $x^\mu = y^\mu \quad \forall \mu \in \{1, 2, \dots, p\}$ , por lo que uno de los requisitos que se debe de cumplir es que  $n = m$ .
- Una memoria Heteroasociativa es aquella en donde  $\exists \mu \in \{1, 2, \dots, p\}$  para el que se cumple que  $x^\mu \neq y^\mu$ . Nótese que puede haber memorias heteroasociativas con  $n = m$ .

En los problemas donde intervienen las memorias asociativas, se consideran dos fases importantes:

- La fase de aprendizaje, que es donde se genera la memoria asociativa a partir de las  $p$  asociaciones del conjunto fundamental.
- La fase de recuperación que es donde la memoria asociativa opera sobre un patrón de entrada.

A fin de especificar las componentes de los patrones, se requiere la notación para dos conjuntos a los que llamaremos arbitrariamente A y B. Las componentes de los vectores columna que representan a los patrones, tanto de entrada como de salida, serán elementos del conjunto A, y las entradas de la matriz M serán elementos del conjunto B.

No hay requisitos previos ni limitaciones respecto de la elección de estos dos conjuntos, por lo que no necesariamente deben ser diferentes o poseer características especiales. Esto significa que el número de posibilidades para escoger A y B es infinito.

Por convención, cada vector columna que representa a un patrón de entrada tendrá  $n$  componentes cuyos valores pertenecen al conjunto A, y cada vector columna que representa a un patrón de salida tendrá  $m$  componentes cuyos valores pertenecen también al conjunto A.

Es decir:

$$x^\mu \in A_n \quad \& \quad y^\mu \in A_m \quad \forall \mu \in \{1, 2, \dots, p\} \quad \dots \quad \text{Ec.3}$$



La  $j$ -ésima componente de un vector columna se indicará con la misma letra del vector, colocando a  $j$  como subíndice ( $j \in \{1, 2, \dots, n\}$  o  $j \in \{1, 2, \dots, m\}$  según corresponda). La  $j$ -ésima componente del vector columna  $x^\mu$  se representa por  $x_j^\mu$

Con los conceptos básicos ya descritos y con la notación anterior, es posible expresar las dos fases de una memoria asociativa:

- Fase de Aprendizaje (Generación de la memoria asociativa). Encontrar los operadores adecuados y una manera de generar una matriz  $M$  que almacene las  $p$  asociaciones del conjunto fundamental  $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$ , donde  $x^\mu \in A_n$  y  $y^\mu \in A_m \forall \mu \in \{1, 2, \dots, k\}$ . Si  $\exists \mu \in \{1, 2, \dots, k\}$  tal que  $x^\mu \neq y^\mu$ , la memoria será heteroasociativa; si  $m = n$  y  $x^\mu = y^\mu \forall \mu \in \{1, 2, \dots, k\}$ , la memoria será autoasociativa.
- Fase de Recuperación (Operación de la memoria asociativa). Hallar los operadores adecuados y las condiciones suficientes para obtener el patrón fundamental de salida  $y^\mu$ , cuando se opera la memoria  $M$  con el patrón fundamental de entrada  $x^\mu$ ; lo anterior para todos los elementos del conjunto fundamental y para ambos modos: autoasociativo y heteroasociativo.

Se dice que una memoria asociativa  $M$  exhibe recuperación correcta si al presentarle como entrada, en la fase de recuperación, un patrón  $x_\omega$  con  $\omega \in \{1, 2, \dots, k\}$ , ésta responde con el correspondiente patrón fundamental de salida  $y_\omega$ .

Una memoria asociativa bidireccional también es un sistema de entrada y salida, solamente que el proceso es bidireccional. La dirección hacia adelante se describe de la misma forma que una memoria asociativa común: al presentarle una entrada  $x$ , el sistema entrega una salida  $y$ . La dirección hacia atrás se lleva a cabo presentándole al sistema una entrada  $y$  para recibir una salida  $x$ .

### 3.1.1. Lernmatrix de Steinbuch

Karl Steinbuch fue uno de los primeros investigadores en desarrollar un método para codificar información en arreglos cuadrículados conocidos como crossbar [35]. La importancia de la



Lernmatrix [36], [37] se evidencia en una afirmación que hace Kohonen [38] en su artículo de 1972, donde apunta que las matrices de correlación, base fundamental de su innovador trabajo, vinieron a sustituir a la Lernmatrix de Steinbuch.

La Lernmatrix es una memoria heteroasociativa que puede funcionar como un clasificador de patrones binarios si se escogen adecuadamente los patrones de salida; es un sistema de entrada y salida que al operar acepta como entrada un patrón binario  $\mathbf{x}^{\mu} \in A^n$ ,  $A = \{0,1\}$  y produce como salida la clase  $\mathbf{y}^{\mu} \in A^p$  que le corresponde (de entre  $p$  clases diferentes), codificada ésta con un método que en la literatura se le ha llamado *one-hot* [39] El método funciona así: para representar la clase  $k \in \{1, 2, \dots, p\}$ , se asignan a las componentes del vector de salida  $\mathbf{y}^{\mu}$  los siguientes valores:

$$y_k^{\mu} = 1, \text{ y } y_j^{\mu} = 0 \quad \dots \quad \text{Ec.4}$$

para  $j = 1, 2, \dots, k-1, k+1, \dots, p$ .

### Algoritmo de la Lernmatrix

#### Fase de Aprendizaje

Se genera el esquema (*crossbar*) al incorporar la pareja de patrones de entrenamiento  $(\mathbf{x}^{\mu}, \mathbf{y}^{\mu}) \in A^n \times A^p$ . Cada uno de los componentes  $m_{ij}$  de  $M$ , la Lernmatrix de Steinbuch, tiene valor cero al inicio, y se actualiza de acuerdo con la regla  $m_{ij} + \Delta m_{ij}$ , donde:

$$\Delta m_{ij} = \begin{cases} +\varepsilon & \text{si } x_j^{\mu} = 1 = y_i^{\mu} \\ -\varepsilon & \text{si } x_j^{\mu} = 0 \text{ y } y_i^{\mu} = 1 \\ 0 & \text{en otro caso} \end{cases} \quad \dots \quad \text{Ec.5}$$

donde  $\varepsilon$  es una constante positiva escogida previamente y es usual que  $\varepsilon$  es igual a 1.



### Fase de Recuperación

La  $i$ -ésima coordenada  $y_i^\omega$  del vector de clase  $y_\omega \in Ap$  se obtiene como lo indica la siguiente expresión, donde  $\vee$  es el operador máximo:

$$y_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} \cdot x_j^\omega = \vee_{h=1}^p \left[ \sum_{j=1}^n m_{hj} \cdot x_j^\omega \right] \\ 0 & \text{en otro caso} \end{cases} \quad \dots \text{ Ec.6}$$

### 3.1.2. Correlograph de Willshaw, Buneman y Longuet-Higgins

El correlograph es un dispositivo óptico elemental capaz de funcionar como una memoria asociativa [40].

#### Algoritmo del Correlograph

##### Fase de Aprendizaje

La red asociativa se genera al incorporar la pareja de patrones de entrenamiento  $(x^\mu, y^\mu) \in An \times Am$ . Cada uno de los componentes  $m_{ij}$  de la red asociativa  $M$  tiene valor cero al inicio, y se actualiza de acuerdo con la regla:

$$m_{ij} = \begin{cases} 1 & \text{si } y_i^\mu = 1 = x_j^\mu \\ \text{valor anterior} & \text{en otro caso} \end{cases} \quad \dots \text{ Ec.7}$$

##### Fase de Recuperación

Se le presenta a la red asociativa  $M$  un vector de entrada  $x^\omega \in An$ . Se realiza el producto de la matriz  $M$  por el vector  $x^\omega$  y se ejecuta una operación de umbralizado, de acuerdo con la siguiente expresión:

$$y_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} \cdot x_j^\omega \geq u \\ 0 & \text{en otro caso} \end{cases} \quad \dots \text{ Ec.8}$$



donde  $u$  es el valor de umbral. Una estimación aproximada del valor de umbral  $u$  se puede lograr con la ayuda de un número indicador mencionado en el artículo [40] de Willshaw et al. de 1969:  $\log_2(n)$

### 3.1.3. Linear Associator de Anderson-Kohonen

El Linear Associator tiene su origen en los trabajos pioneros de 1972 publicados por Anderson y Kohonen [38, 41].

Para presentar el Linear Associator consideremos de nuevo el conjunto fundamental:

$$\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\} \quad \dots \quad \text{Ec.9}$$

con  $A = \{0, 1\}$ ,  $x^\mu \in A^n$  y  $y^\mu \in A^m$

#### Algoritmo del Linear Associator

Fase de Aprendizaje

Para cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se encuentra la matriz  $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$  de dimensiones  $m \times n$ .

Se suman la  $p$  matrices para obtener la memoria

$$\mathbf{M} = \sum_{\mu=1}^p \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \quad \dots \quad \text{Ec.10}$$

de manera que la  $ij$ -ésima componente de la memoria  $\mathbf{M}$  se expresa así:

$$m_{ij} = \sum_{\mu=1}^p y_i^\mu x_j^\mu \quad \dots \quad \text{Ec.11}$$



### Fase de Recuperación

Esta fase consiste en presentarle a la memoria un patrón de entrada  $\mathbf{x}^\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$  y realizar la operación

$$\mathbf{M} \cdot \mathbf{x}^\omega = \left[ \sum_{\mu=1}^p \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^T \right] \cdot \mathbf{x}^\omega \quad \dots \text{ Ec.12}$$

### 3.1.4. La memoria asociativa Hopfield

El artículo de John J. Hopfield de 1982, publicado por la prestigiosa y respetada National Academy of Sciences (en sus Proceedings), impactó positivamente y trajo a la palestra internacional su famosa memoria asociativa [42].

En el modelo que originalmente propuso Hopfield, cada neurona  $x_i$  tiene dos posibles estados, a la manera de las neuronas de McCulloch-Pitts:  $x_i = 0$  y  $x_i = 1$ ; sin embargo, Hopfield observa que, para un nivel dado de exactitud en la recuperación de patrones, la capacidad de almacenamiento de información de la memoria se puede incrementar por un factor de 2, si se escogen como posibles estados de las neuronas los valores  $x_i = -1$  y  $x_i = 1$  en lugar de los valores originales  $x_i = 0$  y  $x_i = 1$ .

Al utilizar el conjunto  $\{-1, 1\}$  y el valor de umbral cero, la fase de aprendizaje para la memoria Hopfield será similar, en cierta forma, a la fase de aprendizaje del Linear Associator. La intensidad de la fuerza de conexión de la neurona  $x_i$  a la neurona  $x_j$  se representa por el valor de  $m_{ij}$ , y se considera que hay simetría, es decir,  $m_{ij} = m_{ji}$ . Si  $x_i$  no está conectada con  $x_j$  entonces  $m_{ij} = 0$ ; en particular, no hay conexiones recurrentes de una neurona a sí misma, lo cual significa que  $m_{ij} = 0$ . El estado instantáneo del sistema está completamente especificado por el vector columna de dimensión  $n$  cuyas coordenadas son los valores de las  $n$  neuronas.

La memoria Hopfield es autoasociativa, simétrica, con ceros en la diagonal principal. En virtud de que la memoria es autoasociativa, el conjunto fundamental para la memoria Hopfield es  $\{(x_\mu, x_\mu) \mid \mu = 1, 2, \dots, k\}$  con  $x_\mu \in A^n$  y  $A = \{-1, 1\}$



## Algoritmo Hopfield

### Fase de Aprendizaje

La fase de aprendizaje para la memoria Hopfield es similar a la fase de aprendizaje del Linear Associator, con una ligera diferencia relacionada con la diagonal principal en ceros, como se muestra en la siguiente regla para obtener la  $ij$ -ésima componente de la memoria Hopfield  $M$ :

$$m_{ij} = \begin{cases} \sum_{\mu=1}^p x_i^{\mu} x_j^{\mu} & \text{si } i \neq j \\ 0 & \text{si } i = j \end{cases} \quad \dots \text{ Ec.13}$$

### Fase de Recuperación

Si se le presenta un patrón de entrada  $\mathbf{x}$  a la memoria Hopfield, ésta cambiará su estado con el tiempo, de modo que cada neurona  $x_i$  ajuste su valor de acuerdo con el resultado que arroje la

comparación de la cantidad  $\sum_{j=1}^n m_{ij} x_j$  con un valor de umbral, el cual normalmente se coloca en cero.

Se representa el estado de la memoria Hopfield en el tiempo  $t$  por  $x(t)$ ; entonces  $x_i(t)$  representa el valor de la neurona  $x_i$  en el tiempo  $t$  y  $x_i(t+1)$  el valor de  $x_i$  en el tiempo siguiente  $(t+1)$ .

Dado un vector columna de entrada  $\mathbf{x}$ , la fase de recuperación consta de tres pasos:

Para  $t=0$ , se hace  $\mathbf{x}(t) = \mathbf{x}$ ; es decir,  $x_i(0) = \tilde{x}_i$ ,  $\forall i \in \{1,2,3,\dots,n\}$   $\forall i \in \{1,2,3,\dots,n\}$  se calcula  $x_i(t+1)$  de acuerdo con la condición siguiente:

$$x_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} x_j(t) > 0 \\ x_i(t) & \text{si } \sum_{j=1}^n m_{ij} x_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^n m_{ij} x_j(t) < 0 \end{cases} \quad \dots \text{ Ec.14}$$



Se compara  $x_i(t+1)$  con  $x_i(t) \forall i \in \{1, 2, 3, \dots, n\}$ . Si  $x(t+1) = x(t)$  el proceso termina y el vector recuperado es  $\mathbf{x}(0) = \mathbf{x}$ . De otro modo, el proceso continúa de la siguiente manera: los pasos 2 y 3 se iteran tantas veces como sea necesario hasta llegar a un valor  $t = \tau$  para el cual  $x_i(\tau+1) = x_i(\tau) \forall i \in \{1, 2, 3, \dots, n\}$ ; el proceso termina y el patrón recuperado es  $x(\tau)$ .

En el artículo original de 1982, Hopfield había estimado empíricamente que su memoria tenía una capacidad de recuperar  $0.15n$  patrones, y en el trabajo de Abu-Mostafa & St. Jacques [43] se estableció formalmente que una cota superior para el número de vectores de estado arbitrarios estables en una memoria Hopfield es  $n$ .

### 3.1.5. Memorias Asociativas Morfológicas

La diferencia fundamental entre las memorias asociativas clásicas (Lernmatrix, Correlograph, Linear Associator y Memoria Asociativa Hopfield) y las memorias asociativas morfológicas radica en los fundamentos operacionales de éstas últimas, que son las operaciones morfológicas de dilatación y erosión; el nombre de las memorias asociativas morfológicas está inspirado precisamente en estas dos operaciones básicas. Estas memorias rompieron con el esquema utilizado a través de los años en los modelos de memorias asociativas clásicas, que utilizan operaciones convencionales entre vectores y matrices para la fase de aprendizaje y suma de productos para la recuperación de patrones. Las memorias asociativas morfológicas cambian los productos por sumas y las sumas por máximos o mínimos en ambas fases, tanto de aprendizaje como de recuperación [44].

Hay dos tipos de memorias asociativas morfológicas: las memorias max, simbolizadas con M, y las memorias min, cuyo símbolo es W; en cada uno de los dos tipos, las memorias pueden funcionar en ambos modos: heteroasociativo y autoasociativo.

Se definen dos nuevos productos matriciales:

El producto máximo entre D y H, denotado por  $C = D \nabla H$ , es una matriz  $[c_{ij}]$   $m \times n$  cuya  $ij$ -ésima componente  $c_{ij}$  es

$$c_{ij} = \bigvee_{k=1}^r (d_{ik} + h_{kj}) \quad \dots \quad \text{Ec.15}$$



El producto mínimo de D y H denotado por  $C = D \Delta H$ , es una matriz  $[c_{ij}]$   $m \times n$  cuya  $ij$ -ésima componente  $c_{ij}$  es

$$c_{ij} = \bigwedge_{k=1}^r (d_{ik} + h_{kj}) \quad \dots \quad \text{Ec.16}$$

Los productos máximo y mínimo contienen a los operadores máximo y mínimo, los cuales están íntimamente ligados con los conceptos de las dos operaciones básicas de la morfología matemática: dilatación y erosión, respectivamente.

### 3.1.5.1. Memorias Heteroasociativas Morfológicas

Una Memoria Asociativa Morfológica es de carácter heteroasociativo si  $\exists \mu \in \{1, 2, \dots, k\}$  cuando cumple que  $x^\mu \neq y^\mu$ . Existen dos tipos de Memorias Morfológicas Heteroasociativas: Max (M) y la Min (W).

#### Algoritmo de las memorias morfológicas Max

Fase de Aprendizaje

Para cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se usa el producto mínimo para crear la matriz  $y^\mu \Delta (-x^\mu)^t$  de dimensiones  $m \times n$ , donde el negado transpuesto del patrón de entrada  $x^\mu$  se define como  $(-x^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, x_n^\mu)$ .

Se aplica el operador máximo  $\vee$  a las  $p$  matrices para obtener la memoria M.

$$\mathbf{M} = \bigvee_{\mu=1}^p [y^\mu \Delta (-x^\mu)^t] \quad \dots \quad \text{Ec.17}$$

Fase de Recuperación

Esta fase consiste en realizar el producto mínimo  $\Delta$  de la memoria M con el patrón de entrada  $x_\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna y de dimensión  $m$ :

$$y = \mathbf{M} \Delta x_\omega \quad \dots \quad \text{Ec.18}$$



Las fases de aprendizaje y de recuperación de las memorias morfológicas min se obtienen por dualidad.

### 3.1.5.2. Memorias Autoasociativas Morfológicas

Para este tipo de memorias se utilizan los mismos algoritmos descritos anteriormente y que son aplicados a las memorias heteroasociativas; lo único que cambia es el conjunto fundamental.

Para este caso, se considera el siguiente conjunto fundamental:

$$\{(x^\mu, x^\mu) \mid x^\mu \in A_n, \text{ donde } \mu = 1, 2, \dots, k\} \quad \dots \quad \text{Ec.19}$$

#### Algoritmo de las memorias morfológicas Max

Fase de aprendizaje.

Se calculan las matrices  $y^\mu \Delta(-x^\mu)$  para cada uno de los  $k$  elementos del conjunto fundamental de asociaciones  $(x^\mu, x^\mu)$ .

Se obtiene la memoria M aplicando el operador máximo V a las matrices resultante. M queda expresada como:

$$M = \bigvee_{\mu=1}^k [x^\mu \Delta(-x^\mu)^t] = [m_{ij}]_{m \times n} \quad \dots \quad \text{Ec.20}$$

$$m_{ij} = \bigvee_{\mu=1}^k (w_i^\mu - x_j^\mu) \quad \dots \quad \text{Ec.21}$$

Fase de recuperación.

Se calcula el producto mínimo  $M \Delta x^\omega$ , donde  $\omega \in \{1, 2, \dots, k\}$ , obteniendo un vector columna  $x = [x_i]_n$ :

$$x_i = \bigwedge_{j=1}^n (m_{ij} + x_j^\omega) \quad \dots \quad \text{Ec.22}$$



### 3.1.6. Memorias Asociativas Alfa-Beta

Las memorias Alfa-Beta [34] utilizan máximos y mínimos, y dos operaciones binarias originales  $\alpha$  y  $\beta$  de las cuales heredan el nombre.

Para la definición de las operaciones binarias  $\alpha$  y  $\beta$  se deben especificar los conjuntos A y B, los cuales son:

$$A = \{0, 1\} \quad \text{y} \quad B = \{0, 1, 2\} \quad \dots \quad \text{Ec.23}$$

La operación binaria  $\alpha: A \times A \rightarrow B$  se define como:

TABLA 3.1 Definición de B

$x$	$y$	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

La operación binaria  $\beta: B \times A \rightarrow A$  se define como:

TABLA 3.2 Definición de A

$x$	$y$	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1



Los conjuntos A y B, las operaciones binarias  $\alpha$  y  $\beta$  junto con los operadores  $\wedge$  (mínimo) y  $\vee$  (máximo) usuales conforman el sistema algebraico (A, B,  $\alpha$ ,  $\beta$ ,  $\wedge$ ,  $\vee$ ) en el que están inmersas las memorias asociativas Alfa-Beta.

El fundamento teórico de las memorias asociativas Alfa-Beta se presenta en el siguiente capítulo de forma más completa, debido a que estas memorias son la base fundamental para la construcción del modelo de BAM propuesto en este trabajo de tesis.

### 3.2.7. Memorias Asociativas Media

Las Memorias Asociativas Media utilizan los operadores A y B, definidos de la siguiente forma:

$$A(x, y) = x - y \quad \dots \quad \text{Ec.24}$$

$$B(x, y) = x + y \quad \dots \quad \text{Ec.25}$$

Las operaciones utilizadas se describen a continuación.

Sean  $P = [p_{ij}]_{m \times r}$  y  $Q = [q_{ij}]_{r \times n}$  dos matrices.

Operación  $\diamond A$ :

$$P_{m \times r} \diamond A Q_{r \times n} = [f_{ij}A]_{m \times n}$$

donde

$$f_{ij}^A = \mathbf{med}_{k=1}^r A(p_{ik}, q_{k,j}) \quad \dots \quad \text{Ec.26}$$

Operación  $\diamond B$ :

$$P_{m \times r} \diamond B Q_{r \times n} = [f_{ij}B]_{m \times n}$$

donde

$$f_{ij}^B = \mathbf{med}_{k=1}^r B(p_{ik}, q_{k,j}) \quad \dots \quad \text{Ec.27}$$



## Algoritmo Memorias Media

### Fase de Aprendizaje

Para cada  $\xi = 1, 2, \dots, k$ , de cada pareja  $(x^\xi, y^\xi)$  se construye la matriz:

$$[y^\xi \diamond_A (x^\xi)_t]_{m \times n} \quad \dots \quad \text{Ec.28}$$

Se aplica el operador media a las matrices obtenidas anteriormente para obtener la matriz  $M$ , como sigue:

$$\mathbf{M} = \mathbf{med}_{\xi=1}^p \left[ y^\xi \diamond_A (x^\xi)_t \right] \quad \dots \quad \text{Ec.29}$$

El  $ij$ -ésimo componente  $M$  está dado como sigue:

$$m_{ij} = \mathbf{med}_{\xi=1}^p A(y_i^\xi, x_j^\xi) \quad \dots \quad \text{Ec.30}$$

### Fase de Recuperación

Se tienen dos casos:

Caso 1. Recuperación de un patrón fundamental. Un patrón  $x^w$ , con  $w \in \{1, 2, \dots, p\}$  se le presenta a la memoria  $M$  y se realiza la siguiente operación:

$$\mathbf{M} \diamond_B x^w \quad \dots \quad \text{Ec.31}$$

El resultado es un vector columna de dimensión  $n$ , con la  $i$ -ésima componente dada como:

$$\left( \mathbf{M} \diamond_B x^w \right)_i = \mathbf{med}_{j=1}^n B(m_{ij}, x_j^w) \quad \dots \quad \text{Ec.32}$$



Caso 2. Recuperación de un patrón alterado. Un patrón  $\tilde{\mathbf{x}}$ , que es una versión alterada de un patrón  $\mathbf{x}^w$ , se le presenta a la memoria  $\mathbf{M}$  y se realiza la siguiente operación:

$$\mathbf{M} \diamond_B \tilde{\mathbf{x}} \quad \dots \quad \text{Ec.33}$$

De nuevo, el resultado es un vector de dimensión  $n$ , con la  $i$ -ésima componente dada como:

$$(\mathbf{M} \diamond_B \tilde{\mathbf{x}})_i = \mathbf{med}_{j=1}^n \mathbf{B}(m_{ij}, \tilde{x}_j) \quad \dots \quad \text{Ec.34}$$

### 3.3. Ácido Desoxirribonucleico

El ácido desoxirribonucleico, abreviado como ADN, es un ácido nucleico que contiene instrucciones genéticas usadas en el desarrollo y funcionamiento de todos los organismos vivos conocidos y algunos virus, y es responsable de su transmisión hereditaria. El papel principal de la molécula de ADN es el almacenamiento a largo plazo de información. Muchas veces, el ADN es comparado con un plano o una receta, o un código, ya que contiene las instrucciones necesarias para construir otros componentes de las células, como las proteínas y las moléculas de ARN. Los segmentos de ADN que llevan esta información genética son llamados genes, pero las otras secuencias de ADN tienen propósitos estructurales o toman parte en la regulación del uso de esta información genética.

En un principio la cantidad de ADN necesaria, así como, el estado del mismo era un requisito indispensable para poder utilizar estos marcadores. Todas estas limitaciones fueron superadas gracias a la aplicación en Genética Forense de una técnica, la Reacción en Cadena de la Polimerasa (PCR), que supuso una revolución en muchos campos de la Biología y de la Medicina. Esta técnica, ideada en 1989 por Kary B. Mullis, permitió que se pudieran obtener resultados de muestras biológicas con mínimas cantidades de ADN.

A finales de los 90 aparecen en nuestro país nuevas tecnologías que permiten acelerar el proceso de análisis de las muestras. Así el procesamiento de un centenar de perfiles genéticos se podía



realizar en un par de días. La aparición de los analizadores genéticos rápidamente desplazó a los laboriosos geles de acrilamida, que limitaban en gran medida los resultados de un laboratorio. La electroforesis capilar ha sido, durante estos últimos quince años, la herramienta más utilizada por los laboratorios de genética forense.

Durante estos años, los grupos de investigación en este campo se han esforzado en descubrir nuevos marcadores, en nuestro genoma, que faciliten la resolución de los casos forenses: marcadores autosómicos, ADN mitocondrial y cromosoma Y. En esta línea muchos grupos siguen buscando marcadores del tipo SNPs (polimorfismos de un único nucleótido) que nos den información sobre el fenotipo y la procedencia geográfica de una muestra biológica. También se ha intentado unificar en todas las poblaciones el tipo de marcador utilizado, siendo el sistema CODIS (Combined DNA Index System) desarrollado en Estados Unidos por el FBI, el que ha conseguido una mayor difusión.

Su aplicación ha hecho que poco a poco fuera necesario el uso de bases de datos genéticos que permitieran la búsqueda y el rastreo de perfiles comparados, lo que ha permitido la incorporación de Bioinformáticos y Bioestadísticos en los grupos de investigación forense.

Las nuevas plataformas de secuenciación, así como, los nuevos kits de extracción y purificación de las muestras, asociados a robot de extracción, son en este momento las nuevas tecnologías de las que los grupos comenzarán a disponer en breve. Estas nuevas herramientas permitirán un aumento exponencial del número de datos generados. Con lo que los grupos podrán analizar miles de muestras a diario, y tendrán que ser los nuevos software, *expert system*, los que valoren los resultados.

El hecho de utilizar el análisis de ADN para identificar una persona sigue un razonamiento sencillo, esto es, por que cada ser humano es diferente. Dos personas pueden ser más o menos parecidas, sobre todo entre familiares cercanos, pero nunca son idénticos, sólo en el caso de los gemelos univitelinos. Esta diferencia entre las personas se debe a que existen millones de combinaciones posibles de ADN entre un óvulo y un espermatozoide, debido a la recombinación genética que se produce en la meiosis.

Todos los seres humanos tenemos sectores del ADN en común y otros que no los son.



Los tipos de ADN en los que se estudian los marcadores genéticos son: ADN nuclear, ADN mitocondrial y poliformismos de cromosoma Y.

El ADN nuclear se encuentra en el núcleo y se hereda mitad de la madre y mitad del padre.

En el caso del ADN mitocondrial existe una mayor cantidad de copias que de ADN nuclear por célula, de forma que sólo hay una copia de ADN nuclear en una célula mientras que puede haber miles de copias de ADN mitocondrial.

En el caso del cromosoma Y sólo existe en varones y todos los individuos varones emparentados por línea paterna; por lo tanto solo se puede identificar linajes paternos, pero no es posible identificar individuos.

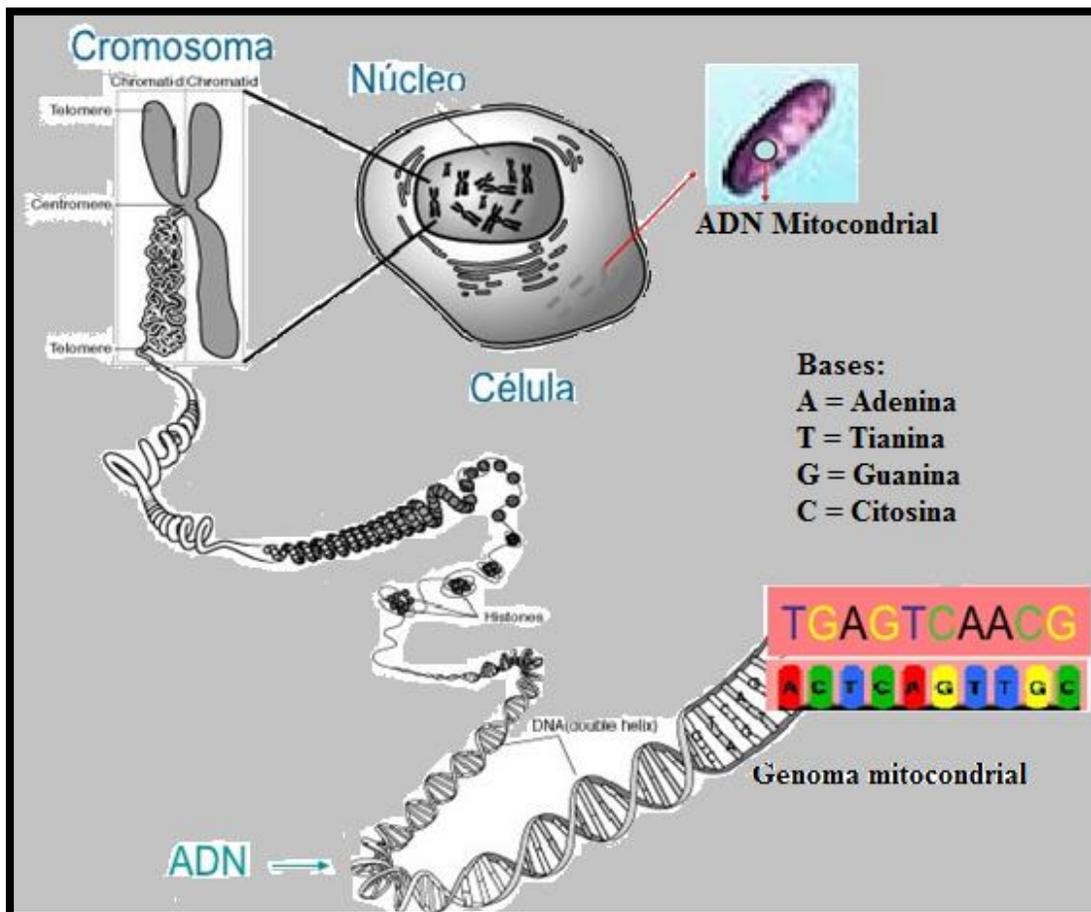


FIGURA 3.2 Estructura de una célula



## CAPÍTULO IV. Diseño del sistema

En este capítulo se detalla la implementación de un sistema de reconocimiento de patrones de ADN para ser utilizada en una interfaz de usuario, realizado en el entorno de programación de Matlab 2014. Para tal fin se propone utilizar memorias asociativas morfológicas.

Para realizar el reconocimiento de los patrones de ADN la metodología está dividida en tres fases; la fase de aprendizaje, que es donde se le presentan los patrones de la base de datos a la memoria; la segunda fase, es donde se puede ingresar cierta cantidad de ruido aditivo o sustractivo y por último la fase de recuperación, donde se puede apreciar si el patrón pertenece a la base de datos y si se ha recuperado correctamente.

### 4.1. Fase de aprendizaje.

La base de datos que se utilizó para el reconocimiento de los patrones de ADN fue obtenida del NDBI (*National Center for Biotechnology Information*), que es parte de la Biblioteca Nacional de Medicina de Estados Unidos. El NCBI ofrece además de algunas herramientas bioinformáticas para el análisis de secuencias de ADN, ARN y proteínas, todas las bases de datos están disponibles en línea de manera gratuitas.



La base de datos consta de 276 patrones, cada patrón tiene 60 caracteres; los caracteres que puede contener este patrón son: adenina (*a*), citosina (*c*), guanina (*g*) y timina (*t*), que representan las cuatro bases nitrogenadas que están contenidas en el ADN.

La base de datos se guardó en un archivo de Excel (*genomas.xlsx*), el cual se lee y se convierte de código ASCII a número entero, el abecedario en minúsculas en el código ASCII empieza en el número decimal 97, por lo tanto, los números que se utilizan para los patrones son 97 para *a*, 99 para la *c*, 103 para la *g* y 119 para la *t*.

Patrón en código ASCII:

gatcacaggt ctatcacctt attaaccact cacgggagct ctccatgcat ttggtatttt

Patrón convertido a números enteros:

Tabla 4.1 Conversión de Patrón

Patrón									
103	97	116	99	97	99	97	103	103	116
99	116	97	116	99	97	99	99	99	116
97	116	116	97	97	99	99	97	99	116
99	97	99	103	103	103	97	103	99	116
99	116	99	99	97	116	103	99	97	116
116	116	103	103	116	97	116	116	116	116

Pasos de la fase de aprendizaje.

Paso 1. Tener convertido todos los patrones de ADN a números enteros.

Paso 2. Para cada asociación  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$ .



Obtener la matriz asociación de cada pareja, la cual se realiza de la manera siguiente:

$$y^\mu \otimes (-x^\mu)^t = \begin{matrix} y_1^\mu - x_1^\mu & \vdots & y_1^\mu - x_n^\mu \\ \dots & \ddots & \dots \\ y_m^\mu - x_1^\mu & \vdots & y_m^\mu - x_n^\mu \end{matrix} \dots \text{ Ec.35}$$

Las memorias Max (M) y Min (W) se obtienen aplicando la operación morfológica de erosión

$$M = \bigvee_{k=1}^p (y^k \otimes (-x^k)^t) \dots \text{ Ec.36}$$

$$W = \bigwedge_{k=1}^p (y^k \otimes (-x^k)^t) \dots \text{ Ec.37}$$

Como se analiza patrón por patrón  $p = 1$ ,  $x = 60$  y  $y = 276$ .

Ejemplo si se le presenta a la memoria el patrón

gatcacaggt ctatcacctt attaaccact cacgggagct ctccatgcat ttggtatttt

Se creó la matriz Y, como una matriz identidad para asociar cada vector columna, con un patrón específico y así poder saber si se recupera el patrón.

Para calcular la memoria Max (M), se utiliza una matriz Y, que es una matriz identidad con una diagonal de 500. Mientras que para calcular la memoria Min (W), se utiliza una matriz Y, que es una matriz identidad con una diagonal de -500.

Se propuso un valor de 500 porque, al hacer pruebas se encontró que un número muy grande comparado con los números que manejan los patrones, la memoria Max (M) es capaz de realizar la recuperación con éxito,

Mientras que para la memoria Min (W) se propuso un valor de -500 dado que al realizar pruebas se encontró que un número mucho menor comparado con los números que manejan los patrones, la memoria Min (W) es capaz de realizar la recuperación con satisfacción.



La memoria morfológica heteroasociativa Max (M), seria:

TABLA 4.2 Ejemplo de Memoria Morfologica heteroasociativa Max (M)

397	403	384	401	403	401	403	397	397	384
401	384	403	384	401	403	401	401	401	384
403	384	384	403	403	401	401	403	401	384
401	403	401	397	397	397	403	397	401	384
401	384	401	401	403	384	397	401	403	384
384	384	397	397	384	403	384	384	384	384

Y la memoria morfológica heteroasociativo min (W), seria:

TABLA 4.3 Ejemplo de Memoria Morfologica heteroasociativa min (W)

-603	-597	-616	-599	-597	-599	-597	-603	-603	-616
-599	-616	-597	-616	-599	-597	-599	-599	-599	-616
-597	-616	-616	-597	-597	-599	-599	-597	-599	-616
-599	-597	-599	-603	-603	-603	-597	-603	-599	-616
-599	-616	-599	-599	-597	-616	-603	-599	-597	-616
-616	-616	-603	-603	-616	-597	-616	-616	-616	-616

## 4.2. Ruido

Cada medida analítica consta de dos componentes. La primera, la señal, lleva la señal relativa al fenómeno que se lleva, la segunda, denominada ruido, está compuesta por información ajena que es indeseada porque degrada la exactitud y la precisión de un análisis.

El termino ruido proviene de la ingeniera radioelectrónica en la que en la presencia de señales no deseadas se pone de manifiesto audio estático o ruido. En la actualidad el término se utiliza



en ciencia y tecnología para describir las desviaciones aleatorias observadas, cuando se repiten medidas de señales que se controlan de forma continua. El ruido es una señal indeseable.

**Ruido aditivo.**

Si  $x \leq \tilde{x}$ . Esto significa que todos los posibles cambios en los valores de las coordenadas de  $x$  para obtener  $\tilde{x}$  consisten en colocar un valor más grande que el mayor valor que puede tomar el patrón en donde se encuentra el patrón; es decir, la única posibilidad de cambio en las coordenadas de  $x$  se traduce en:  $\exists i \in \{1, 2, \dots, n\}$  para el que  $x_i$  puede llegar a tener los valores de cero, dos, seis y diecinueve mientras  $\tilde{x}_i = 21$ . Además:  $x \leq \tilde{x} \rightarrow x_i \leq \tilde{x}_i, \forall i \in \{1, 2, \dots, n\}$ .

**Ruido sustractivo.**

Si  $x \geq \tilde{x}$ . Esto significa que todos los posibles cambios en los valores de las coordenadas de  $x$  para obtener  $\tilde{x}$  consisten en colocar un valor más pequeño que el menor valor que puede tomar el patrón en donde se encuentra el patrón; es decir, la única posibilidad de cambio en las coordenadas de  $x$  se traduce en:  $\exists i \in \{1, 2, \dots, n\}$  para el que  $x_i$  puede llegar a tener los valores de cero, dos, seis y diecinueve mientras  $\tilde{x}_i = -1$ . Además:  $x \geq \tilde{x} \rightarrow x_i \geq \tilde{x}_i, \forall i \in \{1, 2, \dots, n\}$ .

Si el ruido es de 0%, es claro que  $x = \tilde{x}$ .

Las memorias heteroasociativas morfológicas Max (M), son inmunes a cierta cantidad de ruido aditivo, pero son muy sensibles a pequeñas cantidades de ruido sustractivo y las memoria heteroasociativas morfológicas Mín (W) son inmunes a cierta cantidad de ruido sustractivo, pero son muy sensibles a pequeñas cantidades de ruido aditivo.

Usando el mismo patrón del ejemplo anterior

TABLA 4.4 Ejemplo de patrón convertido a números enteros

Patrón									
103	97	116	99	97	99	97	103	103	116
99	116	97	116	99	97	99	99	99	116
97	116	116	97	97	99	99	97	99	116
99	97	99	103	103	103	97	103	99	116
99	116	99	99	97	116	103	99	97	116
116	116	103	103	116	97	116	116	116	116



Ruido aditivo al 10%

TABLA 4.5 Patrón con un 10% de ruido aditivo

Patrón									
103	97	116	99	97	99	97	103	<u>127</u>	116
99	116	97	116	99	97	99	99	99	116
97	116	116	97	97	<u>127</u>	99	97	99	116
99	97	99	103	103	103	97	103	99	<u>127</u>
99	116	99	99	97	116	103	<u>127</u>	97	116
116	116	103	103	<u>127</u>	97	116	<u>127</u>	116	116

Ruido sustractivo 10%

TABLA 4.6 Patrón con un 10% de ruido sustractivo

Patrón									
103	97	<u>0</u>	99	97	99	97	103	103	116
99	116	97	<u>0</u>	99	97	99	99	99	116
97	<u>0</u>	116	97	97	99	99	97	99	116
99	97	99	103	103	103	97	<u>0</u>	99	116
99	116	99	99	97	116	<u>0</u>	99	97	116
116	116	103	103	<u>0</u>	97	116	116	116	116

### 4.3. Fase de recuperación

Fase de Recuperación

Esta fase consiste en realizar el producto mínimo  $\Delta$  de la memoria Max (M) con el patrón de entrada  $x\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna y de dimensión m:

$$\text{Max} = M \Delta x\omega \quad \dots \quad \text{Ec.38}$$



Esta fase consiste en realizar el producto máximo  $\Delta$  de la memoria Min (W) con el patrón de entrada  $x_\omega$ , donde  $\omega \in \{1, 2, \dots, p\}$ , para obtener un vector columna  $y$  de dimensión  $m$ :

$$\text{Min} = W \Delta x_\omega \quad \dots \quad \text{Ec.39}$$

$$M\Delta x = \begin{matrix} y_1 - x_i + x_i \quad n \quad i=1 \\ \vdots \\ y_m - x_i + x_i \quad n \quad i=1 \end{matrix} = y \quad \dots \quad \text{Ec.40}$$

$$W\nabla x = \begin{matrix} y_1 - x_i + x_i \quad n \quad i=1 \\ \vdots \\ y_m - x_i + x_i \quad n \quad i=1 \end{matrix} = y \quad \dots \quad \text{Ec.41}$$

Para el patrón que se utilizó anteriormente.

TABLA 4.7 Ejemplo de patrón convertido a números enteros

Patrón									
103	97	116	99	97	99	97	103	103	116
99	116	97	116	99	97	99	99	99	116
97	116	116	97	97	99	99	97	99	116
99	97	99	103	103	103	97	103	99	116
99	116	99	99	97	116	103	99	97	116
116	116	103	103	116	97	116	116	116	116

$Y_1$  se recupera correctamente utilizando M.

500 481 481 481 481 481 481 481 481 481 483 481 481 481 481 481 481 481 481 481 481 483  
 481 481 481 481 481 481 481 481 481 483 481 481 481 481 481 481 481 481 481 481 481 481  
 481





Se recuperó correctamente el patrón ya que se hizo la asociación del patrón con la primera columna de la matriz  $Y_1$ , y se recuperó correctamente tanto con memoria Max (M) como con memoria Min (W), ya que se puede apreciar un 500 en la primera columna y un -500 en la primera columna respectivamente.

## 4.4 Interfaz de usuario

Para hacer las pruebas correspondientes de la metodología implementada en el programa Matlab y facilitar la visibilidad de dichos resultados, se creó una interfaz de usuario, la cual se muestra a continuación, y se explica de manera breve como utilizarla.

Primero se abre el programa Matlab.

Después de que cargue, seleccionar la carpeta donde se tiene el programa guardado.

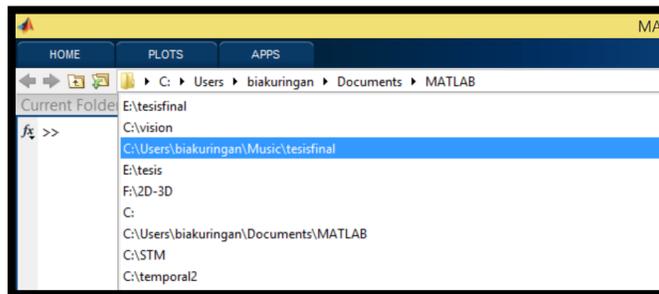


FIGURA 4.1 Ubicar la carpeta de origen

Luego ir al icono que tiene un más y seleccionar Graphical User Interface.

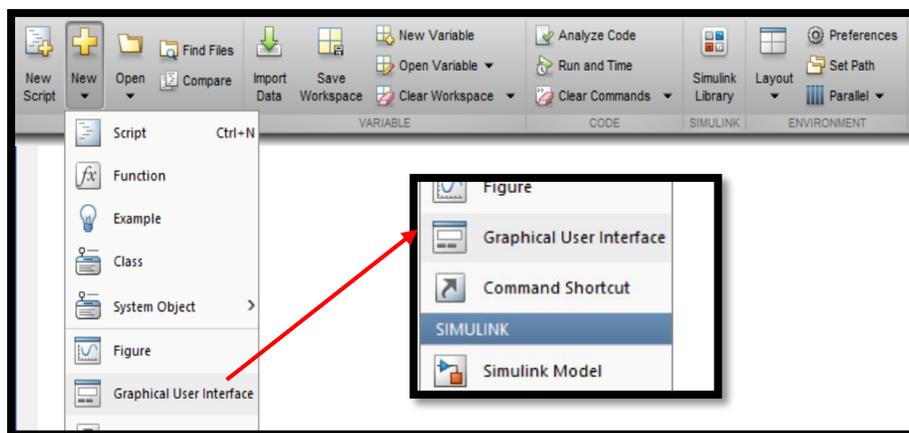


FIGURA 4.2 Abrir la interfaz gráfica de usuario



Ir a la pestaña Open Existing GUI y seleccionar la carpeta donde se tiene el programa.

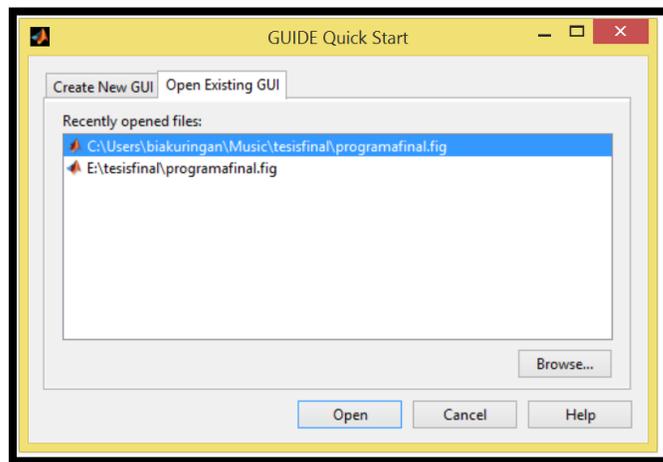


FIGURA 4.3 Seleccionar la ubicación

Se abrirá la plantilla de la interfaz de usuario, y se hace click en el icono que tiene un triángulo de color verde.

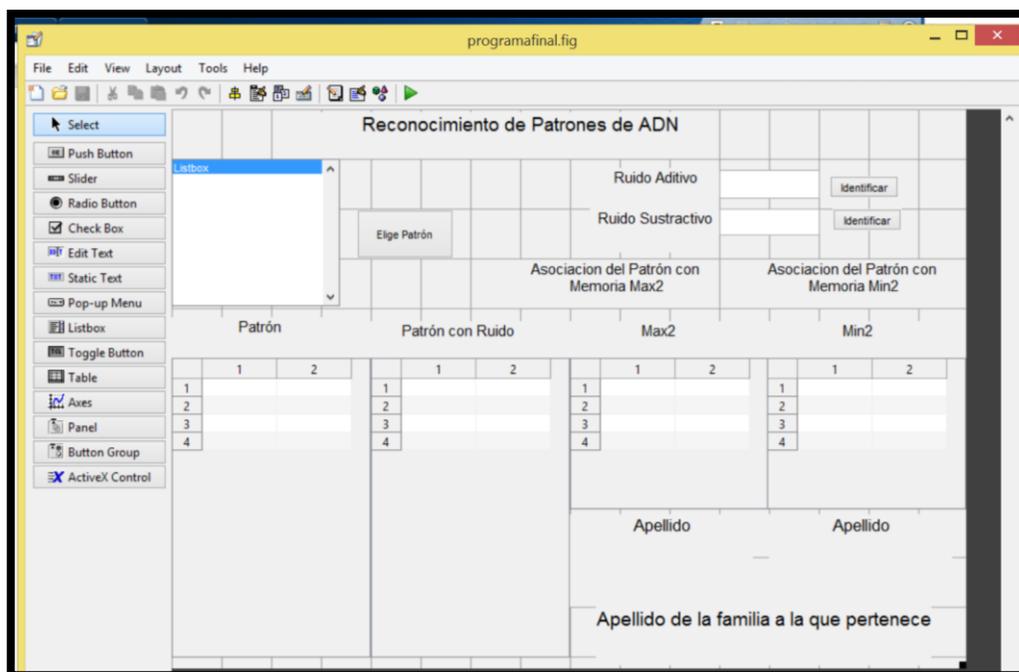


FIGURA 4.4 Plantilla de interfaz de usuario



Entonces la interfaz de usuario se inicializara.

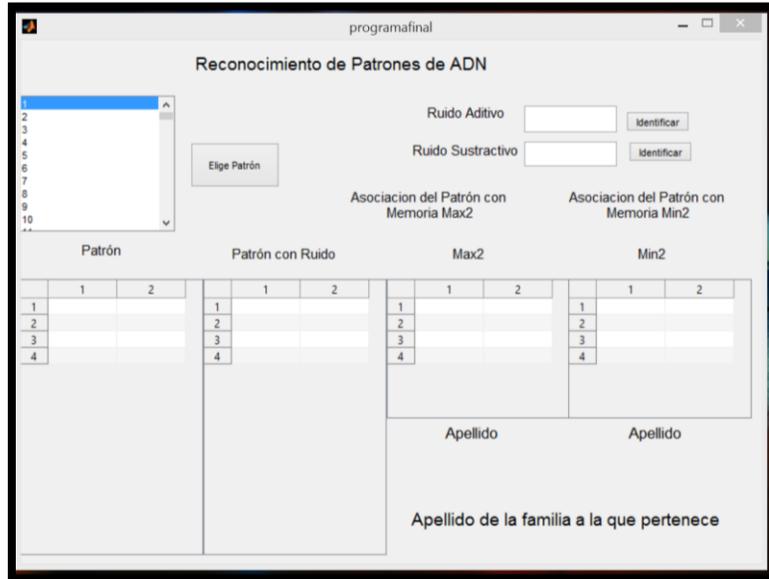


FIGURA 4.5 Interfaz de usuario

Lo que sigue es elegir un patrón de los doscientos setenta y seis posibles de la base de datos, el cual se desplegara, ya convertido en formato decimal, en la tabla Patrón.

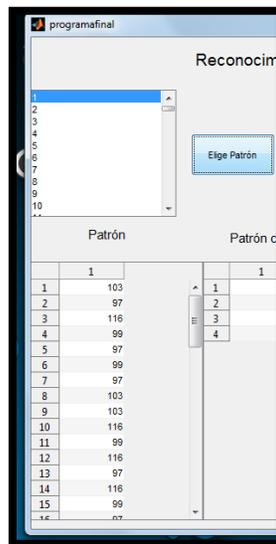


FIGURA 4.6 Selección de patrón en la interfaz de usuario



Después para verificar a que linaje pertenece dicho patrón se puede hacer click en cualquiera de los dos botones que dicen identificar, esto hará que se busque el patrón, sin ninguna alteración, en la base de datos, y nos aparecerá el vector que se obtiene con memoria máxima, y con memoria mínima en cada tabla correspondiente, y debajo de cada tabla nos aparecerá el linaje al que pertenece o nos saldrá si no se logró identificar el patrón.

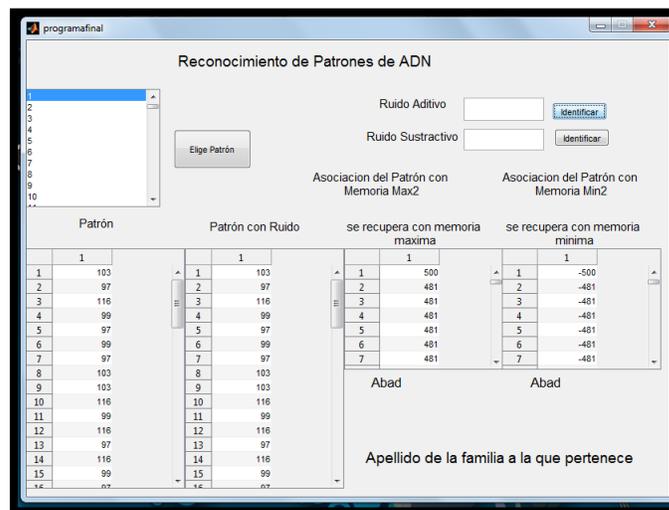


FIGURA 4.7 Probando el botón identificar

Si se escribe un cero ya sea en ruido aditivo o sustractivo, la búsqueda del patrón en la base de datos se efectúa sin alteración alguna.

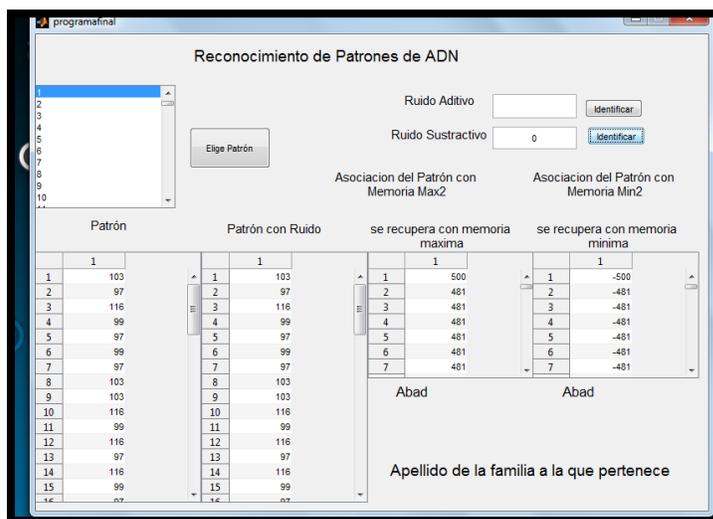


FIGURA 4.8 Identificación con 0% de ruido



Ejemplo de la búsqueda del patrón dos con diez por ciento de ruido aditivo, en donde se puede apreciar que se recuperó con la memoria máxima, y que no se identificó con memoria mínima.

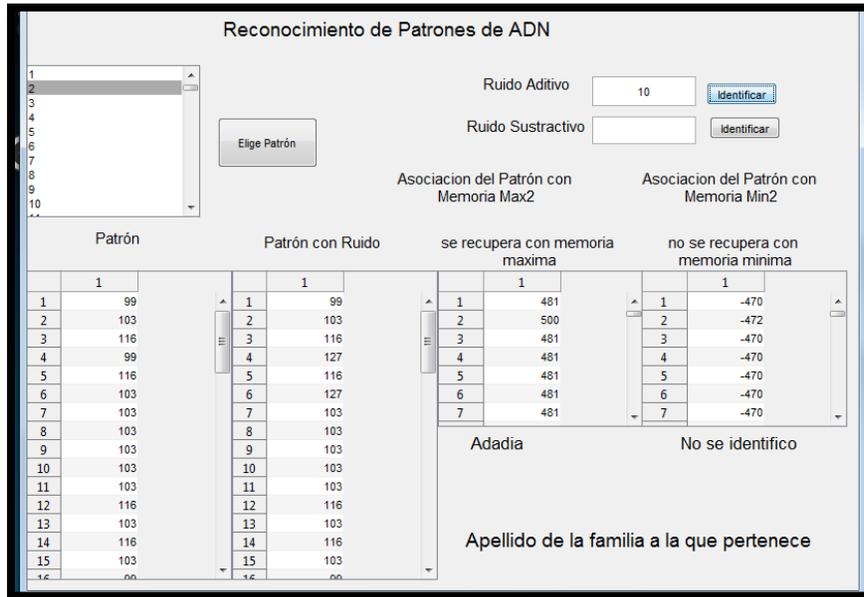


FIGURA 4.9 Identificación con 10% de ruido aditivo

Ejemplo de la búsqueda del patrón dos con diez por ciento de ruido sustractivo, en donde se aprecia que se recuperó con la memoria mínima, y que no se identificó con memoria máxima.

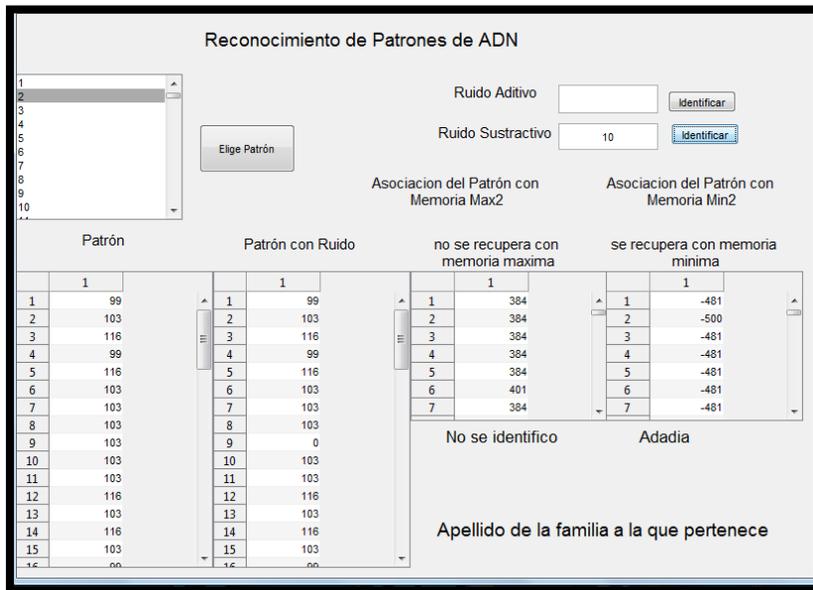


FIGURA 4.10 Identificación con 10% de ruido sustractivo



Si se introduce en ruido un valor menor a cero, o si se introduce un carácter o una cadena de caracteres, la búsqueda se realiza sin ninguna alteración o cero por ciento de ruido, y si se introduce un valor mayor a cien en ruido la búsqueda se realiza con cien por ciento de ruido.



## CAPÍTULO V. Pruebas y resultados

En este capítulo se muestran los resultados experimentales obtenidos al probar la metodología propuesta en el capítulo 4. El desempeño de la misma fue probada con la base de datos obtenida del NCBI (*National Center for Biotechnology Information*), que contiene 276 patrones de ADN, cada una de 60 caracteres. La aplicación correspondiente fue desarrollada en MATLAB. Los experimentos se realizaron en una computadora marca HP, con un procesador Intel Atom z3735, 1GB de RAM, 16Gb de disco Sólido y Sistema Operativo Windows 8.1 con Bing.

### 5.1 Presentación de patrones

En la siguiente tabla se muestran los primeros 5 patrones de la base de datos.

TABLA 5.1 Primeros 5 patrones de la base de datos

1	gatcacaggt	Ctataccct	Attaaccact	cacgggagct	ctccatgcat	ttggtatfff
2	cgtctggggg	gtgtgcacgc	Gatagcattg	cgagacgctg	gagccggagc	accctatgfc
3	gcagtatctg	Tctttgattc	Ctgcctcadc	ctattattta	tcgcacctac	gttcaatatt
4	acaggcgaac	atacttacta	Aagtgtgtta	attaattaat	gctttagga	cataataata
5	acaattgaat	gtctgcacag	Ccgctttcca	cacagacadc	ataacaaaaa	atttccacca



Al convertir los patrones de código ASCII a entero, se obtiene:

TABLA 5.2 Patrón número uno convertido a números enteros

Primer Patrón									
103	97	116	99	97	99	97	103	103	116
99	116	97	116	99	97	99	99	99	116
97	116	116	97	97	99	99	97	99	116
99	97	99	103	103	103	97	103	99	116
99	116	99	99	97	116	103	99	97	116
116	116	103	103	116	97	116	116	116	116

TABLA 5.3 Patrón número dos convertido a números enteros

Segundo Patrón									
99	103	116	99	116	103	103	103	103	103
103	116	103	116	103	99	97	99	103	99
103	97	116	97	103	99	97	116	116	103
99	103	97	103	97	99	103	99	116	103
103	97	103	99	99	103	103	97	103	99
97	99	99	99	116	97	116	103	116	99

TABLA 5.4 Patrón número tres convertido a números enteros

Tercer Patrón									
103	99	97	103	116	97	116	99	116	103
116	99	116	116	116	103	97	116	116	99
99	116	103	99	99	116	99	97	116	99
99	116	97	116	116	97	116	116	116	97
116	99	103	99	97	99	99	116	97	99
103	116	116	99	97	97	116	97	116	116



TABLA 5.5 Patrón número cuatro convertido a números enteros

Cuarto Patrón									
97	99	97	103	103	99	103	97	97	99
97	116	97	99	116	116	97	99	116	97
97	97	103	116	103	116	103	116	116	97
97	116	116	97	97	116	116	97	97	116
103	99	116	116	103	116	97	103	103	97
99	97	116	97	97	116	97	97	116	97

TABLA 5.6 Patrón número cinco convertido a números enteros

Quinto Patrón									
97	99	97	97	116	116	103	97	97	116
103	116	99	116	103	99	97	99	97	103
99	99	103	99	116	116	116	99	99	97
99	97	99	97	103	97	99	97	116	99
97	116	97	97	99	97	97	97	97	97
97	116	116	116	99	99	97	99	99	97

## 5.1 Fase de aprendizaje

Se le presentan los patrones a la memoria asociativa morfológica y se obtienen las memorias Max (M) y Min (W) aplicando la operación morfológica de erosión.

TABLA 5.7 Memoria Max (M) del primer patrón

397	403	384	401	403	401	403	397	397	384	401	384
403	384	401	403	401	401	401	384	403	384	384	403
403	401	401	403	401	384	401	403	401	397	397	397
403	397	401	384	401	384	401	401	403	384	397	401
403	384	384	384	397	397	384	403	384	384	384	384



TABLA 5.8 Memoria Max (M) del segundo patrón

401	397	384	401	384	397	397	397	397	397	397	384
397	384	397	401	403	401	397	401	397	403	384	403
397	401	403	384	384	397	401	397	403	397	403	401
397	401	384	397	397	403	397	401	401	397	397	403
397	401	403	401	401	401	384	403	384	397	384	401

TABLA 5.9 Memoria Max (M) del tercer patrón

397	401	403	397	384	403	384	401	384	397	384	401
384	384	384	397	403	384	384	401	401	384	397	401
401	384	401	403	384	401	401	384	403	384	384	403
384	384	384	403	384	401	397	401	403	401	401	384
403	401	397	384	384	401	403	403	384	403	384	384

TABLA 5.10 Memoria Max (M) del cuarto patrón

403	401	403	397	397	401	397	403	403	401	403	384
403	401	384	384	403	401	384	403	403	403	397	384
397	384	397	384	384	403	403	384	384	403	403	384
384	403	403	384	397	401	384	384	397	384	403	397
397	403	401	403	384	403	403	384	403	403	384	403

TABLA 5.11 Memoria Max (M) del quinto patrón

403	401	403	403	384	384	397	403	403	384	397	384
401	384	397	401	403	401	403	397	401	401	397	401
384	384	384	401	401	403	401	403	401	403	397	403
401	403	384	401	403	384	403	403	401	403	403	403
403	403	403	384	384	384	401	401	403	401	401	403



TABLA 5.12 Memoria Min (W) del primer patrón

-603	-597	-616	-599	-597	-599	-597	-603	-603	-616	-599	-616
-597	-616	-599	-597	-599	-599	-599	-616	-597	-616	-616	-597
-597	-599	-599	-597	-599	-616	-599	-597	-599	-603	-603	-603
-597	-603	-599	-616	-599	-616	-599	-599	-597	-616	-603	-599
-597	-616	-616	-616	-603	-603	-616	-597	-616	-616	-616	-616

TABLA 5.13 Memoria Min (W) del segundo patrón

-599	-603	-616	-599	-616	-603	-603	-603	-603	-603	-603	-616
-603	-616	-603	-599	-597	-599	-603	-599	-603	-597	-616	-597
-603	-599	-597	-616	-616	-603	-599	-603	-597	-603	-597	-599
-603	-599	-616	-603	-603	-597	-603	-599	-599	-603	-603	-597
-603	-599	-597	-599	-599	-599	-616	-597	-616	-603	-616	-599

TABLA 5.14 Memoria Min (W) del tercer patrón

-603	-599	-597	-603	-616	-597	-616	-599	-616	-603	-616	-599
-616	-616	-616	-603	-597	-616	-616	-599	-599	-616	-603	-599
-599	-616	-599	-597	-616	-599	-599	-616	-597	-616	-616	-597
-616	-616	-616	-597	-616	-599	-603	-599	-597	-599	-599	-616
-597	-599	-603	-616	-616	-599	-597	-597	-616	-597	-616	-616

TABLA 5.15 Memoria Min (W) del cuarto patrón

-597	-599	-597	-603	-603	-599	-603	-597	-597	-599	-597	-616
-597	-599	-616	-616	-597	-599	-616	-597	-597	-597	-603	-616
-603	-616	-603	-616	-616	-597	-597	-616	-616	-597	-597	-616
-616	-597	-597	-616	-603	-599	-616	-616	-603	-616	-597	-603
-603	-597	-599	-597	-616	-597	-597	-616	-597	-597	-616	-597



TABLA 5.16 Memoria Min (W) del quinto patrón

-597	-599	-597	-597	-616	-616	-603	-597	-597	-616	-603	-616
-599	-616	-603	-599	-597	-599	-597	-603	-599	-599	-603	-599
-616	-616	-616	-599	-599	-597	-599	-597	-599	-597	-603	-597
-599	-597	-616	-599	-597	-616	-597	-597	-599	-597	-597	-597
-597	-597	-597	-616	-616	-616	-599	-599	-597	-599	-599	-597

Se le agrega ruido aditivo de 25% y de 50% a los patrones 2 y 3 respectivamente, y ruido sustractivo a los patrones 4 y 5 con un porcentaje de 25% y 50% respectivamente y el patrón 1 se le agrega ruido al 0%.

## 5.2 Fase de recuperación

Se aplica la operación morfológica de dilatación para obtener la memoria Y, para poder apreciar si se recuperó correctamente el patrón.

El patrón 1 se recupera correctamente con la memoria  $Y_1$  utilizando memoria Max (M) para patrón 1.

$Y_1$  recupera correctamente.

500 481 481 481 481 481 481 481 481 481 483 481 481 481 481 481 481 481 481 481 481 481 483  
481 481 481 481 481 481 481 481 481 481 483 481 481 481 481 481 481 481 481 481 481 481 481  
481  
481  
481 483  
481  
481  
481  
481  
481 481





Para el patrón 1 con 0% de ruido la interfaz es la siguiente:

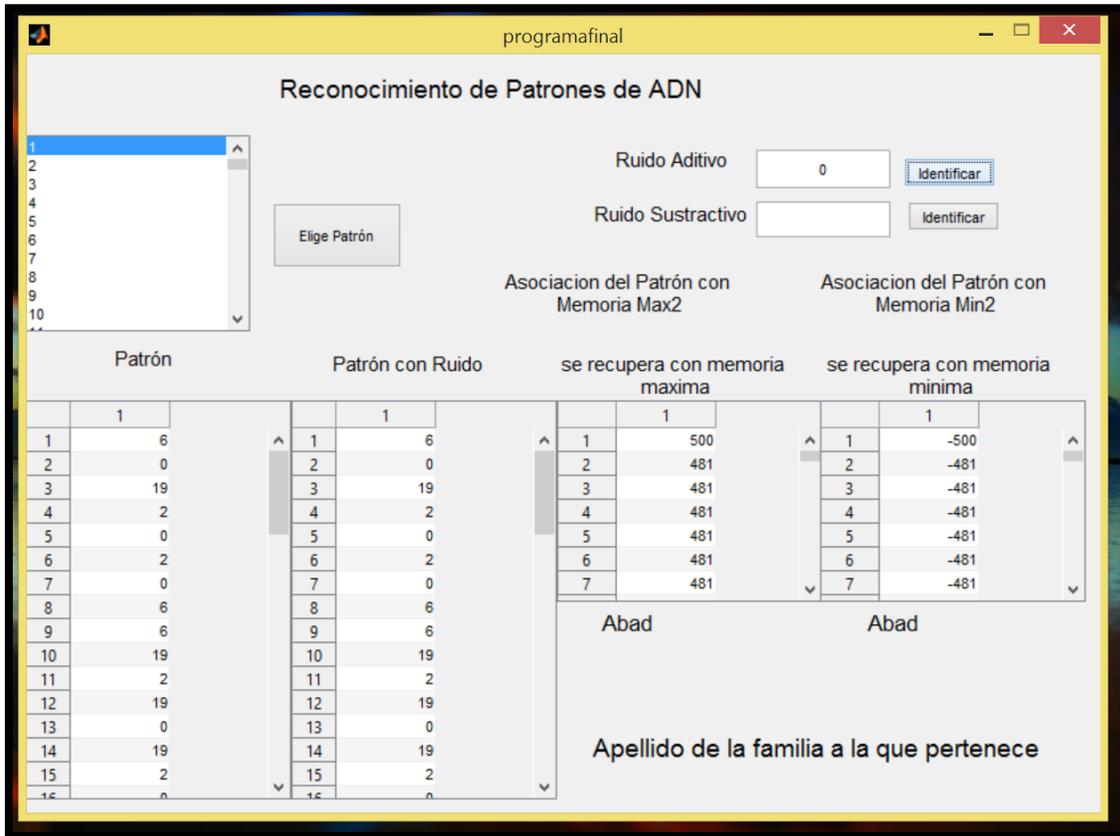


FIGURA 5.1 Interfaz de usuario para el patrón uno con 0% de ruido

El patrón 2 se recupera correctamente con la memoria  $Y_2$  utilizando memoria Max (M) para patrón 2.

$Y_2$  recupera correctamente.

481 500 481 481 481 481 481 481 481 483 481 481 481 481 483 483 481 481 481 481 481 483  
 481 481 483 481 481 481 481 481 481 483 483 481 481 483 481 481 481 481 481 481 481  
 481 483 481 483 483 481 481 481 481 483 481 481 481 481 481 481 481 481 483 481 481 481  
 481 483 481 481 481 481 481 481 481 481 481 481 481 481 481 481 481 481 481 481 481 483  
 481 481 481 481 481 483 481 481 481 481 481 481 481 481 481 483 481 483 483 481 481 483 481  
 481 481 481 481 481 481 483 483 481 481 481 481 481 481 481 481 481 481 481 481 481 483 481 481





Para el patrón 2 con 25% de ruido aditivo la interfaz es la siguiente:

**Reconocimiento de Patrones de ADN**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11

Elige Patrón

Ruido Aditivo

Ruido Sustractivo

Asociacion del Patrón con Memoria Max2

Patrón	Patrón con Ruido
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16

Asociacion del Patrón con Memoria Min2

se recupera con memoria maxima	no se recupera con memoria minima
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Adadia No se identifico

Apellido de la familia a la que pertenece

FIGURA 5.2 Interfaz de usuario para el patrón dos con 25% de ruido aditivo

El patrón 3 se recupera correctamente con la memoria  $Y_3$  utilizando memoria Max (M) para patrón 3.

$Y_3$  recupera correctamente.

481 481 500 481 483 483 481 481 487 481 481 487 481 483 483 481 483 494 483 481 483 481  
481 481 496 481 481 481 487 481 481 487 481 487 481 481 483 481 481 481 483 481 483 481  
481 481 481 494 483 481 481 481 481 481 481 481 481 481 481 481 483 481 481 483 483 494  
498 481 481 481 481 481 483 481 481 481 481 481 483 481 481 481 481 481 481 481 481 481  
483 483 481 481 481 481 481 481 481 481 481 483 481 481 481 481 481 481 483 481 481  
481 481 481 481 483 483 481 481 481 481 483 481 481 481 487 481 481 481 481 481 481 483  
481 481 481 483 481 481 483 481 481 483 481 481 481 483 487 481 481 481 481 483 481 481  
483 483 494 481 481 483 483 481 481 481 483 481 481 481 483 481 481 481 481 481 481 481



481 481 481 481 481 481 481 481 481 481 481 483 487 481 481 481 481 483 481 481 481 481  
 481 481 481 496 481 481 481 481 481 481 483 483 481 481 483 483 483 481 481 483 481 481  
 481 481 481 481 481 481 481 481 481 483 481 483 494 481 481 481 481 494 481 481 487 481  
 481 481 496 481 481 481 481 483 481 481 483 481 481 487 483 481 481 481 481 483 481 481  
 481 481 481 481 483 481 483 481 481 494 483 481

El patrón 3 no se recupera correctamente con la memoria  $Y_3$  utilizando memoria Min (W) para patrón 3, y es igual  $Y_2$  del patrón 2 que utiliza Min (W).

Para el patrón 3 con 50% de ruido aditivo la interfaz es la siguiente:

**Reconocimiento de Patrones de ADN**

Elige Patrón

Ruido Aditivo

Ruido Sustractivo

Asociacion del Patrón con Memoria Max2

Asociacion del Patrón con Memoria Min2

Patrón		Patrón con Ruido		se recupera con memoria maxima		no se recupera con memoria minima	
1	6	1	6	1	481	1	-479
2	2	2	2	2	481	2	-479
3	0	3	0	3	500	3	-479
4	6	4	21	4	481	4	-479
5	19	5	19	5	481	5	-479
6	0	6	0	6	481	6	-479
7	19	7	21	7	481	7	-479
8	2	8	21				
9	19	9	21				
10	6	10	6				
11	19	11	19				
12	2	12	21				
13	19	13	19				
14	19	14	21				
15	19	15	19				
16	6	16	21				

Abascal

No se identifico

Apellido de la familia a la que pertenece

FIGURA 5.3 Interfaz de usuario para el patrón tres con 50% de ruido aditivo





-481  
-481  
-481  
-481  
-481  
-481  
-481  
481 -481 -481 -481 -481 -481

Para el patrón 4 con 25% de ruido sustractivo la interfaz es la siguiente:

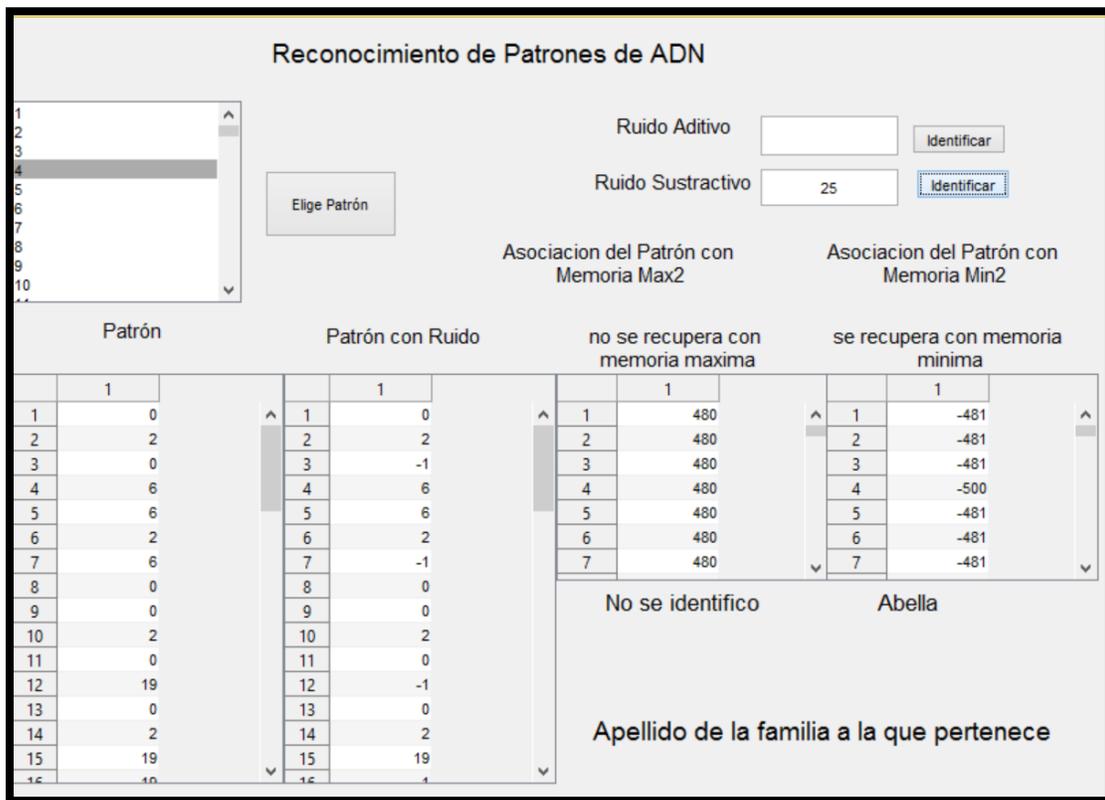


FIGURA 5.4 Interfaz de usuario para el patrón cuatro con 25% de ruido sustractivo

El patrón 5 no se recupera correctamente con la memoria  $Y_5$  utilizando memoria Max (M) para patrón 5.





-481 -481 -481 -481 -481 -481 -481 -481 -483 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481  
-481  
-481 -481 -481 -483 -481 -481 -481 -481 -483 -481 -483 -481 -481 -481 -483 -481 -481 -481 -481  
-481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -483 -481 -481 -481 -481 -481  
-481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -483 -483 -481 -481 -481 -481  
-481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481 -481  
-481 -481 -481 -481 -481 -481

Para el patrón 5 con 50% de ruido sustractivo la interfaz es la siguiente:

**Reconocimiento de Patrones de ADN**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

Elige Patrón

Ruido Aditivo

Ruido Sustractivo

Asociacion del Patrón con Memoria Max2

Asociacion del Patrón con Memoria Min2

Patrón		Patrón con Ruido		no se recupera con memoria maxima		se recupera con memoria minima	
	1		1		1		1
1	0	1	0	1	480	1	-481
2	2	2	2	2	480	2	-483
3	0	3	0	3	480	3	-483
4	0	4	-1	4	480	4	-481
5	19	5	-1	5	480	5	-500
6	19	6	-1	6	480	6	-481
7	6	7	6	7	480	7	-481
8	0	8	0				
9	0	9	-1				
10	19	10	19				
11	6	11	6				
12	19	12	-1				
13	2	13	-1				
14	19	14	19				
15	6	15	6				
16	0	16	0				

No se identifico      Abellán

Apellido de la familia a la que pertenece

FIGURA 5.5 Interfaz de usuario para el quinto patrón con 50% de ruido sustractivo



Se hicieron pruebas para los 276 patrones de ADN, utilizando 0% ruido, y después ruido aditivo en diferentes porcentajes, y ruido sustractivo de igual manera, se tomó el tiempo de recuperación de cada patrón.

TABLA 5.17 Tabla de tiempos de recuperación de patrones sin ruido

Patrón	Tiempo (seg)						
1	1.215597	70	1.197795	139	1.238221	208	1.195778
2	1.205478	71	1.198514	140	1.251726	209	1.197046
3	1.219994	72	1.193541	141	1.21823	210	1.199971
4	1.205073	73	1.197377	142	1.199766	211	1.201731
5	1.212296	74	1.19645	143	1.201971	212	1.19904
6	1.206047	75	1.197194	144	1.199997	213	1.200807
7	1.218222	76	1.199484	145	1.194999	214	1.200211
8	1.200887	77	1.197441	146	1.196917	215	1.204784
9	1.204163	78	1.197838	147	1.198744	216	1.20038
10	1.197738	79	1.195507	148	1.199792	217	1.197668
11	1.199112	80	1.196353	149	1.200443	218	1.199372
12	1.20616	81	1.195897	150	1.198505	219	1.201501
13	1.200012	82	1.198376	151	1.196364	220	1.199983
14	1.207248	83	1.200157	152	1.198064	221	1.19761
15	1.205214	84	1.212875	153	1.201629	222	1.19757
16	1.240836	85	1.19677	154	1.199634	223	1.198856
17	1.215132	86	1.195741	155	1.205588	224	1.197591
18	1.266462	87	1.196461	156	1.199514	225	1.19526
19	1.226761	88	1.196742	157	1.202478	226	1.197552
20	1.241015	89	1.19951	158	1.200126	227	1.196508
21	1.265246	90	1.197705	159	1.198061	228	1.200651
22	1.216362	91	1.197851	160	1.201848	229	1.19537
23	1.207722	92	1.196517	161	1.196488	230	1.1957
24	1.214377	93	1.193525	162	1.19864	231	1.201754
25	1.232603	94	1.19684	163	1.199239	232	1.198901
26	1.21883	95	1.193745	164	1.201488	233	1.19593
27	1.206571	96	1.196995	165	1.201968	234	1.2
28	1.203746	97	1.199006	166	1.200661	235	1.198082
29	1.20332	98	1.199435	167	1.202073	236	1.201207
30	1.203213	99	1.201943	168	1.207122	237	1.197267
31	1.199061	100	1.200499	169	1.201987	238	1.198645



32	1.201987	101	1.199428	170	1.206375	239	1.196442
33	1.202571	102	1.201265	171	1.205302	240	1.202172
34	1.202239	103	1.262076	172	1.21539	241	1.197103
35	1.196502	104	1.291662	173	1.422627	242	1.199739
36	1.196315	105	1.405795	174	1.209132	243	1.196738
37	1.200004	106	1.412709	175	1.210576	244	1.197961
38	1.198256	107	1.36065	176	1.209278	245	1.197623
39	1.197483	108	1.197528	177	1.206346	246	1.199541
40	1.200918	109	1.196356	178	1.219402	247	1.196458
41	1.198936	110	1.200988	179	1.205921	248	1.195896
42	1.203717	111	1.196569	180	1.214502	249	1.201247
43	1.20309	112	1.197462	181	1.206007	250	1.199925
44	1.202048	113	1.197054	182	1.225604	251	1.198612
45	1.198235	114	1.197245	183	1.202502	252	1.197497
46	1.198503	115	1.198422	184	1.205655	253	1.206621
47	1.197335	116	1.199408	185	1.203639	254	1.199501
48	1.199366	117	1.210758	186	1.205937	255	1.199948
49	1.194733	118	1.266926	187	1.203072	256	1.199484
50	1.209609	119	1.266778	188	1.20442	257	1.200317
51	1.200502	120	1.216542	189	1.203371	258	1.195702
52	1.201378	121	1.202188	190	1.205616	259	1.197571
53	1.211635	122	1.198661	191	1.205355	260	1.200171
54	1.198206	123	1.198991	192	1.20274	261	1.199871
55	1.199433	124	1.199176	193	1.199368	262	1.197033
56	1.193927	125	1.19839	194	1.197274	263	1.205679
57	1.197654	126	1.202773	195	1.200169	264	1.227821
58	1.197698	127	1.244926	196	1.201385	265	1.256385
59	1.196882	128	1.219482	197	1.201139	266	1.296144
60	1.196886	129	1.248606	198	1.198029	267	1.294639
61	1.200858	130	1.240535	199	1.201135	268	1.304427
62	1.196932	131	1.200442	200	1.206144	269	1.198219
63	1.198829	132	1.194706	201	1.204342	270	1.201349
64	1.199432	133	1.197893	202	1.203442	271	1.199726
65	1.196448	134	1.200889	203	1.19888	272	1.205114
66	1.193829	135	1.201791	204	1.203509	273	1.201219
67	1.195365	136	1.200967	205	1.197346	274	1.199919
68	1.199368	137	1.247182	206	1.206444	275	1.199983
69	1.196275	138	1.281	207	1.199589	276	1.196948



Tiempo de recuperación de los 276 patrones con ruido al 0%, todos se recuperaron correctamente, tiempo promedio de 1.20861921s.

TABLA 5.18 Tiempo promedio de recuperación de todos los patrones, para cada porcentaje de ruido aditivo

Ruido	Tiempo promedio (seg)	Ruido	Tiempo promedio(seg)	Ruido	Tiempo promedio(seg)
1%	1.22098	36%	1.203886	71%	1.201855
2%	1.208147	37%	1.204056	72%	1.21016
3%	1.203447	38%	1.212424	73%	1.205722
4%	1.206653	39%	1.211931	74%	1.201721
5%	1.216819	40%	1.204097	75%	1.203599
6%	1.206786	41%	1.213452	76%	1.205226
7%	1.202462	42%	1.215903	77%	1.199802
8%	1.202857	43%	1.208603	78%	1.203498
9%	1.203375	44%	1.216378	79%	1.205755
10%	1.199442	45%	1.217574	80%	1.202102
11%	1.200333	46%	1.210331	81%	1.207588
12%	1.199871	47%	1.216093	82%	1.20638
13%	1.202245	48%	1.21319	83%	1.201807
14%	1.19937	49%	1.205051	84%	1.205595
15%	1.201378	50%	1.226311	85%	1.209584
16%	1.202017	51%	1.209606	86%	1.20876
17%	1.20544	52%	1.209123	87%	1.201475
18%	1.197918	53%	1.215181	88%	1.217694
19%	1.200282	54%	1.205457	89%	1.19966
20%	1.204638	55%	1.215632	90%	1.203961
21%	1.203112	56%	1.206512	91%	1.200672
22%	1.199128	57%	1.207903	92%	1.2017



23%	1.198271	58%	1.204039	93%	1.203133
24%	1.200596	59%	1.210342	94%	1.205804
25%	1.199316	60%	1.221876	95%	1.2039
26%	1.200525	61%	1.219352	96%	1.199978
27%	1.202005	62%	1.211009	97%	1.206488
28%	1.206218	63%	1.224651	98%	1.202962
29%	1.204432	64%	1.233263	99%	1.190422
30%	1.206653	65%	1.205769	100%	1.221703
31%	1.205776	66%	1.210599		
32%	1.204187	67%	1.218466		
33%	1.20409	68%	1.206418		
34%	1.222507	69%	1.221492		
35%	1.204478	70%	1.206283		

La recuperación se realiza correctamente en todos los patrones hasta el 55% de ruido aditivo, del 56% en adelante empieza a descender el número de patrones que se recupera.

TABLA 5.19 Tiempo promedio de recuperación de todos los patrones, para cada porcentaje de ruido sustractivo

Ruido	Tiempo promedio(seg)	Ruido	Tiempo promedio(seg)	Ruido	Tiempo promedio(seg)
1%	1.224918	36%	1.224712	71%	1.20416
2%	1.217754	37%	1.221878	72%	1.204443
3%	1.221818	38%	1.200111	73%	1.211983
4%	1.25146	39%	1.215624	74%	1.202384
5%	1.218957	40%	1.229905	75%	1.211202
6%	1.213718	41%	1.224109	76%	1.216799
7%	1.212986	42%	1.223135	77%	1.211116
8%	1.231609	43%	1.210204	78%	1.205681



9%	1.247111	44%	1.211005	79%	1.203453
10%	1.216931	45%	1.250762	80%	1.206118
11%	1.221755	46%	1.205787	81%	1.226329
12%	1.219723	47%	1.218774	82%	1.21211
13%	1.218856	48%	1.220282	83%	1.208444
14%	1.250923	49%	1.210531	84%	1.209439
15%	1.22975	50%	1.207666	85%	1.204899
16%	1.223145	51%	1.218945	86%	1.207975
17%	1.228207	52%	1.215305	87%	1.204332
18%	1.223909	53%	1.207633	88%	1.206025
19%	1.222672	54%	1.204293	89%	1.203701
20%	1.235692	55%	1.210505	90%	1.210282
21%	1.214229	56%	1.205674	91%	1.212813
22%	1.214823	57%	1.207444	92%	1.213617
23%	1.229727	58%	1.202706	93%	1.204777
24%	1.253836	59%	1.206498	94%	1.205152
25%	1.209422	60%	1.210595	95%	1.217078
26%	1.214011	61%	1.209574	96%	1.20601
27%	1.241637	62%	1.207257	97%	1.208249
28%	1.219134	63%	1.221152	98%	1.210063
29%	1.21351	64%	1.211897	99%	1.218773
30%	1.21238	65%	1.208422	100%	1.201937
31%	1.233373	66%	1.205569		
32%	1.216443	67%	1.201015		
33%	1.216228	68%	1.200879		
34%	1.208747	69%	1.207348		
35%	1.216202	70%	1.204421		



La recuperación se realiza correctamente en todos los patrones hasta el 55% de ruido sustractivo, del 56% en adelante empieza a descender el número de patrones que se recupera.

A continuación se presenta los casos en los cuales se recuperan los patrones con un porcentaje de ruido Aditivo y con qué memoria recupera el patrón.

TABLA 5.20 Porcentaje de recuperación de patrones con ruido Aditivo.

Porcentaje de ruido Aditivo (%)	Porcentaje de Patrones recuperados (%)	Se recupera con memoria Max(M)	Se recupera con memoria Min(W)
0	100	Si	Si
5	100	Si	No
10	100	Si	No
15	100	Si	No
20	100	Si	No
25	100	Si	No
30	100	Si	No
35	100	Si	No
40	100	Si	No
45	100	Si	No
50	100	Si	No
55	100	Si	No
60	99	Si	No
65	96	Si	No
70	90	Si	No
75	82	Si	No
80	49	Si	No
85	22	Si	No
90	6	Si	No
95	0	No	No
100	0	No	No

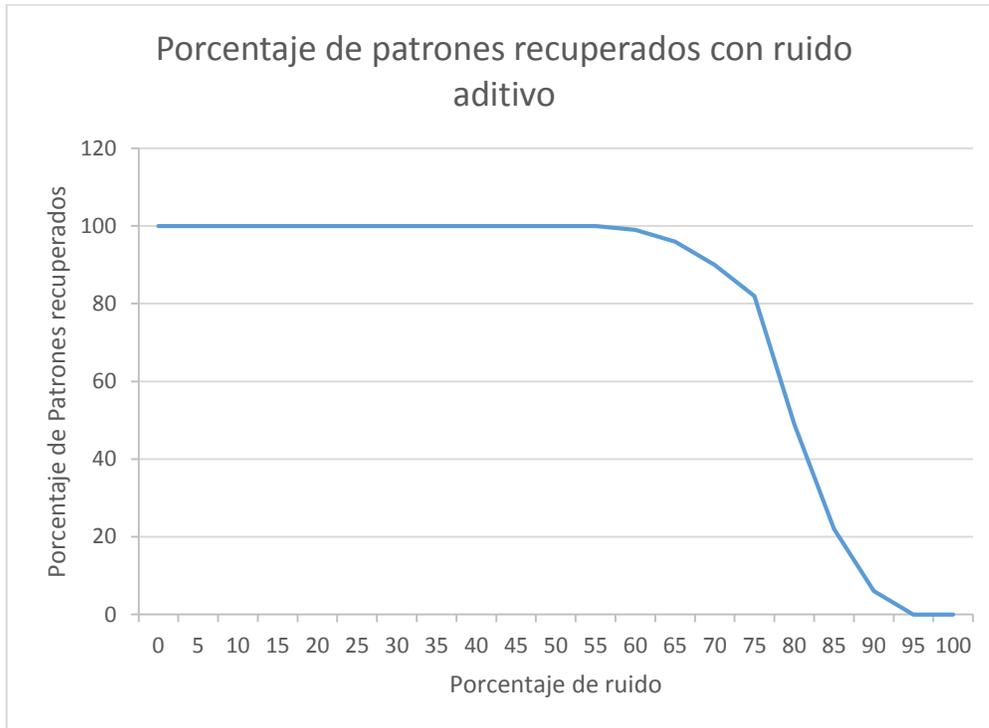


FIGURA 5.6 Porcentaje de patrones recuperados con ruido aditivo

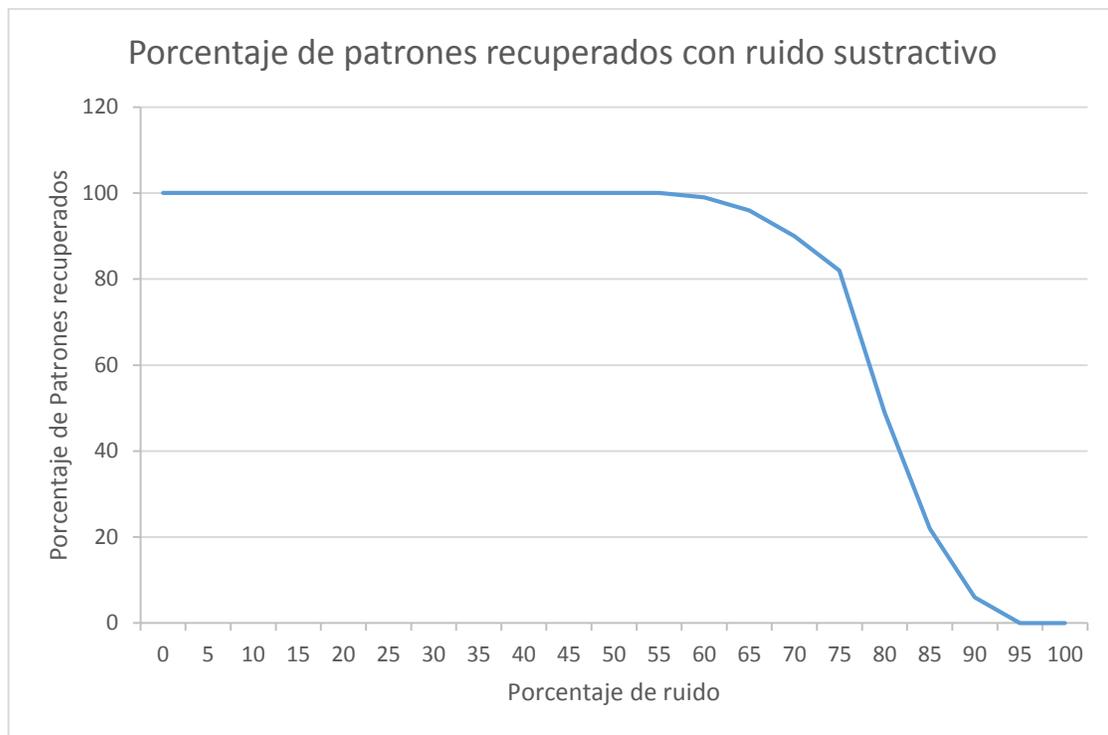
A continuación se presenta los casos en los cuales se recuperan los patrones con un porcentaje de ruido Sustractivo y con qué memoria recupera el patrón.

TABLA 5.21 Porcentaje de recuperación de patrones con ruido Sustractivo.

Porcentaje de ruido Sustractivo	Porcentaje de Patrones recuperados	Se recupera con memoria Min(M)	Se recupera con memoria Max(W)
0	100	Si	Si
5	100	Si	No
10	100	Si	No
15	100	Si	No
20	100	Si	No
25	100	Si	No
30	100	Si	No
35	100	Si	No
40	100	Si	No
45	100	Si	No



50	100	Si	No
55	100	Si	No
60	99	Si	No
65	96	Si	No
70	90	Si	No
75	82	Si	No
80	49	Si	No
85	22	Si	No
90	6	Si	No
95	0	No	No
100	0	No	No



**FIGURA 5.7** Porcentaje de patrones recuperados con ruido sustractivo





## CONCLUSIONES

Al aplicar las memorias asociativas morfológicas para el reconocimiento de patrones de ADN, se encontró que se reconoce el 100% de los patrones si no se aplica ningún tipo de ruido. En las pruebas que se realizaron en este trabajo se logró un porcentaje de acierto, el cual fue de 59% de recuperación de todos los patrones utilizando ruido aditivo o sustractivo, del 60% en adelante empieza a descender el número de patrones que se recupera.

Cuando el ruido es sólo aditivo, se pueden usar, con buen rendimiento, las memorias asociativas morfológicas Max (M), y si el ruido es puramente sustractivo, se pueden utilizar las memorias asociativas morfológicas Min (W), asegurando un buen rendimiento.

Como se mencionó en el capítulo dos, en [31] se pudo observar que aproximadamente el 54% de los patrones fueron biológicamente identificados; mientras que el 46% no pudo ser clasificado dentro del conjunto de repeticiones utilizado, no se permite inserción, modificación, o eliminación de caracteres para la búsqueda de los patrones, esto es sin ruido en las muestras.

En [URL4] se pudo constatar que con BLAST se recuperan el 100% de los patrones, sin ninguna clase de ruido, y lo hace en un tiempo promedio de 30 segundos, pero no soporta ningún tipo de ruido, y por ende, con cualquier clase de ruido no recupera nada.

En [4] se recuperan todos los patrones, pero no acepta ningún tipo de ruido, y lo hace en un tiempo promedio, con la misma base de datos que utilizamos, de 10 minutos.

Comparando UCSC, una interfaz muy popular para buscar ADN y proteínas en bancos de datos, [URL5] para buscar en Genbank, la mayor base de datos de ADN en USA, [URL6] una cadena de 60 caracteres (tamaño pequeño en esa aplicación) y tardó 30 minutos usando una copia local de la base de datos para determinar si existía dicha cadena, se repitió luego el test usando memorias asociativas morfológicas y la búsqueda tardó menos de 1 minuto la mayor parte del tiempo fue debida al movimiento de datos del disco a la memoria, pues realmente la búsqueda se hizo en menos tiempo.

Teniendo en cuenta estos datos se observa que el método aplicado tiene buen rendimiento y un buen porcentaje de recuperación aun con ruido aditivo o sustractivo a diferencia de los métodos anteriormente mencionados.





## GLOSARIO

1. Acanalar. Hacer un canal o una estría en algún lugar u objeto. Dar a una cosa forma de canal o de teja larga y abarquillada.
2. ADN Mitocondrial. Se trata de que el ácido desoxirribonucleico que compone la pared de nuestras mitocondrias está compuesto por una cadena doble de DNA, cuya secuencia, la heredamos del cromosoma "X", es decir, de la madre.
3. ADN Nuclear. El genoma es el conjunto de genes contenidos en los cromosomas, lo que puede interpretarse como la totalidad de la información genética que posee un organismo o una especie en particular. El genoma en los seres eucarióticos comprende el ADN contenido en el núcleo, organizado en cromosomas, y el genoma de orgánulos celulares como las mitocondrias y los plastos; en los seres procarióticos comprende el ADN de su nucleóide.
4. Dilatación. Acción o efecto de dilatar (hacer que algo ocupe más espacio del que ocupa.)
5. Erosión. Desgaste de algo.
6. Estocástico. Proceso que está sometido al azar y que es objeto de análisis estadístico.
7. Fenotipo. Es la expresión del genotipo en función de un determinado ambiente. Los rasgos fenotípicos cuentan con rasgos tanto físicos como conductuales. Un fenotipo es cualquier característica o rasgo observable de un organismo, como su morfología, desarrollo, propiedades bioquímicas, fisiología y comportamiento. La diferencia entre genotipo y fenotipo es que el genotipo se puede distinguir observando el ADN y el fenotipo puede conocerse por medio de la observación de la apariencia externa de un organismo.
8. Isomorfismo. Igualdad de forma.
9. Linaje. Línea de antepasados y descendientes de una persona.
10. Morfológico. De la morfología o relacionado con ella.





## REFERENCIAS

- [1] CORTIJO BON, Francisco José. “Introducción al reconocimiento de formas” y “Técnicas supervisadas II: Aproximación no paramétrica al Reconocimiento de Formas.” En: Reconocimiento de Formas Granada, España: Universidad de Granada, 30 de octubre de 2001.
- [2] DUDA, Richard O.; Meter E. Hart; David G. Stork. Pattern Classification. Second Edition. A Wiley- Interscience Publication, United Sates of América, ISBN 0-47105669-3, 2001.
- [3] GUERRA HERNÁNDEZ, Alejandro. “Árboles de decisión” En: Aprendizaje automático, Veracruz, México: Universidad Veracruzana, 12 de mayo de 2004.
- [4] CAMPOS Javier; Introducción a los algoritmos de biología computacional
- [5] HUCKVALE, Mark “A syntactic pattern recognition method for the automatic Location of potential enhancement regions in running speech”. UCL Working Papers Speech Hearing and Language, 1997.
- [6] VENGUEROV, Mark. “Generalised Syntactic Pattern Recognition as a Unifying Approach in Image Analysis”. 2nd International workshop on Statistical Techniques in Pattern Recognition, Sydney, Australia. August 1998.
- [7] MARTÍ, Eric. “Análisis de Elementos Gráficos en Documentos”. [el línea] Revista electrónica de visión por computador. Octubre 1999-0.
- [8] BELLONE, T.; E. Borgogno; G. Comoglio. “Comparison of Parsing Techniques for The Syntactic Pattern Recognition of Simple Shapes”. International Society for Photogrammetry and Remote Sensing (ISPRS). July 2004, Istanbul, Turkey, Vol. 12, p. 683 ff.
- [9] ACOSTA, Maria Isabel; Salazar Harold; Camilo Zuluaga M. Tutorial de redes neuronales. Colombia: Universidad de Pereira, 2000.
- [10] HERNÁNDEZ MORA, José Juan. ”Introducción a la robótica” En: Tutorial de Técnicas Modernas I. Tlaxcala, México: Instituto Tecnológico de Apizaco, 22 de julio de 2004.



- [11] KOHONEN, T. Self-Organizing Maps. Third Edition. Springer-Verlag Berlin Heidelberg New York, ISSN 0720-678X, ISBN 3-540-67921-9, 2001.
- [12] VILLASEÑOR, Elio Atenógenes G. “Análisis inteligente de datos con redes neuronales artificiales”. Tesis de licenciatura inédita. Universidad Nacional Autónoma de México (UNAM), Agosto de 2004
- [13] GUTIÉRREZ CÁCERES, Juan Carlos. Inteligencia Artificial. Brasil: Universidad de Sao Paulo Brasil, 2003.
- [14] GÓMEZ MARTÍNEZ, Mario. Redes Neuronales Artificiales. Barcelona, España: IIIA – Instituto de Investigación en Inteligencia Artificial y CSIC - Spanish Scientific Research Council, marzo de 2004.
- [15] KOHONEN, T. Self-Organizing Maps. Third Edition. Springer-Verlag Berlin Heidelberg New York, ISSN 0720-678X, ISBN 3-540-67921-9, 2001.
- [16] ACOSTA, Maria Isabel; Salazar Harold; Camilo Zuluaga M. Tutorial de redes neuronales. Colombia: Universidad de Pereira, 2000.
- [17] HERNÁNDEZ MORA, José Juan. ”Introducción a la robótica” En: Tutorial de Técnicas Modernas I. Tlaxcala, México: Instituto Tecnológico de Apizaco, 22 de julio de 2004.
- [18] HOPFIELD, J. J. “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”. Proceedings of the National Academy of Sciences of the USA, Vol. 79, pp. 2554-2558, 1982.
- [19] GUERRA HERNÁNDEZ, Alejandro. “Árboles de decisión” En: Aprendizaje automático. Veracruz, México: Universidad Veracruzana, 12 de mayo de 2004.
- [20] CRUZ LÓPEZ, Maria Luisa. “Implementación y análisis comparativo de una red neuronal ART2 y una Backpropagation, aplicadas al reconocimiento de patrones en imágenes digitales”. Tesis de ingeniería. Universidad Tecnológica de la Mixteca. 2001.
- [21] ARMINGOL, J. M.; et al. “Visión por Computador para Vehículos Inteligentes”. Proyecto inédito del Grupo de Sistemas Inteligentes. Escuela Politécnica Superior de Madrid. 2003.



- [22] DE LA ESCALERA, A.; J. M. Armigol; M. Mata. “Traffic Sign Recognition and Analysis for Intelligent Vehicles”. *Image and Vision Computing*, 2003, vol. 11, No. 3, p. 247-258.
- [23] GODOY, Salvador C. “Evaluación de algoritmos de clasificación basada en el modelo estructural de cubrimientos”. Tesis doctoral. Instituto Politécnico Nacional, Centro de Investigaciones en Cómputo. Mayo de 2006.
- [24] RODRIGUEZ DIEZ, Héctor; et al. *Sistema para la Clasificación basado en Técnicas de Reconocimiento de Patrones*. Cuba: Santiago de Cuba. Universidad de Oriente. 2000.
- [25] DÍAZ, G. S.; y J. R. Shulcloper, "Mid Mining: A Logical Combinatorial Pattern Recognition Approach to Clustering in Large Data Sets" *Memorias del V Simposio Iberoamericano de Reconocimiento de Patrones*, Lisbon, Portugal, September 2000 pp. 475-483.
- [26] YANKOVSKAYA, A. E. “Logical-Combinatorial-Probabilistic Recognition Algorithms”. *Pattern Recognition and Image Analysis*, 2001, Vol. 11, No. 1, pp. 123–126.
- [27] Una Nueva Red Neuronal para Clustering y Segmentación basada en el entorno. Lanzarini, De Giusti. VI Congreso Argentino de Ciencias de la Computación. Ushuaia, Argentina, Octubre 2000. Vol. 3, Pags. 1389-1398. ISBN: 950-763-033-3
- [28] Parallelizing a new neural network for environment based clustering. Lanzarini, De Giusti. 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and 7th International Conference on Information Systems Análisis and Síntesis (ISAS 2001); Orlando, USA, Julio 2001.
- [29] United Nations. Office of Drugs and Crimes. *Toolkit to Combat Trafficking in Persons*, 2009. 7.
- [30] UNODD, UN. *GIFT - Global Initiative to Fight Human Trafficking*, 2009.
- [31] Marcos A. Foglino; *Identificación biológica de secuencias de ADN del genoma humano*. Agosto 2009
- [32] Hassoun, M. H. 1993, *Associative Neural Memories*, Oxford University Press, New York.
- [33] Kohonen, T. 1989, *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.



- [34] Yáñez, C. 2002, *Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios*, Tesis de Doctorado, Centro de Investigación en Computación, México.
- [35] Simpson, P. K. 1990, *Artificial Neural Systems*, Pergamon Press , New York.
- [36] Minsky, M. & Papert, S. 1969, *Perceptrons*, MIT Press ,Cambridge.
- [37] Steinbuch, K. & Frank, H. 1961, “Nichtdigitale Lernmatrizen als Perzeptoren”, *Kybernetik*, vol. 1, no.3, pp. 117-124.
- [38] Kohonen, T. 1972, “Correlation matrix memories”, *IEEE Transactions on Computers*, C-21, vol. 4, pp. 353-359.
- [39] Papadomanolakis, K., Kakarountas, A., Sklavos, N. & Goutis C. E., 2002, A Fast Johnson-Mobius Encoding Scheme for Fault Secure Binary Counters, *Proceedings of Design, Automation and Test in Europe*, (DATE '02), pp. 1-7.
- [40] Willshaw, D., Buneman, O. & Longuet-Higgins, H. 1969, “Non-holographic associative memory”, *Nature*, no. 222, pp. 960-962.
- [41] Anderson, J. A. 1972, “A simple neural network generating an interactive memory”, *Mathematical Biosciences*, vol. 14, pp. 197-220.
- [42] Hopfield, J.J. 1982, “Neural networks and physical systems with emergent collective computational abilities”, *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554-2558.
- [43] Abu-Mostafa, Y. & St. Jacques, J. 1985, “Information capacity of the Hopfield model”, *IEEE Transactions on Information Theory*, IT-31, no. 4, pp. 461-464.
- [44] Sossa, H., Barrón, R. & Vázquez, R. 2004, “New Associative Memories to Recall Real-Valued Patterns”, *CIARP*, LNCS 3287, pp. 195-202.



**[URL]**

[URL1] CARRASCO OCHOA, Jesús Ariel; Martínez Trinidad, José Francisco. Reconocimiento de Patrones. [en línea] Puebla, México: Instituto Nacional de Astrofísica Óptica y Electrónica. <<http://ccc.inaoep.mx/~ariel/>> [Consulta: 13 de septiembre de 2006].

[URL2] MATICH, Damián Jorge. “Redes neuronales: conceptos básicos y aplicaciones”. En: Informática aplicada a la ingeniería de procesos I. [en línea] Universidad Tecnológica Nacional, Facultad Regional Rosario. Marzo de 2001. <<http://www.modeladoingenieria.edu.ar/>> [Consulta: 10 Enero de 2015]

[URL3] PROGRAMA PROKIDS <<http://www.dna-prokids.org/>> [Consulta:13 de Enero de 2015]

[URL4][http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE\\_TYPE=BlastSearch&BLAST\\_SPEC=OGP\\_9606\\_9558&LINK\\_LOC=blasthome](http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch&BLAST_SPEC=OGP_9606_9558&LINK_LOC=blasthome)

[URL5] Programa UCSC<<http://genome.ucsc.edu/>>[Consulta:15 de Mayo de 2015]

[URL6] Base de Datos <<http://www.ncbi.nlm.nih.gov/Genbank/>>[Consulta:15 de Mayo de 2015]