VNIVERSIDAD NACIONAL AVPNOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"MAPEOS AUTOORGANIZADOS PARA LA VISUALIZACIÓN DE LA DISTRIBUCIÓN DE ORGANISMOS CON BASE EN SU USO DE CODONES"

T E S I S

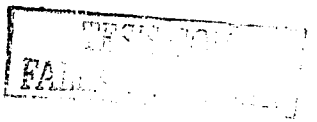
QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A:

JOSÉ ANTONIO NEME CASTILLO

DIRECTOR DE TESIS: DR. PEDRO MIRAMONTES



MÉXICO, D.F. **2003.**

"...En aquel Imperio, el Arte de la Cartografía logró tal Perfección que el mapa de una sola Provincia ocupaba toda una Ciudad, y el mapa del Imperio, toda una provincia. Con el tiempo, esos Mapas Desmesurados no satisficieron y los Colegios de Cartógrafos levantaron un Mapa del Imperio, que tenía el tamaño del Imperio y coincidía puntualmente con él. Menos Adictas al Estudio de la Cartografía, las Generaciones Siguientes entendieron que ese dilatado Mapa era Inútil y no sin Impiedad lo entregaron a las Inclemencias del Sol y de los Inviernos. En los desiertos del Oeste perduran despedazadas Ruinas del Mapa, habitadas por Animales y por Mendigos; en todo el País no hay otra reliquia de las Disciplinas Geográficas."

-Jorge Luis Borges. 'Del rigor de la ciencia,' en *El Hacedor*.

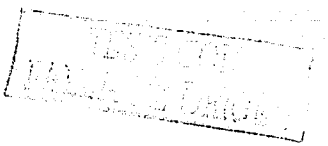
Índice General

Capítulo 1

Introducción

1.1 Descripción del problema

Los humanos, al igual que muchos otros animales, interactuan en un ambiente donde existen muchas variables físicas, que son registradas por los órganos correspondientes. El sistema nervioso en general y la corteza cerebral en particular (bidimensional para fines prácticos), son un ejemplo de la utilidad de representar en baja dimensión (en la corteza cerebral, por ejemplo) lo que ocurre en un ambiente de alta dimensión [80]. Podemos evadir obstáculos, calcular trayectorias, anticipar movimientos, apreciar objetos artísticos, etcétera [4]. Somos, al menos intuitivamente, buenos manipuladores y analizadores de objetos inmersos en espacios de alta dimensionalidad precisamente porque somos capaces de *mapear* lo que ocurre en dicho espacio a nuestro cerebro [82].
Algunos de los fenómenos interesantes para los científicos de las diversas ramas del conocimiento son descritos por pocas variables, dos, tal vez tres, lo que les permite ser representados sin demasiados problemas en la terminal gráfica de una computadora [65]. Incluso, objetos descritos por tres variables, esto es, tridimensionales, pueden ser representados adecuadamente en la misma terminal gráfica sin demasiados problemas [80].
Las razones para querer observar objetos multidimensionales en el plano son varias: encontrar, ya sea gráfica o analíticamente, la distribución de dichos objetos en el espacio que ocupan; encontrar las relaciones, posiblemente de distancia o de *vecindad* (aquellos objetos cercanos entre sí son objetos vecinos), entre los objetos analizados es otra de las razones [70].
Existen otros fenómenos, la mayoría, en los que las variables que intervienen, y con esto, la dimensión del problema, son mas de tres. Ahora, la representación en un espacio bidimensional, como el monitor de la computadora, de



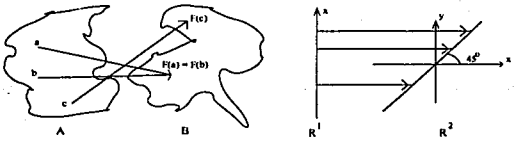


Figura 1.1: Algunos ejemplos de mapeos.

objetos con dimensión mayor a tres, ya no resulta tan sencilla. Para poder visualizar dichos objetos en un espacio de dimensión más baja es necesario un *mapeo*.
Un mapeo es una función que va de cierto conjunto A a otro conjunto B de tal forma que cada elemento a contenido en A está asociado con un y sólo un elemento b contenido en B [5, 98]. La figura 1.1 muestra algunos ejemplos de mapeos. En lo sucesivo nos referiremos a la aplicación de un mapeo como *mapear*.

Una proyección sobre un plano, a diferencia de un mapeo, no preserva la topología. En una proyección sobre el plano se descartan todas las variables, salvo dos, que son las que definen el plano de proyección y cada objeto multidimensional será ahora representado por un punto en ese plano [5].
Veamos el siguiente problema, extraído de la teoría del reconocimiento de patrones. Tenemos dos clases de objetos, naranjas y limones. Queremos diseñar un sistema que sea capaz de decidir si un objeto dado es o una naranja o un limón. Para ello, contamos con dos características de los objetos que nos serán de utilidad: el diámetro y el color. El diámetro puede ser expresado en centímetros y el color en el intervalo [0, 1], con un valor cercano a cero si el color es azul y cercano a 1 si el color es rojo [90].
Los puntos en este problema se representan en el espacio de dimensión dos, pues existen únicamente dos características. Si muestreamos un conjunto de objetos e indicamos en este espacio, de dimensión dos, el punto que representa a cada objeto, obtendremos una distribución similar a la mostrada en la figura 1.2.

Los puntos que conforman al primer grupo, con un radio aproximado de 3 cm y un color cercano a .25 (verde) son los limones, en tanto que el segundo grupo, con radio aproximado de 4.5cm, y con color aproximado de 0.7 (anaranjado) son las naranjas. En esta representación gráfica puede apreciarse la distribución de los objetos en el espacio que ocupan y se puede concluir

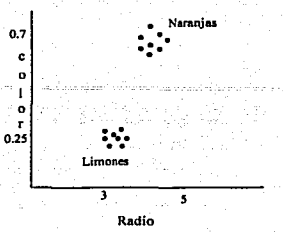


Figura 1.2: Objetos en el espacio de características de bidimensional.

que, en general, naranjas y limones son fácilmente distinguibles, pues podría trazarse una línea o separatriz que divida el espacio en dos regiones, una para los limones y otra para las naranjas [31].
¿Qué ocurre si queremos diferenciar naranjas y toronjas? El color de ambas es mas o menos semejante, lo mismo que el radio. Si sólo conservásemos las dos características mencionadas, la diferenciación se convertiría en un proceso inexacto, pues la separatriz no podría ser trazada. Tenemos que recurrir a alguna otra característica que pudiera, en conjunto con las dos previamente mencionadas, diferenciar a ambas.
Se podría recurrir al nivel de acidez, pero incluso así, podría no ser suficiente. Si a estas tres variables añadimos otras más, tales como la textura de la superficie, y el color interno, tendremos entonces el conjunto de características necesarias para diferenciarlas [6]. Cada objeto estará, entonces, representado por un vector de cinco características. ¿Cómo analizamos la distribución de dichos objetos en el espacio de dimensión 5?
Imaginemos que el problema original es modificado para reconocer no únicamente naranjas, toronjas y limones, sino un conjunto mayor de frutas, usando, para todas ellas, las mismas 5 características. Quisiéramos saber qué frutas se encuentran, en el *espacio de características* de dimensión 5, cercanas entre sí, y alejadas de otras. Dicho de otra forma, quisiéramos saber la topología existente en el espacio referido (el de características).
Para tener una idea, al menos gráfica, de esta topología, es que necesitamos representar a los objetos de dimensión 5. Visualizarlos en el espacio de dimensión cinco es imposible, por lo que recurrimos a un mapeo, que va de ese espacio al de dimensión dos. Recordemos, por la definición de mapeo, que es necesario asignarle a cada uno de los objetos de dimensión cinco un objeto en dimensión dos. Para hacer esto, podríamos definir el siguiente mapeo:

$$x = \alpha_0 + \alpha_1\alpha_2$$
$$y = \alpha_3 * cos(\alpha_4 - \alpha_1)$$

Este mapeo asigna, para cada objeto j descrito por el vector de características $(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$, un y solo un punto en el espacio bidimensional, (x, y). Desgraciadamente, este mapeo no es de gran utilidad, pues no *preserva la topología* [56]. Por lo pronto, aunque en el capítulo tres profundizaremos en este concepto, veremos la preservación de la topología como la conservación de vecindades [3].

Lo que queremos de un mapeo es que muestre como cercanos en el espacio de baja dimensión a aquellos objetos que en realidad están cerca en el espacio de la dimensión original. Para ello, en el presente trabajo recurrimos a un mapeo especial, denominado *mapeo autoorganizado* [100]. El mapeo autoorganizado es una red neuronal artificial no supervisada (carece de un instructor que le diga si lo que hace está bien o mal) [61] que mapea objetos de dimensión generalmente mayor a tres, a una malla discreta de nodos, de dimensión generalmente dos, pero que cuenta con la ventaja de preservar la topología [61]. El tercer capítulo ahonda en las definiciones y características de dicho mapeo.

Una vez definido, aunque superficialmente, el concepto de mapeo autoorganizado, conviene hablar de la segunda frase que describe el título del presente trabajo, el *uso de codones.*

Un codón o triplete es una secuencia de tres nucleótidos (adenina, guanina, citosina y timina) que codifica un aminoácido. Existen 64 tripletes que dan lugar a 20 aminoácidos diferentes.

El problema del sesgo en el uso de codones proviene de la biología molecular y se refiere a como un mismo *aminoácido*, la unidad básica de construcción de proteínas, puede ser codificado mediante sinónimos por diversos organismos [68]. Diferentes teorías se han propuesto para explicar este fenómeno, sin embargo, continua siendo un problema abierto pues aún no se cuenta con una explicación razonable que cubra todos los casos [48].

En el problema del sesgo en el uso de codones, el espacio de características es de dimensión 64 pues, como se explica en el capítulo dos, donde se presenta el uso de codones, con el esquema de codificación establecido en el *código genético*, se podría representar un total de 64 instancias [7], puesto que se tienen 64 secuencias codificadoras o *tripletes.* Cada uno de los codones es una dimensión en el espacio de características. De esta forma, los objetos analziados (oranismos) son puntos en el espacio de dimensión 64.

Organismos que están relacionados filogenéticamente no necesariamente hacen un uso de codones semejante, como se explica con detalle en el capítulo

dos [86]. Una visualización que muestre organismos que hacen un uso de codones semejantes como entidades cercanas, en tanto que muestre como entidades lejanas aquellos organismos que hacen un uso de codones diferente, sería una herramienta adecuada. En este trabajo se muestra que el mapeo autoorganizado es una herramienta que cumple adecuadamente con lo anterior [93].
Cada especie en este trabajo es representado por un vector de dimensión 64. La topología en este espacio la intentamos preservar precisamente con un mapeo autoorganizado. Al visualizar los objetos, en este caso organismos, en el espacio de dimensión dos (la malla del mapeo), nos podemos dar una idea aproximada de la topología que en realidad existe en el espacio orginial. Al igual que con el problema de las frutas, en donde naranjas, limones y toronjas se ubicaban en una misma *vecindad*, y el melón y la sandía en otra vecindad, alejada de la primera, en el caso del uso de codones que presentan diversos organismos pertenecientes a los diversos reinos taxonómicos, queremos saber cuales de ellos estan cerca entre si, y cuales alejados. Queremos identificar las vecindades que se forman en el mapeo, pues estas pueden ser de utilidad para los especialistas.

1.2 Antecedentes

El sesgo en el uso de codones ha sido estudiado mediante diversas técnicas. Con herramientas de análisis multivariado, Nesti ha analizado y encontrado grupos de organismos que hacen un uso de codones semejante [74]. Los métodos empleados, tales como análisis de componentes principales, han mostrado cierta utilidad en el análisis de codones entre organismos de la misma especie, pero si se analizan organismos incluso de unas pocas especies diferentes , el poder de resolución del método se degrada considerablemente [48].
Wang y Nanaya han analizado el sesgo en el uso de codones para grupos de genes, incluso utilizando para ello el mapeo autoorganizado [97, 48]. Nikkila y Kaski han aplicado el mapeo autoorganizado para el análisis de la expresividad de genes en un mismo organismo [75, 49].
Adicionalmente, en la mayoría de los trabajos se han analizado genes de organismos del mismo superreino, como bacterias [97], en donde se intenta, principalmente, encontrar aquellos genes que fueron incorporados al genoma de la bacteria provenientes de otro organismo [48], o agrupar a los genes por su funcionalidad [49].
En todos los casos, se ha utilizado para el análisis el total de características disponibles, ésto es, no se ha intentado encontrar aquellas que presenten

mayor poder explicativo para el mapeo autoorganizado que forman los organismos o genes estudiados. En el presente trabajo, se realizan esfuerzos por encontrar dichas características.
En cuanto al uso de mapeos autoorganizados como herramientas de visualización, el trabajo es extenso [61, 8, 29]. En lo referente a la eliminación de variables o características para el mapeo autoorganizado, existen diversos trabajos [57, 50], que contrastamos en el capítulo cinco con una metodología propuesta por nosotros, consistente en encontrar dicho subconjunto de características por medio de un algoritmo genético.

1.3 Propuesta

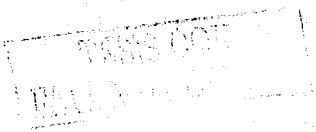
Uno de los objetivos del presente trabajo es utilizar un mapeo autoorganizado para obtener, en el espacio de dimensión dos, la distribución de los organismos con base en su uso de codones, y con ésto, observar la topología que presentan dichos organismos en el espacio de dimensión 64. El conocer las vecindades en el espacio del mapeo, podrá dar elementos a los biólogos moleculares para tratar de dilucidar posibles escenarios evolutivos [72].
Otro objetivo del presente trabajo es encontrar el mejor mapeo autoorganizado, de tal forma que la topología existente en el espacio de dimensión 64 sea preservada lo mejor posible. Como se verá en el capítulo cuatro, siempre que en un mapeo la dimensión se reduce, la topología se viola. Dos mapeos autoorganizados aplicados a un mismo conjunto de datos proporcionarán resultados semejantes, pero no idénticos [56]. Cuantificar la violación de la topología es escencial a fin de identificar aquel o aquellos mapeos que sean los que mejor preserven la topología. En ese capítulo, serán analizadas diversas métricas para evaluar la *bondad* o *desempeño* de un mapeo.
Un tercer objetivo del presente trabajo es identificar cuales de las 64 características son en realidad necesarias para el mapeo. Para explicar mejor este proceso de identificación, regresemos por un momento al ejemplo de las frutas. Supongamos que una sexta variable, la región donde fueron cosechados los frutos, es tomada en cuenta. El mapeo será entonces de la dimensión seis a la dimensión dos.
Puesto que muchas frutas son cultivadas en la misma región (la naranja, la toronja, el melón, el limón, la sandía, la papaya, el mango, etcétera, son cultivados en una región de poco mas de 400 km^2 en la zona centro norte del estado de Veracruz [45]), esta nueva característica en realidad no aporta ningúna información adicional. En otras palabras, esta característica puede ser descartada por lo que el conjunto de 5 características originales es suficiente para describir a las frutas.

Una característica que no es necesaria para describir un objeto o fenómeno puede ser eliminada. Dicho de otra forma, se puede eliminar siempre y cuando no viole la topología que existe en el espacio original (las relaciones de vecindad), o lo viole lo menos posible. El capítulo cinco ahonda en las diversas técnicas existentes para eliminar características, ademas de mostrar una metodología propuesta por nosotros.
Las técnicas referidas en dicho capítulo seleccionan aquellos rasgos que hacen que el mapeo formado por ellas presente el error mas bajo; esto es, que la preservación de la topología sea máxima, con lo que puede afirmarse que los rasgos no seleccionados no solo no cuentan con poder explicativo del fenómeno, sino que además, podrían intervenir de manera negativa en el entendimiento del problema.
El capítulo seis analiza otros mapeos que también preservan la topología, profundizando en aquellos poco estudiados en la bibliografía, como los obtenidos por medio de programación genética, el cual muestra errores bajos que lo hacen una alternativa al mapeo autoorganizado.
El capítulo siete es dedicado a las conclusiones, tanto desde el punto de vista computacional como a las posibles implicaciones de los resultados en el área de la biología computacional.

1.4 Discusión

Visualizar objetos en un espacio de dimensión mayor a tres requiere de un mapeo que preserve la topología. Un mapeo autoorganizado cumple con el requisito de conservar la topología presente en el espacio original de características, definida por los objetos analizados. En el presente trabajo, los objetos son especies biológicas y las variables que los describen son sus usos de codones.
El uso de codones es la frecuencia que cada organismo presenta para cada uno de los 64 tripletes o codones, que codifican los veinte aminoácidos existentes en la naturaleza. Al utilizar un mapeo autoorganizado para estudiar diversos organismos, se pretende establecer la relación de vecindad entre cada organismo y los restantes.
Siempre que existe un mapeo en donde se reduce la dimensión, la topología se viola. Cuantificar esta pérdida es necesario, a fin de identificar aquel mapeo que sea el que mejor preserve la topología. No todas las variables pueden ser necesarias para obtener un buen mapeo. Aquellas que al ser removidas preserven la topología lo mejor posible, serán de mucho mayor interés no sólo computacional, sino también para los especialistas, en este caso, en el uso de codones.

Aunque el mapeo autoorganizado es una de las mejores herramientas para visualizar objetos con una dimensionalidad alta, existen otras que pudieran incluso presentar un error menor al del mapeo autoorganizado. Tal es el caso de los mapeos obtenidos con programación genética.



Capítulo 2

Uso de codones

2.1 Aminoácidos y proteínas

Las proteínas son los componentes fundamentales de las células. Son las encargadas de las diversas funciones vitales para el organismo, desde catalizadoras en reacciones químicas, hasta propulsores para organelos, que son componentes de la célula.
Existen proteínas que actúan como enzimas catalizadoras en diversas reacciones químicas, como la *pepsina*, que se encuentra en el estómago de los mamíferos, la cual permite degradar los alimentos, o la enzima *ribulosa bifosfato carboxiasa* que ayuda a la plantas en la conversión de dióxido de carbono en azúcar [9].
Otras proteínas son de tipo estructural, como la *α-keratina*, que se encuentra en abundancia en el pelo, o el *colágeno*, que se encuentra en las células de la piel. Ambas proteínas, al igual que todas las de tipo estructural, proveen soporte mecánico a las células y tejidos [2].
Otro tipo de proteína es la que se encarga del transporte de moléculas dentro del organismo. La *hemoglobina* lleva oxígeno a los diversos órganos y tejidos, en tanto que la *transfernina* se encarga de transportar hierro.
Un cuarto tipo lo constituyen las proteínas motrices, como la *kinesina*, considerada una auténtica máquina molecular, que impulsa a los organelos a través del citoplasma, que es la región de la célula que rodea al núcleo y donde se encuentran diversas estructuras que componen la célula [10].
Las proteínas de señalización se encargan del envío de señales entre células, como el *factor de crecimiento neuronal*, que estimula a las neuronas para que desarrollen axones y puedan de esta forma conectarse con otras neuronas.
Un sexto tipo son las proteínas de regulación genética, encargadas de acoplarse a la molécula del ADN con el objetivo de permitir que un gen sea

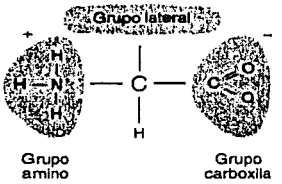


Figura 2.1: Estructura general de los aminoácidos. La estructura común incluye un átomo central de carbono, un grupo amino y un grupo carboxil. Lo que los distingue es el grupo lateral

expresado o no en el fenotipo.
En la molécula conocida como ácido desoxiribonucleico, o ADN, se encuentra codificada la información suficiente para sintetizar las proteínas necesarias para la vida [2][10].
A pesar de la variedad de funciones que las proteínas realizan, las unidades que las constituyen son únicamente veinte. Lo que diferencia una proteína de otra es el conjunto de esas unidades básicas de construcción que las forman y la secuencia en la que aparecen. Las unidades básicas de construcción son los *aminoácidos* [44].
Los aminoácidos son monómeros que se unen para formar proteínas, tienen una estructura similar entre ellos y solo varían en uno de sus componentes, como se muesta en la figura 2.1 [68]. Un monómero es una molécula orgánica relativamente pequeña que se puede unir a otros monómeros para dar lugar a un polímero, como las proteínas [2].

La estructura general de los aminoácidos es la misma: constan de un grupo amino (NH_3), un grupo carboxil ($COOH$) y un átomo central de carbono, variando solamente en el grupo lateral. En la tabla 2.1 se muestra el nombre de los 20 aminoácidos, así como su abreviatura, en tanto que en la figura 2.2 se muestra la estructura de los veinte aminoácidos.

Las proteínas que están presentes en un organismo se determinan especificando los aminoácidos que las forman. Esos aminoácidos son conocidos como la *secuencia de aminoácidos*. Las proteínas puede estar formadas por una secuencia de entre 30 y 10000 aminoácidos [2]. De esta forma, la proteína *aminoreceptora III*, relacionada con la *rodpsina*, proteína encargada de detectar la luz en el ojo de los vertebrados, está formada por 348 aminoácidos, mostrados en la tabla 2.2. Cada letra corresponde a uno de los 20 aminoácidos (ver tabla 2.1). Para fines de facilidad en la interpretación, se listan en columnas de diez [25].

Glicina	Gly	G	Triptofano	Trp	W
Alanina	Ala	A	Histidina	His	H
Valina	Val	V	Arginina	Arg	R
Leucina	Leu	L	Lisina	Lys	K
Isoleucina	Ile	I	Aspartato	Asp	D
Serina	Ser	S	Glutamato	Glu	E
Cisteína	Cys	C	Aspargina	Asn	N
Metionina	Met	M	Glutamina	Gln	Q
Tirosina	Tyr	Y	Prolina	Pro	P
Fenilalanina	Phe	F	Treonina	Tre	T

Tabla 2.1: Nombre de los 20 aminoácidos. Se indica su abreviatura y la letra que lo designa.

mvnnfsqaea	velcyknvnc	sciktpyspg	prsilyavlg	fgavlaafgn	llvmiailhf
kqlhtptnfl	iaslacadfl	vgvtvmpfst	vrsvescwyf	gdsyckfhtc	fdtsfcfasl
fhlccisvdr	yiavtdplty	ptkftvsvsg	icivlswffs	vtysfsifyt	ganeegieel
vvaltcvggc	qaplnqnwvl	lcfllffipn	vamvfiyski	flvakhqark	iestasqaqs
ssesykcrva	krcrkaaktl	giamaaflvs	wlpylvdavi	daymnfitpp	yvyeilvwcv
yynsamnpli	yaffyqwfgk	aiklivsgkv	lrtdssttnl	fsceveted	

Tabla 2.2: Aminoácidos que constituyen la proteína aminoreceptora III. Un aminoácido es representado por una letra (ver tabla 2.1).

La figura 2.3 muestra esquemáticamente como se sintetiza una sustancia en donde una proteína participa como enzima. Se muestra también como dicha proteína fue sintetizada a partir de aminoácidos [7].

2.2 El Código genético

Todos los organismos emplean un alfabeto de cuatro letras para codificar los 20 aminoácidos, que son las unidades básicas de construcción de proteínas. El alfabeto se muestra en la tabla 2.3. Esta es una tabla de equivalencia entre secuencias del alfabeto codificador y los aminoácidos y se le denomina *código genético estándar* [68].

Las cuatro letras del alfabeto de aminoácidos son los *nucleótidos* Adenina(A), Guanina(G), Citosina(C) y Timina(T). Los nucleótidos son las moléculas que forman la cadena de ADN. Así como los aminoácidos son

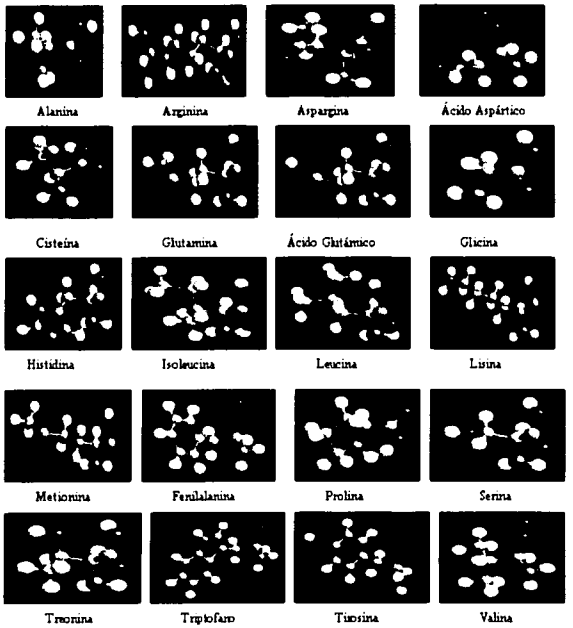


Figura 2.2: La estructura de los 20 aminoácidos. Los átomos de carbono se representan en color claro; Los átomos mas obscuros indican átomos de oxígeno, los de tonalidad grisácea clara son átomos de nitrógeno y el azufre se indica con tono gris mas intenso.

monómeros que dan lugar a los polímeros proteínas, los nucleótidos son monómeros que forman al polímero llamado ADN [10].
Los 20 aminoácidos deben ser especificados de manera que no exista confusión. Esto es, una secuencia particular de nucleótidos codificará para un y sólo un aminoácido. Si la longitud de dicha secuencia fuera 1, se podrían codificar solamente cuatro diferentes aminoácidos, puesto que en esa secuencia solo podría contener uno de los cuatro nucleótidos ($4^1 = 4$). Si la secuencia fuera de longitud dos, el número de aminoácidos que podría codificarse sería 16 ($4^2 = 16$).
Con una secuencia de longitud 3, es posible representar hasta 64 (4^3) instancias, por lo que, puesto que solamente existen 20 aminoácidos, diversas secuencias codificarán el mismo aminoácido. El motivo por el cual una determinada secuencia codifica para un amino ácido en específico y no para

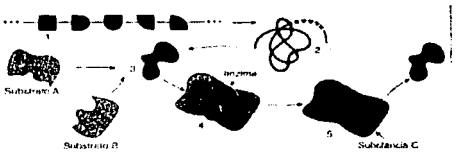


Figura 2.3: Síntesis de una enzima

otro, esto es, la razón por la cual la tabla de equivalencias o código genético presenta esta y no otra conformación, es un problema abierto y se aborda desde diversas teorías sobre el origen del código genético [71].

2.3 Síntesis de proteínas

La molécula de ADN es la encargada de almacenar la información genética. Está formada por dos cadenas complementaria de nucleótidos, que indican las secuencias de aminoácidos que dan lugar a las proteínas La figura 2.4 muestra dicha molécula. Esta molécula está formada por una doble hélice que asemeja una escalera, en donde los *peldaños* de ésta son las uniones entre un nucléotido y su complementario. Si de un lado del ADN aparece una adenina (A), del otro lado aparecerá una timina (T), en tanto que si se encuentra la guanina (G), del otro lado existirá una citosina (C) [2].

En las células de los eucariontes, que tienen núcleo, el ADN se encuentra en él, en tanto que en las células procariotes, que carecen de núcleo, el ADN se encuentra en el citoplasma. EL ADN es *empaquetado* en los *cromosomas* [68]

En el proceso de síntesis de proteínas, dos tareas principales se llevan a cabo de manera secuencial: la *transcripción* y la *traducción*.
La transcripción comienza con una molécula de ARN en donde se copia la secuencia de nucleótidos complementarios a una de las hélices del ADN. En el ARN, la timina es sustituida por el nucleótido uracilo (U). Una vez que se obtiene este ARN, llamado ARN nuclear, se eliminan de él secuencias no codificadoras, llamadas intrones y el ARN abandona el núcleo de la célula, convirtiéndose ahora en ARN mensajero (mARN). La figura 2.6 muestra este proceso.
Una vez que el ARN mensajero (mARN) se encuentra fuera del núcleo, el mARN es transportado al citoplasma, donde se unirá a los ribosomas, que son moléculas compuestas por ARN y proteínas, y que se encargan de leer las secuencias del mARN para sintetizar la proteína codificada. El proceso de

	T	C	A	G	
T	Phe(F)	Ser(S)	Tyr(Y)	Cys(C)	T
T	Phe(F)	Ser(S)	Tyr(Y)	Cys(C)	C
T	Leu(L)	Ser(S)	Ter(.)	Ter(.)	A
T	Leu(L)	Ser(S)	Ter(.)	Trp(W)	G
C	Leu(L)	Pro(P)	His(H)	Arg(R)	T
C	Leu(L)	Pro(P)	His(H)	Arg(R)	C
C	Leu(L)	Pro(P)	Gln(Q)	Arg(R)	A
C	Leu(L)	Pro(P)	Gln(Q)	ARg(R)	G
A	Ile(I)	Thr(T)	Asn(N)	Ser(S)	T
A	Ile(I)	Thr(T)	Asn(N)	Ser(S)	C
A	Ile(I)	Thr(T)	Lys(K)	Arg(R)	A
A	Met(M)	Thr(T)	Lys(K)	Arg(R)	G
G	Val(V)	Ala(A)	Asp(D)	Gly(G)	T
G	Val(V)	Ala(A)	Asp(D)	Gly(G)	C
G	Val(V)	Ala(A)	Glu(E)	Gly(G)	A
G	Val(V)	Ala(A)	Glu(E)	Gly(G)	G

Tabla 2.3: Código genético. El primer nucleótido del triplete es indicado por la primer columna del lado izquierdo, el segundo es indicado por las cuatro columnas centrales y el último nucleótido del codón es indicado por la última columna

traducción depende de la presencia de moléculas que mapeen las secuencias codificadoras de longitud 3 (llamadas *tripletes*) al aminoácido respectivo, llamadas ARN de transferencia (tARN). Una vez que la secuencia que codifica a una proteína es leída y traducida, el polipeptido sale del ribosoma y comienza a *plegarse* hacia su configuración natural, o de menor energía [44][2]. La estructura tridimensional de una proteína está dada por la secuencia de aminoácidos que la conforman. La función de una proteína depende de su estructura, por lo que para predecir su función es necesario conocer su estructura. La estructura tridimensional de las proteínas es variable y puede llegar a ser compleja. La figura 2.5 muestra algunos ejemplos de estructuras de proteínas [9][10].

Anticipar la estructura de una proteína analizando la secuencia principal o *estructura primaria*, es una tarea compleja, por lo que se aborda desde varios enfoques. Uno de tales enfoques es el de minimización de energía, que se basa en las propiedades fisicoquímicas de los aminoácidos, pero que presenta la desventaja de la enorme complejidad de cálculo [68] [12].
Otro enfoque para predecir la estructura tridimensional de las proteínas es el

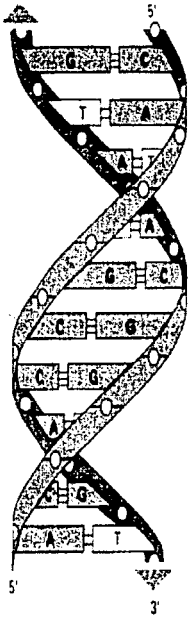


Figura 2.4: La molécula del ADN. Se observa que nucléotidos complementarios aparecen en los *peldaños* que conforman la molécula.

uso de redes neuronales artificiales. Con ellas, se intenta encontar patrones en la estructura de la molécula y aprenderlos para referencias futuras [11]. La predicción de la estructura tridimensional de las proteínas es una de las áreas de mayor actividad dentro del campo de la *bioinformática*, y no solo es abordada por alguno de los dos métodos mencionados, sino por una gran variedad de ellos [87].

2.4 Uso de codones

La secuencia de tres espacios que se asocia con un aminoácido recibe el nombre de triplete o codón, y tal como se observa en la tabla 2.3, un mismo aminoácido puede ser representado por mas de un triplete [68]. Puesto que con un alfabeto de cuatro letras (A, G, C, T) y secuencias de longitud tres se pueden representar hasta 64 instancias (aminoácidos, en este caso), existen tripletes diferentes que codifican a un mismo aminoácido. Estos tripletes son

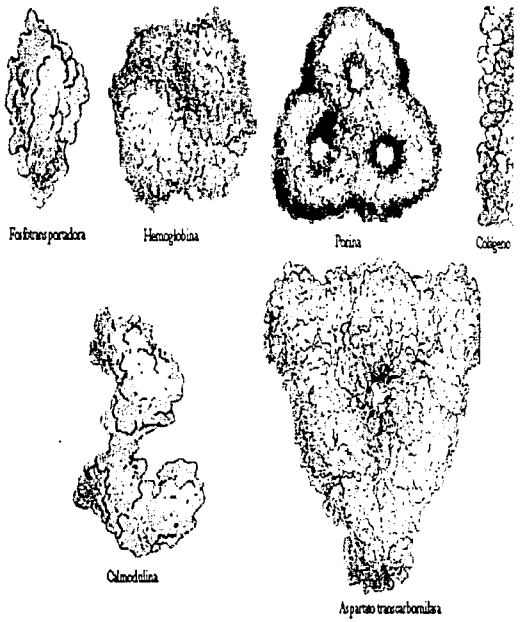


Figura 2.5: La estructura de diversas proteínas.

conocidos como sinónimos.
Tres tripletes son no codificadores: TAA, TAG y TGA indican que la secuencia de aminoácidos que forma una proteína ha terminado(Ter). La figura 2.6 muestra el proceso de traducción del DNA a proteínas.
El uso de sinónimos en el código genético se denomina *uso de codones* [86]. Un mismo organismo podría codificar un aminoácido con cualquiera de los tripletes que se relacionan al aminoácido. Si el uso de codones que hace un organismo fuera aleatorio, la proporción de cada triplete codificador de un aminoácido seria la misma. Por ejemplo, la Valina se relaciona con los tripletes GTT, GTC, GTA y GTG. Si el uso de codones fuese aleatorio, entonces, al analizar la codificación de las proteínas que requieren de la Valina para formarse, se observaría una distribución igual entre cada uno de los cuatro tripletes [78].
En muchos organismos se ha encontrado que los codones para un mismo ami-

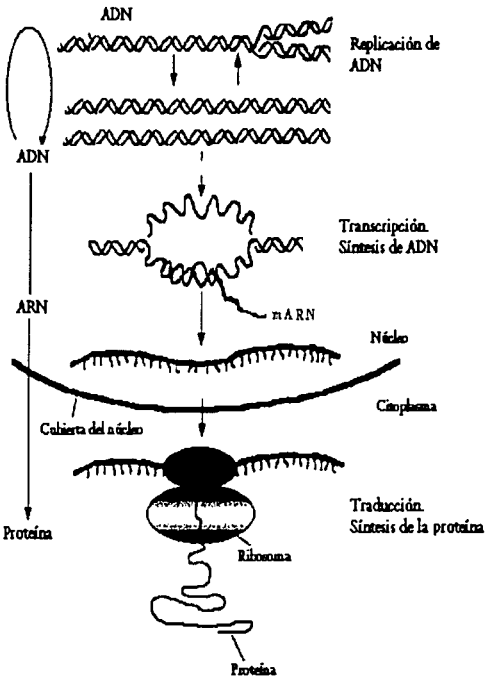


Figura 2.6: Síntesis de proteínas. Se muestra el proceso de transcripción en donde la información de una de las cadenas del ADN es copiada al ARN; El proceso de traducción, donde se ensamblan los amninoácidos especificados en el ADN (en esta fase el ARN tiene esa información), también se muestra.

noácido no son usados en la misma proporción, sino que algunos son usados mucho mas frecuentemente que otros. El patrón de uso de codones difiere no solo entre genes de un mismo organismo sino también entre organismos [78][85]. En la tabla 2.4 se muestra el triplete mas usado para la codificación de cada aminoácido en cinco organismos [78].

La primer columna de la tabla es la de los aminoácidos. Para cada organismo, se presenta el codón mas utilizado para la codificación del aminoácido mostrado, así como el porcentaje de uso del codón. Puede observarse que, como se mencionó con anterioridad, el uso de codones no parece ser aleatorio, puesto que la distribución sería semejante para todos los tripletes, lo que no ocurre, según se muestra en la tabla 2.4.

Aminoácido	Humano		Rata		E. Coli		S. Cerevisae		S. Frugiperda	
	C. pref.	% uso	C. pref.	% uso	C. pref.	% uso	C. pref.	% uso	C. pref.	% uso
Ala	GCC	41	GCC	41	GCC	34	GCT	38	GCT	37
Arg	CGG	21	AGG	21	CGC	38	AGA	48	AGA	24
Asn	AAC	55	AAC	60	AAC	54	AAT	59	AAC	63
Asp	GAC	54	GAC	58	GAT	63	GAT	65	GAC	58
Cys	TGC	56	TGC	56	TGC	55	TGT	63	TGC	58
Gln	CAG	75	CAG	76	CAG	66	CAA	69	CAG	51
Glu	GAG	59	GAG	62	GAA	68	GAA	71	GAG	52
Gly	GGC	35	GGC	35	GGC	39	GGT	47	GGA	32
His	CAC	59	CAC	62	CAT	57	CAT	64	CAC	60
Ile	ATC	50	ATC	55	ATT	50	ATT	46	ATC	47
Leu	CTG	41	CTG	42	CTG	49	TTG	29	CTG	31
Lys	AAG	58	AAG	64	AAA	75	AAA	58	AAG	58
Met	ATG	100	ATG	100	ATG	100	ATG	100	ATG	100
Phe	TTC	56	TTC	60	TTT	57	TTT	59	TTC	65
Pro	CCC	33	CCC	32	CCG	51	CCA	41	CCT/CCA	29
Ser	AGC	24	AGC	25	AGC	26	TCT	27	TCC	20
Thr	ACC	37	ACC	38	ACC	42	ACT	35	ACT	32
Trp	TGG	100	TGG	100	TGG	100	TGG	100	TGG	100
Tyr	TAC	57	TAC	61	TAT	58	TAT	56	TAC	67
Val	GTG	48	GTG	48	GTG	36	GTT	39	GTG	39
Trm	TGA	51	TGA	50	TAA	62	TAA	48	TAA	64

Tabla 2.4: Triplete mas usado para cada aminoácido en cinco organismos

El uso de codones ha sido atribuido a las presiones selectivas [9], entre las que se encuentra la eficiencia en la traducción, es decir, la habilidad del organismo para sintetizar la proteína codificada. El uso de codones también ha sido relacionado con el nivel de expresión de los genes y con la composición de nucleótidos en ciertas regiones del gen [33].
La frecuencia de aparición del par de bases GC (G + C) es también frecuentemente citado como una posible causa que determina el uso de codones del organismo [86][84]. Lo anterior se refiere al número de veces que ya sea del nucleótido G (guanina) o C (citosina) aparecen en la secuencia codificadora [84].
Por ejemplo, la secuencia AGCAGCAGCTTTATATATATGCGCGCATATCGA tiene un contenido de GC de 42.42%, en tanto que la misma secuencia, para ventanas de longitud 5, exhibe las siguientes frecuencias: posiciones 1-5: 60%, posiciones 6-10: 60%, posiciones 11-15: 0%, posiciones 16-20: 0%,

La tabla 2.5 muestra el vector de características para el perro. Puesto que existen 64 tripletes (variables o características), la dimensión del espacio es 64. Cada una de las variables tiene asociado un número: Cuando nos referimos a la característica 7, nos estaremos refiriendo al codón UCG, o, si hacemos referencia a la variable 32, estaremos hablando del codón AUU. Los 54 organismos analizados en el presente trabajo se muestran en la tabla 2.6 Parte central de éste es encontrar cuales de las 64 características resultan mas relevante para que el mapeo autoorganizado preserve la topología lo mejor posible.
Como se mencionó en la sección 1.2, el uso de codones ha sido estudiado con diversas técnicas estadísticas entre ellas la formación de dendrogramas [47]. La figura 2.7 muestra el dendrograma de los 54 organismos mostrados en la tabla 2.6 [73]. Con la formación de cúmulos por medio de dendrogramas, se intenta localizar parejas de objetos con la menor distancia (en el dendrograma mostrado, se uso la distancia euclidiana). Una vez encontrada la pareja de objetos mas cercano, se considera a ambos como un solo objeto, y se procede a localizar de nueva cuenta a las parejas con una distancia euclidiana mas pequeña, pudiendo en esa etapa ser tomado en cuenta el objeto conformado por los dos objetos mas cercanos previos [40].
Para explicar con algo más de detalle el dendrograma, veamos que los organismos *Fusobacterium nucleatum* y *Mycoplasma pulmonis* se encuentran en el mismo cúmulo, pues la distancia euclidiana entre el uso de codones de ambos es pequeña. Se observa que ambos organismos estan asociados por un par de líneas que se unen aproximadamente en el valor de 850, lo que indica que la distancia entre ambas es justamente esa. Los objetos mas cercanos de acuedo a su uso de codones son el *Homo sapiens* y el *Mus musculus* (ratón doméstico), separados por una distancia de alrededeor de 400.

2.5 Discusión

El uso de codones se refiere a como un organismo codifica un cierto aminoácido. Los aminoácidos son los bloques de construcción de las proteínas, las unidades fundamentales de la vida. Un mismo aminoácido puede ser codificado por diversos codones o tripletes.
El que un organismo prefiera un determinado triplete a otro no ha sido adecuadamente explicado por las teorías existentes y se observa que dicho uso no parece estar relacionado con el superreino, la frecuencia de $G + C\%$ o a presiones selectivas.
La síntesis de proteínas se lleva a cabo en dos fases principales: la transcripción, donde el ADN es copiado a ARN y la traducción, donde, para

posiciones 21-25: 100%, posiciones 26-30:20% y las posiciones 31-33: 66.6%. El contenido de GC en cada una de las posiciones del triplete ha sido considerado como una explicación del uso de codones. En el capítulo siete, se muestran algunas gráficas en las que se observa la distribución de los organismos analizados con base en su uso de codones, mostrando adicionalmente el contenido GC de cada uno de ellos.
Las observaciones anteriores han sido enunciadas con base en estudios realizados principalmente en bacterias. Sin embargo, en otros organismos las teorías ya no resultan adecuadas, pues aparecen demasiados contraejemplos [78]. No existe ninguna explicación satisfactoria para el uso de codones [72]. Haciendo uso de la información contenida en la base de datos de uso de codones [54], se visualiza, mediante mapeos autoorganizados, la distribución de los organismos analizados. Los organismos que hacen un uso de codones similar se encontrarán más cerca que aquellos organismos con un uso de codones diferente. Por medio de las distancias, es posible formar cúmulos y con ello, determinar algunas características evolutivas sobre dichas especies, tales como posibles escenarios o secuencias evolutivas. En la tabla 2.5 se muestra el formato de la base de datos.

UUU(17.6)	0	UCU(14.2)	4	UAU(12.1)	8	UGU(10.8)	12
UUC(24.3)	1	UCC(17.8)	5	UAC(18.1)	9	UGC(14.4)	13
UUA(5.9)	2	UCA(10.2)	6	UAA(0.6)	10	UGA(1.1)	14
UUG(11.5)	3	UCG(4.6)	7	UAG(0.4)	11	UGG(13.4)	15
CUU(11.6)	16	CCU(16.2)	20	CAU(9.3)	24	CGU(3.8)	28
CUC(20.7)	17	CCC(20.9)	21	CAC(13.9)	25	CGC(10.1)	29
CUA(6.4)	18	CCA(14.6)	22	CAA(11.4)	26	CGA(5.1)	30
CUG(40.7)	19	CCG(6.7)	23	CAG(30.9)	27	CGG(10.3)	31
AUU(15.9)	32	ACU(12.8)	36	AAU(17.1)	40	AGU(11.3)	44
AUC(25.5)	33	ACC(21.8)	37	AAC(22.0)	41	AGC(18.8)	45
AUA(7.3)	34	ACA(14.6)	38	AAA(23.2)	42	AGA(10.7)	46
AUG(22.9)	35	ACG(7.3)	39	AAG(32.8)	43	AGG(10.7)	47
GUU(10.0)	48	GCU(17.7)	52	GAU(20.9)	56	GGU(12.4)	60
GUC(17.4)	49	GCC(28.6)	53	GAC(27.1)	57	GGC(23.9)	61
GUA(7.0)	50	GCA(13.5)	54	GAA(26.8)	58	GGA(17.5)	62
GUG(30.4)	51	GCG(7.7)	55	GAG(38.1)	59	GGG(16.6)	63

Tabla 2.5: Uso de codones del *Canis familiaris* (perro). Para cada codón, se muestra el número que se le asignó para su referencia en este trabajo. p.e. el codón GCG es referido como característica 55, en tanto que el triplete UUC es la variable 1.

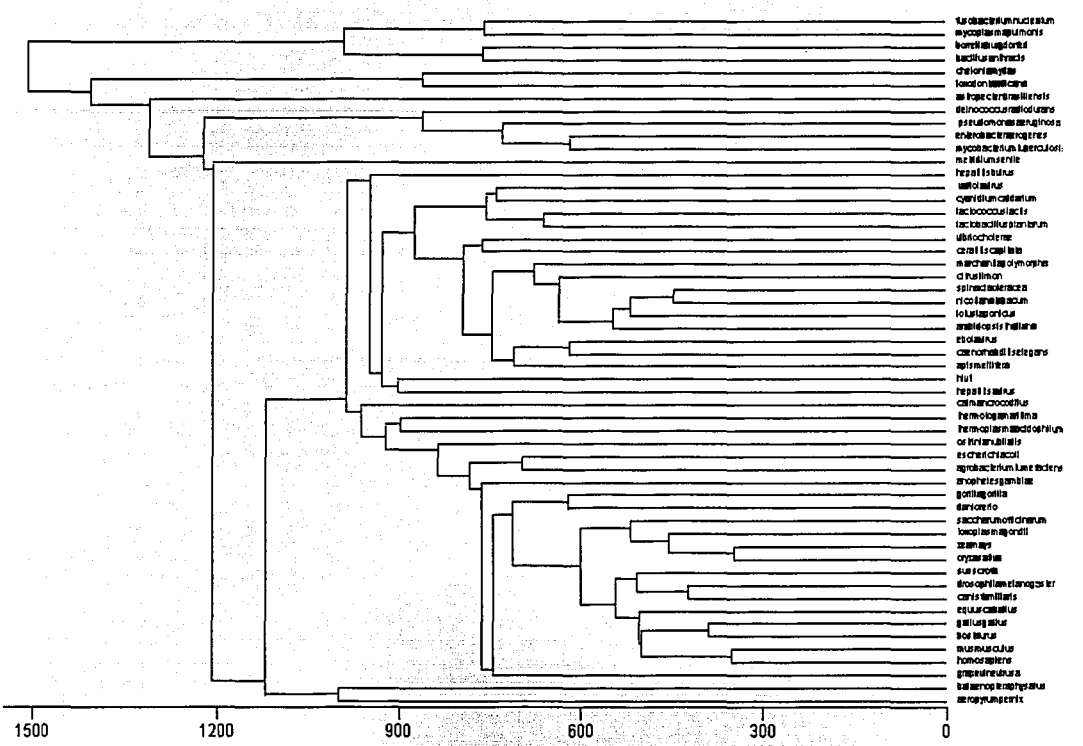
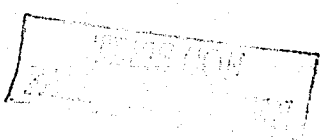


Figura 2.7: Dendrograma para el uso de codones de los organismos mostrados en la tabla 2.6.

Nombre	Reino	Nombre	Reino	Nombre	Reino
Aeropyrum pernix	AB	Drosophila melanogaster	E	Mycobacterium tuberculosis	B
Agrobacterium tumefaciens	B	Virus del Ébola	V	Mycoplasma pulmonis	B
Anopheles gambiae	E	Entero bacteraerogenes	B	Nicotiana tabacum	E
Apis mellifera	E	Equus caballus	E	Oryza sativa	E
Arabidopsis thaliana	E	Escherichia coli	B	Ostrinia nubilalis	E
Astropecten brasiliensis	E	Fusobacterium nucleatum	B	pseudomonas aeruginosa	B
Bacillus anthracis	B	Gallus gallus	E	Saccharum officinarum	E
Balaenoptera physalus	B	Gorilla gorilla	E	Spinacia oleracea	E
Borrelia burgdorferi	B	Virus A de la uva	V	Sus crofa	E
Bos taurus	E	Virus de la Hepatitis A	V	Thermoplasma acidophilum	AB
Caenorhabditis elegans	E	Virus de inmunodeficiencia humana 1	V	Thermotoga maritima	B
Caiman crocodilus	E	Homo sapiens	E	Toxoplasma gondii	E
Canis familiaris	E	Lactobacillus plantarum	B	Virus de la Viruela	V
Ceratitis capitata	E	Lactococcus lactis	B	Vibrio cholerae	B
Chelonia mydas	E	lotusjaponicus	E	Zea mays	E
Citrus limon	E	Loxodonta africana	E		
Cyanidium caldarium	E	Marchantia polymorpha	E		
Danio rerio	E	Metridium senile	E		
Deinococcus radiodurans	B	Mus musculus	E		

Tabla 2.6: Nombre científico de los 54 organismos analizados en este trabajo. Se muestra además el superreino al que pertenecen: E: Eucarionte, V: Virus, B: Bacteria, AB: Arqueobacteria.



Capítulo 3

Mapeos autoorganizados

3.1 Descripción General

En 1982, Teuvo Kohonen, mientras trabajaba en el problema de reconocimiento de voz, dio a conocer una topología de red neuronal no supervisada que mapea del espacio de características al espacio de baja dimensión de la red, preservando la topología existente en el espacio original [61].
Kohonen define el mapeo autoorganizado como "el resultado de una regresión no paramétrica que se usa principalmente para representar datos con dimensiones mayores que dos, que generalmente están relacionados de manera no lineal"[61].
El mapeo autoorganizado va del espacio de características al espacio de análisis, espacio formado por la malla de neuronas. En general, el espacio de análisis es de dimensión dos, pero esto no es una restricción. Este mapeo es una proyección no lineal que mapea un conjunto de datos multivariados (en general, de más de tres características) a un espacio de baja dimensión, fácilmente visualizable, como, por ejemplo, el espacio de dimensión dos. La figura 3.1 muestra esquemáticamente el proceso de mapeo autoorganizado [36].

En el mapeo autoorganizado cada neurona i cuenta con un vector de pesos, μ_i, que es el parámetro que permite comparar la semejanza entre una neurona y el vector de características que describe al objeto que en ese momento se analiza. Una vez que se muestra a la red el vector de características que describe al objeto a aprender, se elige una neurona *ganadora* (aquella que se parece mas al objeto).
La neurona ganadora *modificará* su vector de pesos y permitirá, además, que sus neuronas *vecinas* modifiquen sus vectores de pesos. El aprendizaje en el mapeo autoorganizado es de tipo *competitivo*, pues se elije aquella neurona

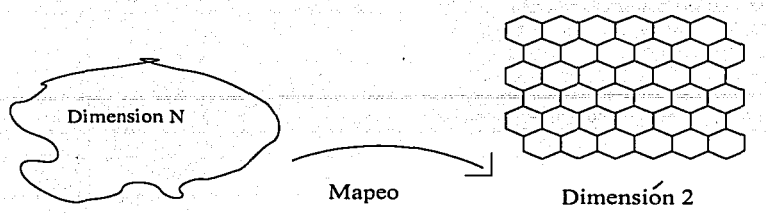


Figura 3.1: Proceso de mapeo. Del lado izquierdo se tiene un espacio multidimensional en tanto que del lado derecho se observa un espacio de dimensión dos.

con el vector de pesos con mayor semejanza al objeto a aprender y esta neurona ganadora únicamente permitirá aprender, durante un cierto tiempo, a sus neuronas más cercanas e, inclusive, después de otro lapso, el aprendizaje quedará restringido exclusivamente a ella [27, 17]. Esta modificación en el vector de pesos constituye el aprendizaje de la red [88]. Una neurona puede, además de permitir que sus neuronas aprendan, *inhibir* a las neuronas más distantes a ella.
La figura 3.2 muestra una malla bidimensional de neuronas. En esta red neuronal no existe una comunicación intensa entre neuronas, como es el caso de las redes de Hopfield o en los perceptrones multicapa, sino que existe la función de vecindad que permite a una neurona vecina a la neurona ganadora modificar sus pesos [88]. Puede observarse también el vector de pesos de cada neurona, que en general será diferente al vector de pesos de las restantes neuronas.

En la figura 3.3 se muestra la configuración de neuronas que resulta de aplicar el mapeo autoorganizado a las caracteristicas que describen a algunos países, mostrados en la figura 3.4. Las características que describen a los países fueron de carácter económico y de desarrollo. El número de estas características consideradas fue de 39, por lo que se aplicó un mapeo para poder apreciar la distribución de esos países con base en las mencionadas variables. [52].

Un objeto i es descrito mediante un vector de características x_i. La neurona ganadora n_g es aquella cuyo vector de pesos μ_g tiene menor distancia al vector de características x_i en el momento t

$$n_g = min(d(\mu_g, x_i)) \tag{3.1}$$

Aunque existen otras métricas para comparar el vector de características y el vector de pesos para identificar a la neurona ganadora, la distancia euclidiana

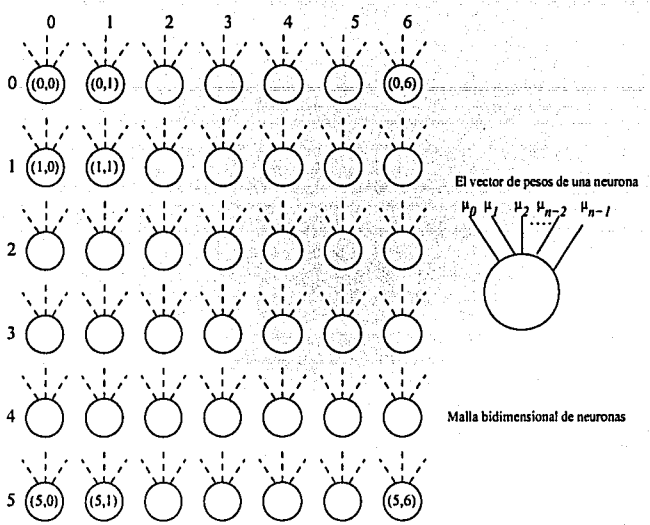


Figura 3.2: Malla bidimensional de neuronas. Cada neurona tiene un vector de pesos, detallado en la figura del lado derecho. En esta red, cada neurona es identificada por su posición. La neurona en la primera fila y columna es la $(0,0)$, en tanto que la neurona a su derecha es la $(0,1)$. Esta numeración permite calcular las vecindades.

es la más utilizada [13]. La modificación de los pesos en la neurona ganadora y en sus neuronas vecinas está dada por la siguiente ecuación:

$$\mu_n(t+1) = \mu_n(t) + \alpha_n h_t(g,n)(x_i - \mu_n(t)) \tag{3.2}$$

donde $h_t(g,n)$ es la función de vecindad de la neurona ganadora g a la neurona n, $\mu_n(t)$ es el vector de pesos de la neurona n en el tiempo t, x_i es el vector de características del objeto i y α_n es el factor de aprendizaje de la neurona n.
La función de vecindad es decreciente con respecto a la distancia y al tiempo. Al inicio del aprendizaje, ésta función suele ser cercana a 1 para todas las neuronas, lo que permite la formación de cúmulos. Hacia el final del aprendizaje, esta función de vecindad es cercana a 0 [36].
La explicación para la variabilidad en la función $h_t(g,n)$ es que al principio

Figura 3.3: Mapeo de países del espacio de características formado por diversos datos económicos y de desarrollo. Los países que son *mapeados* a neuronas cercanas, como Bélgica y Suecia, se encuentran en regiones cercanas en el espacio original, en tanto que países mapeados a neuronas alejadas, como Noruega y Etiopía, se encuentran alejados en el espacio de características (espacio original)[61].

del proceso, para preservar las vecindades, es necesario que se permita el aprendizaje de las neuronas cercanas a la ganadora; sin embargo, también es deseable que objetos muy cercanos en el espacio de características sean mapeados a neuronas diferentes. Esto es, que se cuente con cierta resolución para diferenciar aún a objetos semejantes, por lo que en algún momento durante la autoorganización, se detendrá la *ayuda* que otorga la neurona ganadora a sus neuronas vecinas [36].
Cada objeto en el espacio multidimensional es mapeado únicamente por una neurona en el espacio de baja dimensión del mapeo. Dos objetos pueden, sin embargo, ser mapeados por la misma neurona, lo que indica que dichos objetos son muy semejantes en sus características [76]. El conjunto de neuronas que mapean a los diversos objetos son conocidas como *neuronas ganadoras* [88].
La neurona que resulta ganadora para un cierto vector que describe un objeto, esto es, aquella cuyo vector de pesos es el más semejante al objeto, es la que mejor *responde* para el objeto. El mapeo autoorganizado ha sido propuesto como una herramienta que modela algunas regiones del cerebro de los mamíferos superiores, en particular los conocidos como *mapas ordenados* en la corteza cerebral [59].
Esta red está inspirada en ciertas regiones del cerebro de diversos anima-

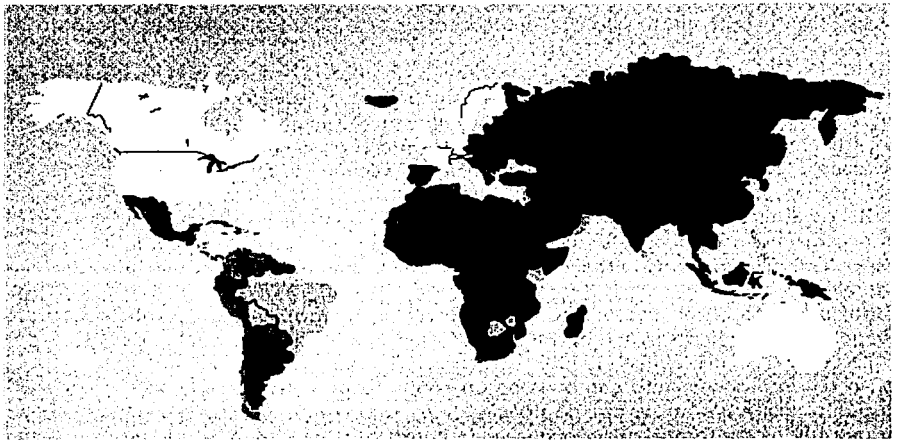


Figura 3.4: Mapa de los países analizados.

les. Cuando el cerebro recibe información sensorial, un grupo de neuronas responde a dicha información. Este grupo puede extenderse incluso varios centímetos, como es el caso del mapa *retinóptico*[1] del campo visual. Este grupo de neuronas recibe el nombre de *mapa cerebral* y se encarga de procesar la información recibida [60][69].
El funcionamiento de esta red se debe a la retroalimentación no lineal que impera en la función de aprendizaje o modificación de los pesos. En el cerebro, se han encontrado regiones en las que ciertas neuronas se conectan hasta con 10000 neuronas, con *excitación* a las neuronas mas cercanas (entre 50 y 100 micrones) y con inhibición a las neuronas que se encuentran entre 200 y 500 micrones [22].

3.2 Ordenamiento y convergencia

El por qué una neurona ganadora permite, durante una gran parte del proceso de aprendizaje, la modificación del vector de pesos de sus neuronas vecinas, se explica por la idea de preservación de topología: Objetos cerca-

[1] El mapa retinóptico es la región de la corteza que recibe los estímulos provenientes de la retina [80].

nos en el espacio de características lo serán también en el espacio de mapeo, generalmente de dimensión dos [29]. La *actividad* en la vecindad de una neurona tiene como consecuencia que las neuronas localizadas en esa vecindad representen al objeto de manera semejante a sus vecinas, lo que resulta en un mapa *topográfico* [80].

Si la distancia entre los vectores de pesos de las neuronas que mapean dos organismos es pequeña, la distancia entre las dos neuronas también lo será [58]. Si $d(\mu_a, \mu_b) < \Theta$ entonces $d(N_a, N_b) < f\Theta$, donde f es una función no lineal.

El ordenamiento topológico ocurre en dos fases. En la primera, la fase de ordenamiento, los pesos se modifican de tal forma que quedan ordenados ya sea ascendente o descendentemente. Supongamos, para el caso en que el mapeo sea unidimensional, que se tiene un conjunto de n neuronas. Cada neurona i cuenta con un vector de pesos μ_i y $\mu_{i\alpha}$ representa el peso α de la neurona i. De esta forma, una vez que se da el ordenamiento, ocurre una de las dos situaciones siguientes [61]:

$$a)\ \mu_{0\alpha} \leq \mu_{1\alpha} \leq \ldots \leq \mu_{i\alpha} \leq \ldots \mu_{n\alpha} \tag{3.3}$$

ó

$$b)\ \mu_{0\alpha} \geq \mu_{1\alpha} \geq \ldots \geq \mu_{i\alpha} \geq \ldots \mu_{n\alpha} \tag{3.4}$$

Empíricamente, se ha encontrado que durante la fase de ordenamiento, la vecindad inicial suele ser grande (una $h(.)$ cercana a 1) y disminuir hasta su valor final, que en general es cercano a 0.2 [62] [36].

En la fase de convergencia, después que los pesos alcanzan una de las configuraciones expresadas en las ecuaciones. 3.3 y 3.4, éstos continuarán modificándose de tal forma que no violen la configuración alcanzada durante el ordenamiento [13].

En la fase de convergencia, los pesos de la neurona que mapea a cada objeto se modificarán de tal forma que la distancia entre el vector de pesos y el vector que describe al objeto sea mínima, incluso llegando a 0 [52]. Esto se conoce como precisión en el mapeo.

La vecindad inicial en la fase de convergencia suele elegirse cercana a 0.15, en tanto que la vecindad final para esta fase (y por lo tanto, para el mapeo) es cercana a cero.

En la figura 3.5 se muestra la distribución de pesos de una red de neuronas de dos entradas. Se tienen 30 neuronas, cada una con su respectivo vector de pesos.

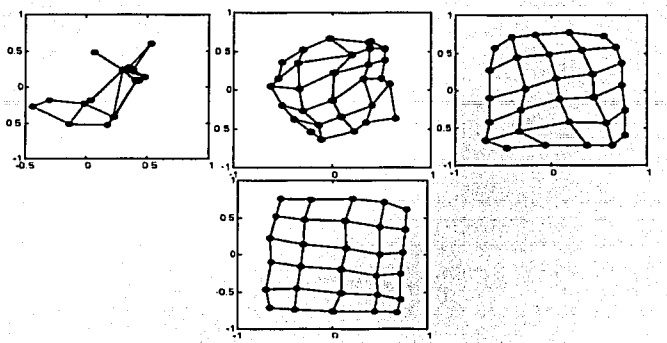


Figura 3.5: Ordenamiento de los pesos de las neuronas. Los pesos de neuronas contiguas en la malla se unen por medio de líneas. Las imágenes son para $t = 0, t = 60, t = 300, t = 500$.

Una línea recta une a neuronas contiguas. Por ejemplo, las neuronas contiguas de la neurona $(0, 0)$ en la figura 3.2 son las neuronas $(0, 1)$ y $(1, 0)$. En el tiempo $t = 0$, y puesto que los pesos son inicializados de manera aleatoria, no se observa ningún patrón en las líneas que unen a las neuronas. En el tiempo $t = 60$, después de que los pesos se han modificado de acuerdo a las ecuaciones 3.1 y 3.2, las neuronas contiguas comienzan a ordenarse, es decir, presentan alguna de las configuraciones expresadas en las ecuaciones 3.3 y 3.4
Una vez concluida la fase de ordenamiento, al entrar a la fase de convergencia, los pesos se modificarán a un ritmo menor. Esto se observa al analizar las figuras 3.5c y 3.5d. Las neuronas $(0, 0)$ y $(0, 1)$ continúan siendo vecinas, pero ahora sus vectores de pesos serán también contiguos [62], lo que se observa en la malla casi simétrica que resulta.

3.3 Vecindad

Cuando una neurona resulta *ganadora* modificará sus pesos con lo que *aprenderá* mejor el vector de características que describe al objeto que le fue presentado. En otras palabras, la neurona ganadora se parecerá mas al objeto que esta mapeando [88].
El aprendizaje no está limitado únicamente a la neurona ganadora, elegida

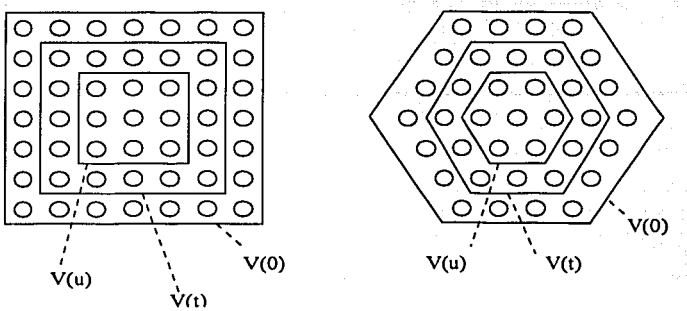


Figura 3.6: Reducción de la vecindad de una neurona como función del tiempo

mediante la ecuación 3.1, sino que también sus *vecinos* aprenderán. Por vecinos se entienden las neuronas *cercanas* a una neurona.
Una neurona es vecina de otra si la distancia entre ambas es menor a cierto umbral. De la figura 3.2, podemos ver que la neurona $(0, 0)$ tiene como vecina a la neurona $(0, 1)$ si el valor del umbral es uno, pero, para ese mismo umbral, la neurona $(2, 0)$ ya no resulta vecina. Si el umbral es mayor, el número de neuronas vecinas también es mayor.
La vecindad no es siempre la misma. Al comienzo del entrenamiento, una neurona tiene por vecindad a todas, o casi todas las neuronas, en tanto que hacia el final del proceso, la vecindad queda restringida solamente a la misma neurona ganadora. La figura 3.6 muestra como la vecindad es decreciente con respecto al tiempo [13, 27].

En el tiempo 0, la neurona ganadora permitirá que todas las neuronas aprendan. Después de un cierto tiempo t, la vecindad se ve reducida y las neuronas más alejadas ya no aprenderán para el vector presentado en ese momento, sino exclusivamente aquellas que se encuentren dentro del segundo polígono. La disminución de la vecindad llevará, hacia el final del proceso, a no considerar a ninguna neurona como vecina, salvo a la neurona ganadora. Existe una alternativa para definir la vecindad de una neurona a través de la función de vecindad [36]:

$$h(g, i) = \alpha(t) e^{(-d(g,i)^2 / 2\sigma^2(t))} \tag{3.5}$$

donde $\alpha(t)$ es el factor de aprendizaje, $d(N_g, N_i)$ es la distancia entre las neuronas N_g (la ganadora) y N_i y $\sigma(t)$ se define como el radio de la vecindad (indicado en la figura 3.6 como V(t)). Tanto $\alpha(t)$ como $\sigma(t)$ son funciones

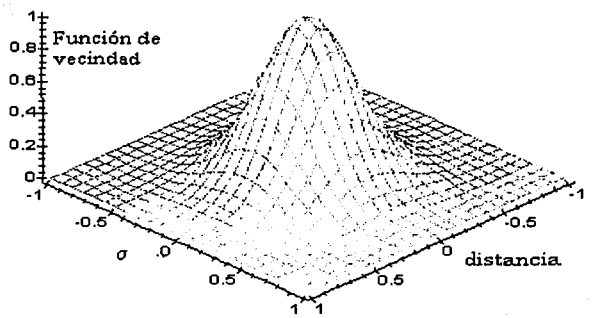


Figura 3.7: Función de vecindad de la neurona ganadora g a la neurona i, $h(g,i)$

monotónamente decrecientes.
$h(g,i)$ es una función decreciente para la distancia entre g y i, de tal forma que $h(k,k) = 1$. Los esquemas de vecindad mostrados en la figura 3.6 son un caso especial de esta ecuación [13]. La figura 3.7 muestra la gráfica para la función de vecindad anterior.

El algoritmo de entrenamiento para esta topología de red se muestra en el recuadro siguiente [17].

1. Iniciar la red.
Para cada neurona i, definir su vector de pesos μ_i. Existen dos formas de iniciar los pesos: aleatoria y ordenada [2]. Se elige la vecindad ($\sigma(0)$) de tal forma que incluya a todas, o casi todas las neuronas de la red. Se elige un factor de aprendizaje inicial, α, cercana a 1 para todas las neuronas.

2. Se presenta el vector de características del objeto n, x_n.

3. Calcular las distancias entre x_n y el vector de pesos de to-

[2]Por inicio ordenado se entiende que los pesos se inician de tal forma que cumplan con la ec. 3.3 ($\mu_{0\alpha} \leq \mu_{1\alpha} \leq ... \leq \mu_{i\alpha} \leq ...\mu_{n\alpha}$) o con la ec. 3.4 ($\mu_{0\alpha} \geq \mu_{1\alpha} \geq ... \geq \mu_{i\alpha} \geq ...\mu_{n\alpha}$)

das las neuronas, $d(x_n, \mu_g)\ \forall g$.

4. Seleccionar como neurona ganadora aquella con distancia mínima al vector de características:
$i = min(d(x_n, \mu_i))$

5. Modificar el vector de pesos de la neurona ganadora y de aquellas neuronas que esten en la vecindad ($\sigma(t)$) de la neurona ganadora:
$\mu_n(t+1) = \mu_n(t) + \alpha h_t(g,n)(x_i - \mu_n(t))$

6. Reducir la vecindad $\sigma(t+1)$: $\sigma(t) = f(\sigma(t), t)$

7. Reducir el factor de aprendizaje para la neurona ganadora, $\alpha_g = 1/(\alpha_g + 1)$ [3]

8. Repetir los pasos 2 al 7 hasta que los pesos no se modifiquen (convergencia).

3.4 Preservación de topología

Los mapeos autoorganizados tienen la propiedad de conservar en el espacio de análisis la topología que presentan los objetos en el espacio de características; la representación de la estructura básica del ambiente no es la única ventaja: representar las relaciones de vecindad definidas sobre los objetos analizados es la principal de ellas [80].
Puesto que el mapeo es utilizado para la visualización de datos a través de la distribución de las neuronas ganadoras, es necesario que se tengan medios para evaluar los mapeos. En otras palabras, si acontece que dos objetos se encuentran en posiciones cercanas en el espacio de análisis (esto es, si dos neuronas vecinas resultan las ganadoras para dichos objetos), deberán entonces encontrarse en regiones cercanas en el espacio de características [49]. En la figura 3.8 se observa que la topología presente en el espacio original de dimensión tres se conserva en el espacio de análisis de dimensión dos [61].

Durante la autoorganización ocurren dos sucesos que preservan la topología. En el primero, el conjunto de vectores de pesos tiende a describir la

[3] Se modifica el factor de aprendizaje solamente para la neurona ganadora para recrear lo que parece que ocurre en las neuronas naturales: cuando una neurona ha tenido cierta actividad, ésta no aprenderá con la misma efectividad pues se encuentra en un periodo de agotamiento[60].

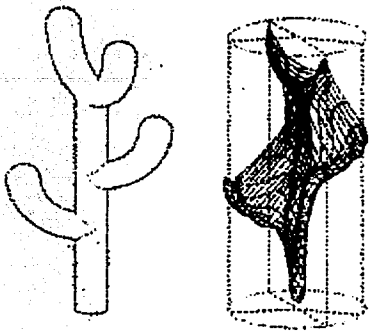


Figura 3.8: Conservación de la topología

función de densidad de los vectores de entrada (vectores de características). En el segundo, las interacciones locales entre neuronas tienden a preservar la *continuidad* en las secuencias de pesos [61][62].
Formalmente, una topología en un conjunto de datos D es generada al definir una métrica o distancia sobre ese conjunto de datos. Una distancia en un conjunto D es una función real definida en el producto cartesiano DxD tal que para todo elemento $x, y, z \in D$ se tiene que [3]:

(a) $d(x, y) \geq 0$ con igualdad sí y solo sí $x = y$.
(b) $d(x, y) = d(y, x)$.
(c) $d(x, y) + d(y, z) \geq d(x, z)$.

La dimensión del espacio de análisis es definida por la hipermalla en la que se encuentran las neuronas. Para la visualización de objetos multidimensionales, la dimensión del espacio de análisis es dos o tres. La figura 3.9 muestra la configuración de neuronas en una malla bidimensional y en una malla tridimensional [88].

Definido de manera formal, un mapeo que preserva la topología es una transformación $\Phi : R^k \rightarrow R^p$ que preserva ya sean las *similitudes* o únicamente los órdenes de similitud de los puntos en el espacio de entrada R^k cuando son mapeados al espacio de salida R^p[29].

Una transformación $\Phi : \hat{x} = \Phi(x)$ que preserva las similitudes indica que

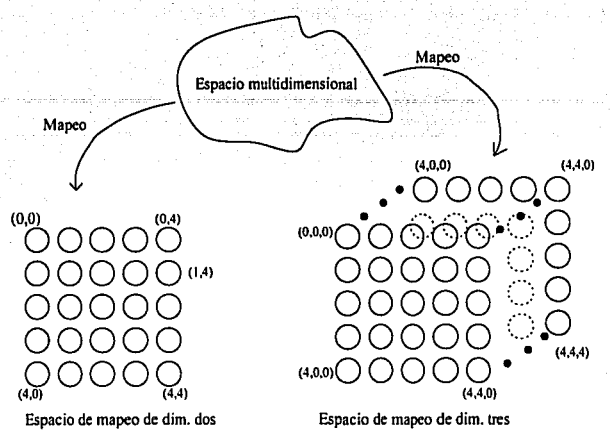


Figura 3.9: Espacios de análisis bidimensional (malla bidimensional, izq.) y tridimensional (malla tridimensional, der.)

$d(x_i, x_j) = \hat{d}(\hat{x}_i, \hat{x}_j)\ \forall x_i, x_j \in R^k,\ \forall \hat{x}_i, \hat{x}_j \in R^p$. Una transformación $\Phi : \tilde{x} = \Phi(x)$ que solamente preserva el orden de similitud indica que $d(x_1, x_2) \leq d(x_3, x_4) \iff \tilde{d}(\tilde{x}_1, \tilde{x}_2) \leq \tilde{d}(\tilde{x}_3, \tilde{x}_4)$. El mapeo autoorganizado corresponde a transformaciones del segundo tipo.

La preservación de la topología se puede resumir de la siguiente forma. Para todo objeto i en el espacio de características (espacio de entrada), si todos los objetos que estén dentro de la hiperesfera con radio r y con centro en i, son mapeados a neuronas que se encuentran a una distancia $r\alpha$ de la neurona que mapea al objeto i, la topología se habrá preservado [56]. Lo anterior es válido únicamente para espacios lineales [8]

Otra manera de explicar la idea anterior es la siguiente. Si un objeto i tiene a $v_1, v_2, ..., v_k$ como a los k vecinos en el espacio de características, y las neuronas $n_1, n_2, ..., n_k$ mapean respectivamente a los objetos mencionados, con la restricción adicional de que $n_1, n_2, ..., n_k$ son las k neuronas activas mas cercanas a n_i, la topología se habrá preservado.

El mapeo autoorganizado preserva las relaciones de vecindad que existen en el espacio de características, aunque las relaciones que *mejor* se preservan son aquellas entre los *vecinos mas próximos*. Esto es, para una k pequeña, el mapeo autoorganizado es mejor que otras alternativas, como se muestra

en el capítulo seis [91]. El parámetro k será analizado con mayor detalle en el capítulo cuatro.
Si la dimensión del espacio de características, N, es mayor a la dimensión del espacio de análisis, S, los pesos de las neuronas intentarán plegarse de tal forma que la topología presente en el espacio de características para los objetos analizados, se vea reflejada en los pesos de la neurona.

3.5 La selección de parámetros para el mapeo autoorganizado

Tanto el factor de aprendizaje, α, así como la función de vecindad, h, son dos parámetros en los que se debe tener especial cuidado a fin de obtener un mapeo adecuado (ver sección 4.1).
Para el caso del factor de aprendizaje, se tiene evidencia de una mejor preservación de la topología cuando éste es monótonamente decreciente. Esto es, para el factor de aprendizaje en el tiempo t, $\alpha(t) \leq \alpha(t-1)$ [61]. En general, el factor de aprendizaje no es crítico en su decremento: puede ser lineal, exponencial o inversamente proporcional a t y los resultados no se alteran significativamente [37].
Haykin sugiere que durante las primeras 1000 iteraciones, el valor de $\alpha(t)$ debe estar por encima de 0.1 y con un $\alpha(0) \approx 1.0$ [36]. Esta fase coincide con el ordenamiento de los pesos (ver secc. 3.3). Después de las primeras mil iteraciones, el mismo autor recomienda que $\alpha(t)$ muestre un valor pequeño, del orden de 0.01 y continúe decreciendo.
En cuanto a la función de vecindad $h(t)$, se puede definir de tal forma que incluya a los vecinos en una región rectangular (ver figura 3.6a), pero también se puede definir una vecindad de tipo hexagonal u octagonal (figura 3.6b). Sin embargo, como se comenta en la sección 3.3, existe la posibilidad de incluir una función de vecindad contínua [61]. La forma de la función de vecindad no es crítica en tanto sea monótonamente decreciente [36].
Durante las primeras 1000 iteraciones, [36] recomienda que la vecindad disminuya desde el máximo posible (cubrir todas o casi todas las neuronas en en tiempo $t = 0$), hasta una fracción que cubra las neuronas más próximas. Después de ese tiempo, en la fase de convergencia, se recomienda que únicamente las vecinas inmediatas sean consideradas; hacia el final del proceso, solamente la neurona ganadora es la que modificará sus pesos pues la función de vecindad solo la incluye a ella.
Si la función de vecindad no fuese aplicada, esto es, si una neurona no permitiera que, al menos durante un tiempo, sus vecinas aprendieran, la topología

no se preservaría. Por otro lado, si la función de vecindad no fuera decreciente, no habría una diferenciación entre objetos similares [37].

3.6 El mapeo autoorganizado como herramienta de visualización

El mapeo autoorganizado ha sido ampliamente usado como herramienta de minería de datos y visualización [43], pese a que se ha sugerido que algunas técnicas tradicionales podrían ser mejores elecciones [30]. Sin embargo, en la mayoría de esos trabajos, se ha evaluado la distribución que genera el mapeo autoorganizado en términos de lo que hace la técnica tradicional con la que se compara.

Si quisieramos saber cuales son los objetos más cercanos a un cierto objeto en el espacio de características, solo habría que listarlos en orden de proximidad calculando la distancia euclidiana entre el objeto referido y los restantes. Sin embargo, si el propósito es *visualizar* la distribución que dichos objetos muestran en el espacio multidimensional, el listado de proximidades no es adecuado.

En este punto es conveniente recordar que el mapeo autoorganizado va de la dimensión del espacio de características al espacio del mapeo o análisis, generalmente de dimensión dos o tres, lo que se traduce en una reducción de la dimensión; además, es necesario recordar que el mapeo autoorganizado preserva la topología, entendida ésta como las relaciones de vecindad.

Para visualizar objetos multidimensionales es necesario mapearlos a un espacio bidimensional, procurando preservar la topología al máximo. Es aquí que las características del mapeo autoorganizado le dan la posibilidad de ser empleado como herramienta de visualización, pues preserva la topología [53]. La configuración de neuronas ganadoras (aquellas que mapean a los objetos analizados, esto es, aquellas cuyo vector de pesos se asemeja mas al vector de características que describe al objeto que mapean) mostrará una aproximación de la distribución de los objetos en el espacio de alta dimensión original. Con esta configuración de neuronas podemos afirmar, puesto que la topología se ha preservado [4], que conocemos, en términos generales, la distribución de los objetos en el espacio de características.

Por el momento, es suficiente mencionar que la bibliografía es extensa en cuanto al uso del mapeo como herramienta de visualización de objetos multidimensionales [93, 55], pero en el capítulo seis compararemos el desempeño del mapeo autoorganizado con técnicas tradicionales de visualización y mos-

traremos que en general es una herramienta adecuada.

3.7 El mapeo autoorganizado y la criticalidad autoorganizada

Se ha intentado demostrar que el conjunto de neuronas activas del mapeo autoorganizado, o *configuración organizada*, es un conjunto de puntos que preservan la definición de topología para los objetos analizados en el espacio de características de alta dimensión [95][8].
Por otro lado, en [28] se afirma que los intentos actuales por definir lo que significa una configuración de neuronas ganadoras en el mapeo autoorganizado no son útiles para explicar analíticamente este proceso. Para intentar darle una explicación alternativa al mapeo autoorganizado, propone que dicho proceso sea visto como un sistema con *criticalidad autoorganizada.*
La criticalidad autoorganizada es definida por Bak, Tang y Wiesenfeld [14] como un marco para entender la ocurrencia de las leyes de potencia en la naturaleza: los sistemas que presenten una criticalidad autoorganizada desarrollan de manera espontánea correlaciones de largo alcance sin la necesidad de que factores externos intervengan. La criticalidad autoorganizada tiene relación con la invarianza de escala espacio temporal en los eventos de un cierto fenómeno. Existen, por ejemplo, muchos temblores de tierra de baja magnitud, pero pocos temblores de alta magnitud. Al graficar el logaritmo de la magnitud de dichos movimientos contra el logaritmo del número de temblores para una magnitud dada, como se muestra en la figura 3.10, se observa que los puntos aproximan una línea recta.
Cuando se grafica el logaritmo de la intensidad de cierto fenómeno contra el número de veces que se ha presentado y se obtiene una línea recta, nos encontramos con un evento que sigue la *ley de potencias.* Los sistemas que son descritos por la ley de potencias presentan la característica de que entre mas intensos sean los eventos, menos frecuentes serán [34]. Matemáticamente, tenemos que:

$$N(s) = s^{-\tau}$$

Donde $N(s)$ es el número de eventos de intensidad s y τ es la pendiente de la línea recta que se obtiene al graficar la ecuación anterior.

[4]En realidad, cada vez que se mapea de un espacio de dimensión N a un espacio de dimensión M, con $M < N$, la topología se viola, pero lo que diferencia al mapeo autoorganizado de otras técnicas, es que las relaciones de vecindad son mejor preservadas, al menos para los primeros vecinos.

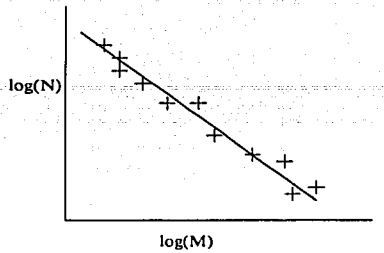


Figura 3.10: Magnitud de eventos sísmicos y número de eventos para esa magnitud. En el eje x se encuentra la magnitud, en tanto que en el eje y se encuentra el número de eventos para esa magnitud. Se observa que a medida que el logaritmo de la magnitud aumenta, el número de eventos decrece [14].

Los sistemas que se rigen por la ley de potencias presentan la misma estructura en todas las escalas. Un sistema es libre de escalas si se mantiene sin cambio, en sentido estadístico, al ser analizado en diversas escalas: Esto es, si al profundizar en el detalle (acercamiento) o al perder resolución (alejamiento), se observa la misma estructura original, el fenómeno analizado es libre de escala. Los eventos libres de escala se rigen por la ley de potencias [46].

El atractor [28] en los mapeos autoorganizados consiste en una colección de estados organizados. Para esta configuración, el sistema es crítico, por lo que muestra correlaciones espaciales y temporales. En la fase de ordenación, la entrada puede ser considerada como ruido que guía al mapeo a su configuración ordenada, por lo que el ordenamiento es independiente de la entrada. En el sistema que es el mapeo autoorganizado, la variable dinámica, aquella que presenta valores libres de escala, está dada por:

$$y = h(n_i, n_g)|x - \mu_i| \tag{3.6}$$

donde $h(a, b)$ es la función de vecindad de la neurona ganadora g a la neurona i; x es el vector de características y μ_i es el vector de pesos de la neurona i.

De esta forma, cuando se modifican los pesos de una neurona i, la variable y se puede modificar *poco* o *mucho*, esto es, se modificará siguiendo una distribución libre de escalas, como la mostrada en la figura 3.10. En otras palabras, en muchos ciclos de modificación (aprendizaje), el cambio en y

será pequeño, en tanto que en solo algunos ciclos, el cambio que sufra y será grande.

3.8 Discusión

El mapeo autoorganizado es una red neuronal no supervisada que presenta la característica de preservar la topología. La visualización de objetos multidimensionales mediante un mapeo es de gran importancia, y el mapeo autoorganizado es una herramienta adecuada para ello.
En el mapeo autoorganizado se tiene una red, generalmente bidimensional, de neuronas, de las cuales una se activará para cada objeto analizado. El conjunto de neuronas que mapean (responden) para cada objeto, recibe el nombre de configuración ordenada o neuronas ganadoras.
La autoorganización se lleva a cabo en dos fases: la ordenación y la convergencia. En la primera, los pesos de las neuronas se ordenan ya sea ascendente o descendentemente. En la segunda fase, los pesos se moverán, sin violar el orden, para tratar de representar lo mejor posible la distribución de los objetos en el espacio multidimensional.
La preservación de la topología es una característica deseada en las herramientas de visualización. Al preservarse la topología en un mapeo, sabemos que estamos visualizando una aproximación adecuada de lo que ocurre en el espacio de alta dimensión.

Capítulo 4

Evaluación de mapeos

4.1 Introducción

Visualizar objetos en espacios de alta dimensión (mayor a tres) requiere de un mapeo a un espacio de baja dimensión, generalmente de dimensión dos, a fin de poder darnos una idea de la distribución de dichos objetos en el espacio original [82].
Cuando se mapea a un espacio de menor dimensión suele violarse la topología presente en el espacio original [93], pero, la *bondad* de un mapeo será mejor en medida que la topología se viole lo menos posible.
Una de las características mas notables de los mapeos autoorganizados es la de preservar la topología. Recordemos que la preservación de topología se refiere a la conservación, en el espacio del mapeo, de los objetos vecinos en el espacio de características, esto es, que se conserven las relaciones de vecindad. Puesto que el mapeo autoorganizado reduce la dimensión, la preservación de la topología no es exacta [56][91].
Recordemos también que los pesos son inicializados aleatoriamente y el orden en que se encuentran los datos de entrenamiento puede variar, por lo que los mapeos que se realicen sobre un mismo conjunto de datos no serán iguales [62].
Si bien los mapeos autoorganizados para un mismo conjunto de datos no serán iguales, si serán *semejantes*. Por mapeos semejantes nos referimos a que presenten una distribución semejante de las neuronas ganadoras, o configuración ordenada, lo que significa, en última instancia, que preservan la topología de manera equivalente. Las figuras 4.1 y 4.2 muestran dos mapeos autoorganizados para un mismo conjunto de datos con raices de generación diferentes. En las figuras se observa que las neuronas ganadoras no son las mismas, pero las relaciones de vecindad son semejantes.

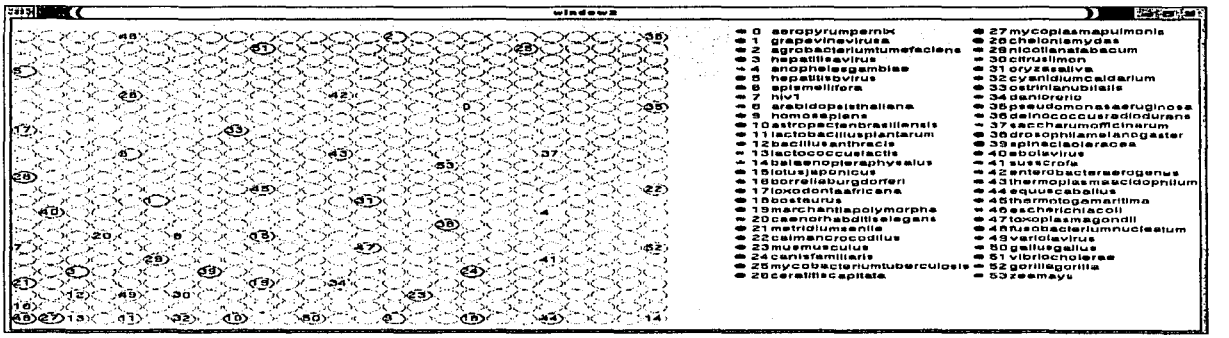


Figura 4.1: Distribución de los organismos analizados con base en su uso de codones obtenida por un mapeo autoorganizado.

Los mapeos son de la frecuencia del uso de codones de 54 organismos. La tabla 2.7 muestra el nombre científico y el nombre común para cada uno de ellos. Tal como comentamos en el capítulo 2, existen 64 codones que pueden codificar ya sea un aminoácido o una secuencia de paro, por lo que el espacio de características es de dimensión 64. Cada característica es la frecuencia de uso para cada codón.

¿Cuál de los dos mapeos es mejor? Ambos son semejantes, pero alguno de ellos preserva mejor las vecindades (preserva la topología) en mayor medida que el otro. Queremos encontrar el mejor mapeo porque al identificarlo, sabremos que ese es la mejor aproximación a lo que está ocurriendo en el espacio original. Al analizarlo, sabremos que estamos analizando algo muy parecido a la distribución de mediciones u objetos que identifican al fenómeno o proceso que queremos analizar [91].

Aunque en este capítulo analizaremos diversas métricas de preservación de topología, no existe una definición precisa de lo que en realidad es esa preservación [63]. En las secciones siguientes se revisarán diversas métricas de preservación de la topología.

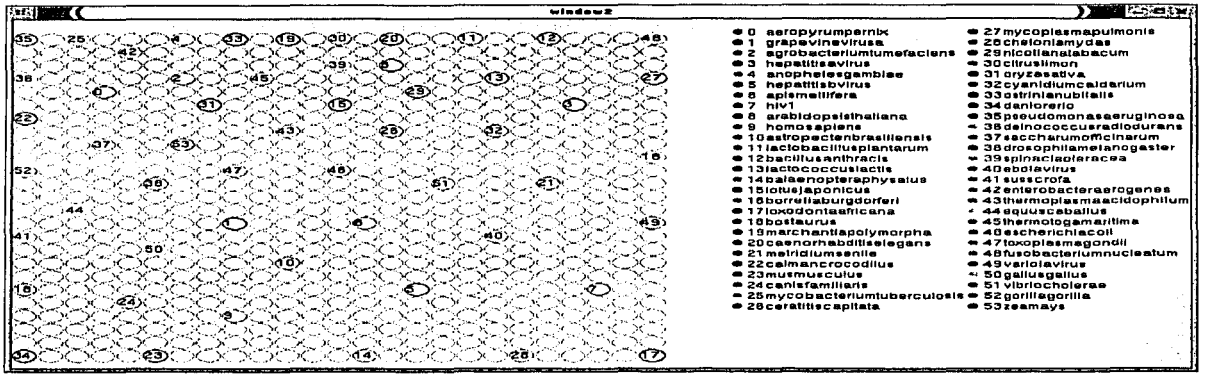
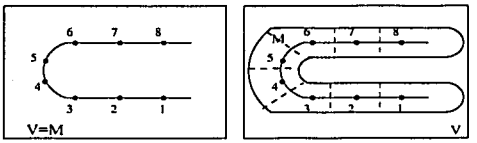


Figura 4.2: Segundo mapeo autoorganizado del uso de codones de los 54 organismos listados en la tabla 2.7

4.2 Producto topográfico

Esta métrica compara entre sí los vectores de pesos de las neuronas de la malla y si encuentra discontinuidades, o *pliegues*, considera que el mapeo está intentando aproximar lo que ocurre en el espacio multidimensional de características, lo que genera un error topográfico [56].
Esta metodología fue propuesta originalmente en la dinámica no lineal y en el análisis de series de tiempo con el propósito de seleccionar la dimensión óptima para la reconstrucción de atractores caóticos [8, 18].
La figura 4.3 muestra el principal inconveniente de usar esta métrica. El producto topográfico es incapaz de diferenciar entre un pliegue que en realidad si ocurre en el espacio de características y un pliegue erróneo. De esta manera, solo se garantiza que de resultados correctos para casos en que el espacio de características sea lineal. La figura 4.3a muestra un espacio lineal en tanto que la figura 4.3b muestra un espacio no lineal, en forma de U, lo que produciría que la métrica arrojara errores.



a) b)

Figura 4.3: Ejemplo de un espacio lineal en donde los poliedros de Voronoi pueden cubrir el espacio ($V = M$) y un no lineal en donde los poliedors de Voronoi no pueden cubrir al espacio. ($M \subset V$). [56]

4.3 Función topográfica

Para evitar los errores de la métrica anterior, Villman propuso otra métrica de cálculo de preservación de la topología. Para ello, se define el campo receptivo de una neurona n_i como $R_i = V_i M$, donde M denota el espacio de entrada y V_i es el poliedro de Voronoi de la neurona n_i [95]. La definición del poliedro de Voronoi es la siguiente:

$$V_i = [z|z \in R^n; |z - \mu_i| \leq |z - \mu_j| \forall j \neq i].$$

Ahora, la función que indica la bondad del mapeo es:

$$\Phi_L^M(s) = \sum \#(n_j|j \in L, |n_i - n_j| > s, n_i \text{ y } n_j \text{ adyacentes})$$

Donde # denota la cardinalidad de un conjunto y L es el conjunto de índices de las neuronas en la malla del mapeo, expresando el número de neuronas que tienen campos receptivos adyacentes en el espacio de entrada, pero que se encuentran mapeados a neuronas cuya distancia de bloque es mayor que s.
La figura 4.4 muestra la gráfica de esta función contra s. Un mapeo es mejor que otro si la función decae con mayor rapidez a 0 [94]. El inconveniente de este método es la dificultad de comparar funciones, pues resulta mucho mas simple la comparación entre escalares.

4.4 Error topográfico

Cuando el énfasis en la métrica de preservación de la topología dejan de ser los campos receptivos adyacentes y se toma en cuenta ahora la proporción de los vectores de entrenamiento que indiquen una discontinuidad en el mapeo,

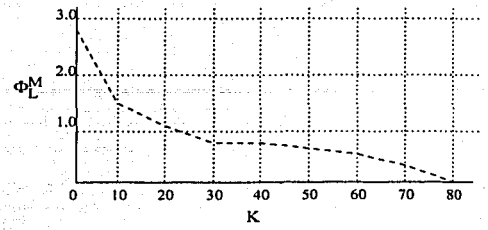


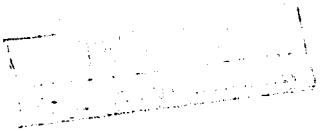
Figura 4.4: La función topográfica para un mapeo de un espacio de características rectangular a una cadena de 80 neuronas.

se habla entonces de un error topográfico [56].
Dado el vector de características $\mu \in R^m$, donde M es la dimensión del espacio de características, sea w_i el vector de pesos mas cercano a μ (el vector de la neurona ganadora para x) y w_j el segundo vector de pesos mas cercano a μ. Entonces, algunos objetos en R^m entre μ y w_j son mapeados a w_i en tanto que los restantes objetos entre μ y w_j son mapeados a w_j. Si las neuronas ganadoras n_i y n_j son adyacentes, el mapeo es localmente continuo; si no son adyacentes, existe una discontinuidad, llamada error topográfico.
El error topográfico ϵ_t para el mapeo en su totalidad es obtenido al sumar los errores topográficos locales para todos los vectores de características [56][91]:

$$\epsilon_t = \sum_{k=1}^{N} \eta(\mu_k) = \begin{cases} 1 & \text{si los vectores de pesos mas cercano y segundo mas cer-} \\ & \text{no representan neuronas que son no adyacentes} \\ 0 & \text{en otro caso} \end{cases}$$

4.5 Vecinos en la hiperesfera de radio r

Imaginemos ahora al objeto i en el espacio de características. Recordemos que este objeto es descrito por n características, por lo que el espacio es de dimensión n. A diferencia de la métrica expuesta en la subsección anterior, no queremos encontrar un número determinado de sus vecinos, pero si queremos encontrar todos los vecinos que se encuentren a una distancia r del objeto i, pudiendo incluso no contar con vecinos. Ahora, ese conjunto V de vecinos deberá ser mapeado por neuronas vecinas a la neurona que mapea a i para poder sustentar que el mapeo conserva la topología. La figura 4.5 muestra este mapeo.



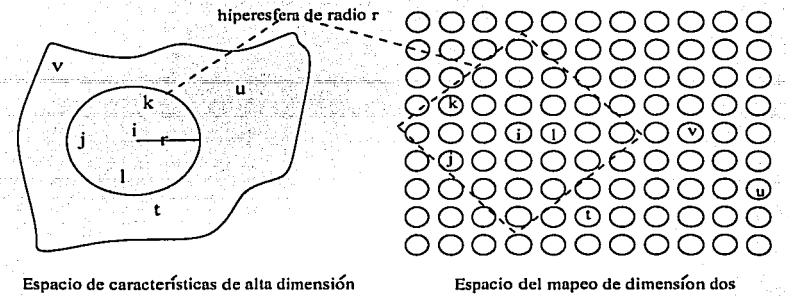


Figura 4.5: Los objetos que se encuentran a una distancia r del objeto i son mapeados por las neuronas ganadoras mas cercanas a la neurona que mapea al objeto i

En otras palabras, sí el objeto i en el espacio de características contiene, en la hiperesfera de radio r y centro en i a los objetos j, k y l, entonces, la neurona que mapea al objeto i, n_i, deberá contener, en la esfera de radio r a las neuronas que mapean a j, k y l, n_j, n_k y n_l.
Si los vecinos en la hiperesfera no se conservan en el espacio del mapeo, la topología se estará violando. Es importante notar que el orden de cercanía, al igual que con la métrica de confiabilidad, analizada en la siguiente sección, no es importante: si el objeto j es el vecino más cercano al objeto i en el espacio de características, pero en la malla del mapeo la neurona que representa j no es la neurona ganadora mas cercana a i, pero j está contenido en la esfera de radio r, la vecindad no se habrá violado. En la figura 4.6 el objeto más cercano a i en el espacio de características es el objeto k, pero en la malla de mapeo el más cercano es la neurona que mapea al objeto l

4.6 Confiabilidad

En el espacio de características, la *vecindad* de un objeto i se define como el conjunto i_k formado por los k objetos más cercanos a él. Si el conjunto i_k, o vecindad de i, se preserva en la malla de la red, es decir, en el espacio del mapeo, la topología se preserva. A esto se le conoce como confiabilidad del mapeo [49].
La figura 4.6 muestra a los k vecinos de i en el espacio de alta dimensión, y los k vecinos de la neurona que mapea al objeto i. Para i, la vecindad no se

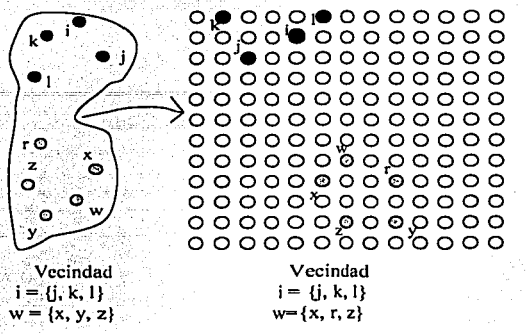


Figura 4.6: Vecindad en el espacio de características y en el espacio de mapeo. Los $k = 3$ vecinos de i, i_k si se preservan en tanto que los $k = 3$ vecinos de w, w_k no.

violó, en tanto que para el objeto w, si fue violada su vecindad.

Esta métrica recibe el nombre de confiabilidad porque cuantifica el número de veces que se viola la vecindad para todos los objetos. Formalmente, la métrica se define como:

$$C = \sum_{i=1}^{N} \sum_{j=1}^{K} (f(j, n_{ik}) + g(n_j, i_k)) \tag{4.1}$$

donde

$$\mathrm{f(j,\ n_{ik})} = \begin{cases} 0 & \text{si } j \in n_{ik} \\ \alpha & \text{en otro caso} \end{cases} \quad y \quad \mathrm{g(n_j, i_k)} = \begin{cases} 0 & \text{si } j \in n_{ik} \\ \beta & \text{en otro caso} \end{cases}$$

Recordemos que i_k es el conjunto de los k objetos más cercanos a i en el espacio de características y n_{ik} son las k neuronas ganadoras más cercanas a la neurona n. La función $f(.)$ identifica aquellos objetos j vecinos a i que no son mapeados a neuronas vecinas a la neurona que mapea a i, en tanto que la función $g(.)$ identifica aquellas neuronas en la vecindad de la neurona que mapea a i, que en realidad mapean objetos que no son vecinos de i en el espacio de características.

Dos son los errores que esta métrica cuantifica:

(a) Si un objeto j está en la vecindad de i, i_k, pero no es mapeado a una de las neuronas en la vecindad de la neurona n_i, que mapea a i, se suma una cantidad α al error total.
(b) Si un objeto m que no está en la vecindad de i es mapeado a la neurona n_m y esta neurona se encuentra en la vecindad de la neurona n_i, que mapea al objeto i, se suma al error total un cantidad β [91].

Un mapeo que preserva la topología debe mostrar, lo mejor posible, las relaciones de vecindad que ocurren en el espacio de características [56]. El error (a) indica que un objeto salió de la vecindad de otro objeto, en tanto que el error (b) indica que un objeto entró a la vecindad de otro. Venna señala que el error (b) es mas grave que el (a), por lo que $\beta \geq \alpha$ [91].
Entre mayor sea la C de la ecuación 4.1, menos se preserva la topología; esto es, el error en la confiabilidad aumenta.
Ahora, el parámetro k, el número de objetos que están en la vecindad de otro objeto, es un factor determinante. Si $k = |I|$, en donde I es el conjunto de objetos analizados, el error de la métrica sería igual a 0 ($C = 0$). Lo mismo ocurre si $k = 0$. Conforme k se aproxima a $I/2$, el error es máximo.
Veamos la explicación de la afirmación anterior. Sea $i \in I$ un objeto en el espacio multidimensional. Si $k = 0$, esto quiere decir que su vecindad, i_k, está compuesta únicamente por el objeto i, por lo que la neurona que mapea a i, n_i, es vecina de si misma, lo que no produce error ($C = 0$).
Veamos ahora el caso en el otro extremo, $k = |I|$. Esto quiere decir que la vecindad de i, i_k, está compuesta por todos los objetos en I, esto es, $i_k = I$. La neurona que mapea a i, n_i, tiene (como máximo, pues dos objetos pueden ser mapeados a una misma neurona) otras $|I - 1|$ neuronas vecinas, que son parte de la configuración ordenada de neuronas ganadoras. Puesto que la métrica penaliza cuando un objeto *entra* o *sale* de la vecindad en el espacio del mapeo, la penalización será $C = 0$, pues ningún objeto puede salir o entrar de la vecindad cuando ésta es máxima.
La confiabilidad del mapeo varía de acuerdo a k. La figura 4.7 muestra este evento. En el eje x se indica el valor del parámetro de vecindad k, en tanto que en el otro se muestra el error en la confiabilidad, C. Se puede apreciar que el error máximo para un mapeo dado se presenta alrededor de $k = |I|/2$.

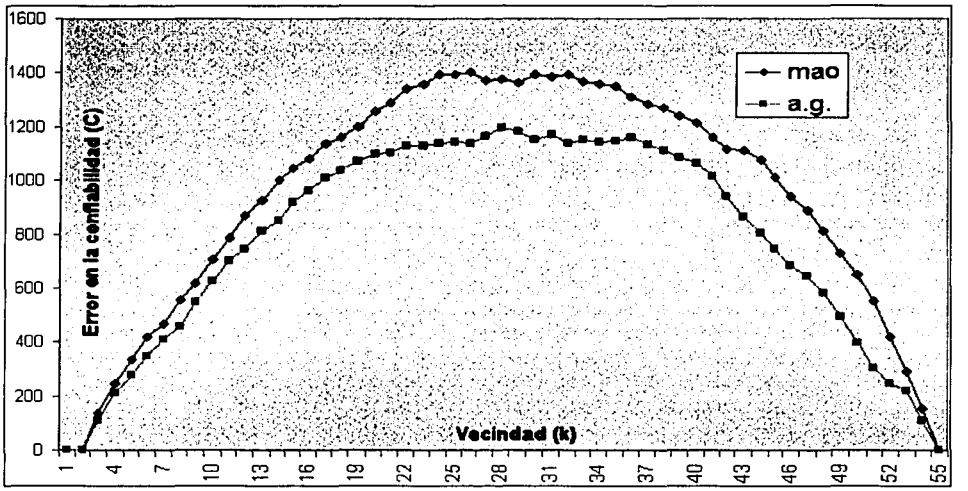


Figura 4.7: Error en la preservación de la topología debido al parámetro de vecindad k para dos mapeos autoorganizados para el uso de codones de los 54 organismos analizados. En color obscuro se muestra el error cuando se considera la totalidad de características (los 64 codones); en color claro se muestra el error para un mapeo para los mismos organismos, pero considerando solamente un subconjunto de características (ver Cap. 5). Se observa que para valores de k pequeños, el error es bajo y comienza a incrementarse una vez que el valor k llega a cierto punto, en donde comienza a disminuir hasta llegar a 0 para $k = 54$.

4.7 Elección de la métrica de preservación de topología

No existe una métrica aceptada como estándar para su uso en el mapeo autoorganizado [63]. De las métricas mostradas en las secciones anteriores, algunas muestran ventajas sobre otras, pero no existe una que analíticamente pueda ser considerada como la mejor [93, 63].
Para el problema del sesgo en el uso de codones se eligió la métrica de confiabilidad. Aunque su definición es más simple que las tres primeras métricas analizadas, su poderío es suficiente [51]. Los motivos de optar por esta métrica son de índole computacional y de comparación con mapeos de otro

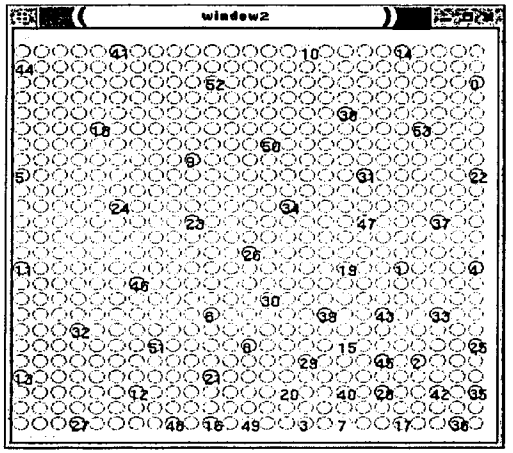


Figura 4.8: Mapeo para el uso de codones de los 54 organismos listados en la tabla 2.7

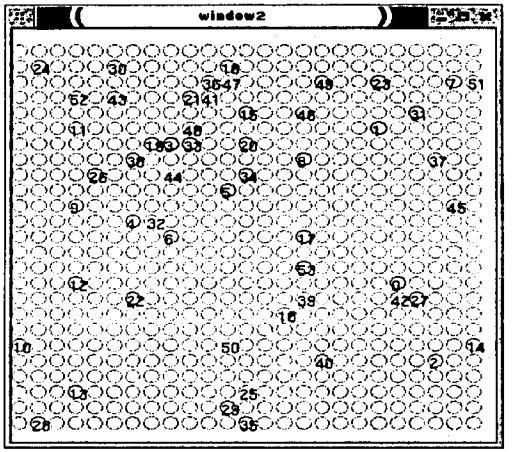


Figura 4.9: Mapeo para 54 vectores de dimensión 64 creados aleatoriamente.

tipo. En primer lugar, la métrica obtiene un escalar, fácilmente comparable, a diferencia de, por ejemplo, la función topográfica, pues resulta mas fácil comparar dos escalarares que dos funciones.
En segundo lugar, los métodos mostrados en las secciones anteriores trabajan directamente sobre el vector de pesos de las neuronas, lo que no les permite ser usados para cuantificar la preservación de la topología de mapeos de naturaleza diferente a aquella del mapeo autoorganizado, como el escalamiento multidimensional, pues en estos mapeos no existe el concepto de vector de pesos. En el capítulo seis, se habla de esos mapeos.
Hablaremos ahora del desempeño del mapeo. Un mapeo tendrá un mejor desempeño que otro si la violación de la topología en el primero es menor que en el segundo.
En las figuras 4.8-4.11 se muestran diversos mapeos. El mapeo mostrado en la figura 4.8 es un mapeo autoorganizado para los organismos analizados en este trabajo, el mostrado en la figura 4.9 es un mapeo para vectores aleatorios de dimensión 64 y el mapeo de la figura 4.10 es una configuración aleatoria de neuronas ganadoras (no se obtuvo por medio de ningún mapeo).

El mapeo de la figura 4.11, que muestra un desempeño mayor al mostrado por los otros, fue hecho para los mismos 54 organismos, con una salvedad: solamente fueron consideradas 53 de las 64 variables originales, es decir, el vector de características para cada organismo es de dimensión 53, en contraste con la dimensión original de 64. Se eliminaron variables que aparentemente

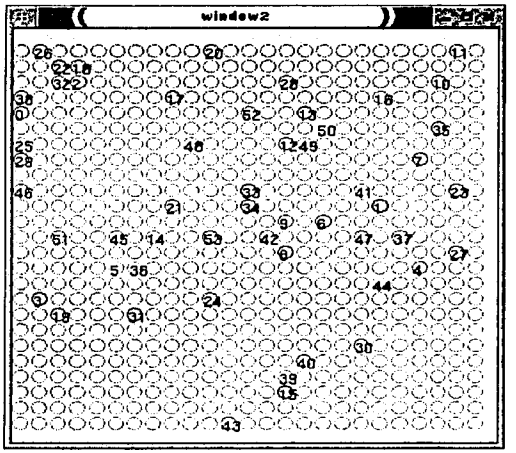


Figura 4.10: Mapeo para el uso de codones de los 54 organismos listados en la tabla 2.7

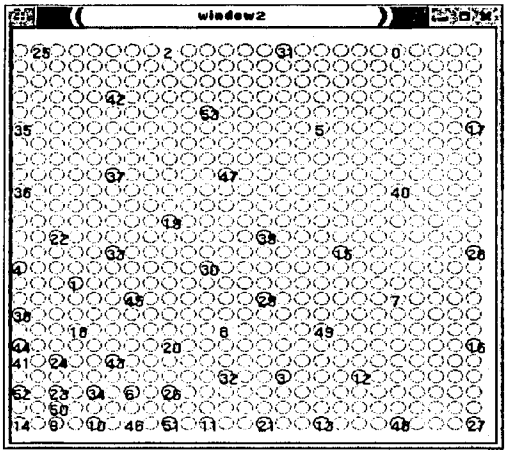


Figura 4.11: Mapeo para 54 vectores de dimensión 64 creados aleatoriamente.

no intervienen significativamente en la definición de la topología. El siguiente capítulo detallará la eliminación de variables de este tipo.

La figura 4.12 muestra el error en la confiabilidad como función del parámetro de vecindad k.

4.8 El parámetro de vecindad k

Una representación en baja dimensión de la distribución que presentan los objetos en un espacio de alta dimensión (el espacio de características), mostrará errores en la preservación de las vecindades (no conserva la topología en todos los casos). Algunas representaciones, como el mapeo autoorganizado, preservan adecuadamente las relaciones de vecindad para los objetos mas cercanos en el espacio de características, en tanto que la preservación se hace inexacta cuando se toma en cuenta no solamente a los objetos mas cercanos, sino a un número mayor de ellos [53]. La figura 4.13 muestra lo anterior.

En la figura anterior se observa que los tres primeros vecinos del objeto i, que son j, k, l, son mapeados a alguna de las tres neuronas más cercanas a la neurona que mapea a i. Se observa también que los objetos m, n, p, q son mapeados a neuronas que no representan necesariamente la relación de vecindad presente en el espacio de características: el objeto q es mapeado a una neurona más cercana que la neurona que mapea al objeto p, siendo que p se encuentra más cerca de i que q en el espacio de características.
Lo que se observa en el mapeo mostrado en la figura 4.13 es lo que se conoce

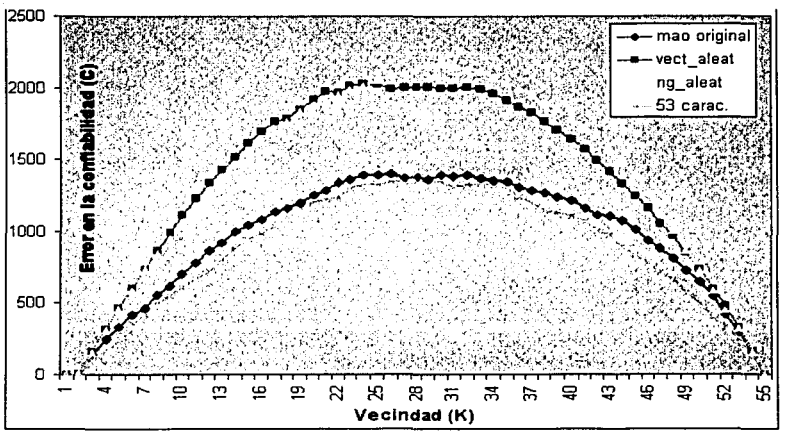


Figura 4.12: Error en la confiabilidad (C) como función de k para los mapeos mostrados en las figuras 4.8-11. Los mapeos de las figuras 4.8 y 4.11 (mao original y mapeo para 53 características en lugar de las 64 originales) muestran un desempeño mayor a los mapeos aleatorios $\forall k$.

como preservación de primeros vecinos: el mapeo autoorganizado preserva las relaciones de vecindad adecuadamente para los vecinos más cercanos. Por vecinos más cercanos nos referimos a aquellos entre el 5 y 10% del total de objetos [51]. Dicho de otra forma, el mapeo autoorganizado preserva las relaciones de vecindad en mayor medida para los k objetos más próximos, donde $k = 5 - 10\%$ del total de objetos. El rango del 5 al 10% para el cual el mapeo autoorganizado preserva adecuadamente la topología se debe a la naturaleza del algoritmo de aprendizaje, en el que se considera una vecindad decreciente [56].

Para valores de k mayores al 10%, el mapeo autoorganizado no preserva las relaciones de vecindad tan bien como lo hace para valores de k menores. Para el problema tratado en este trabajo, donde se analizan 54 organismos, lo esperable es que las relaciones de vecindad para 5 organismos sea conservada. Los 5 objetos más cercanos a cada organismo, (p.e. el perro) en el espacio de características, de dimensión 64, son la vaca, el cerdo, el caballo, el ratón y el humano. En el mapeo mostrados en la figura 4.1, las neuronas que mapean al perro tienen como vecinas a neuronas que mapean a los cinco organismos mencionados: la topología se ha preservado. En cambio, en el mapeo de la figura 4.2, las neuronas que mapean a los 5 vecinos del perro ya no se preservan: la topología no se ha preservado. Al menos en lo que al perro se refiere, el primer mapeo es mejor que el segundo para $k = 5$.

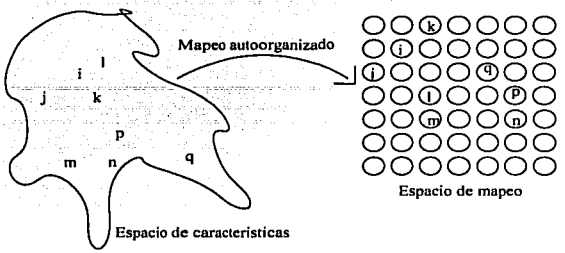


Figura 4.13: La topología se preserva para el objeto i y $k = 3$, pues j, k, l, los objetos más cercanos a i en el espacio original, son mapeados a las 3 neuronas mas próximas;sin embargo, para $k = 4$ ya no se preserva, pues p es uno de los 4 vecinos de i en el espacio de características, pero la neurona que lo mapea no es una de las 4 vecinas de la neurona i.

Los tres siguentes vecinos del perro en el espacio de características son el gallo, el pez *Danio rerio* y el gorila. Sin embargo, las neuronas que mapean a estos organismos no se encuentran necesariamente en las proximidades de la neurona que mapea al perro: la topología no se ha preservado cabalmente. Si una herramienta de visualización es adecuada para preservar únicamente las relaciones de vecindad para $\alpha \leq k \leq \beta$, para valores de α y β grandes, la herramienta no es adecuada para la visualización. Veamos el motivo de lo anterior.
Si las relaciones de vecindad se preservan para los k primeros vecinos, lo que se expresa es que las k primeras neuronas corresponden a los k primeros objetos, pero no en orden de *cercanía*. La neurona que mapea a la vaca, el objeto más cercano al perro, no es la neurona mas próxima a la neurona que mapea al perro, sino la segunda mas cercana. La mas próxima es la que mapea al caballo. Pero se habla de que el mapeo preserva las relaciones de vecindad para los primeros $5 - 10\%$ de los objetos, lo que parece una incongruencia. Sin embargo, la preservación de vecindades se refiere a los k objetos más cercanos como grupo, no individualmente.
Si deseamos cuantificar el número de errores para el vecino más cercano, nos estamos refiriendo a una $k = 1$. Este es el caso más rígido, en tanto que hablar de una preservación de topología para $k = N$, donde N es el número de objetos analizados, no cuantifica en absoluto la bondad del mapeo, pues la preservación de las relaciones de vecindades sería exacta.
Por tal motivo, una herramienta que preserva la topología solamente para valores de k grandes, no es necesariamente adecuada, pues es mas fácil preser-

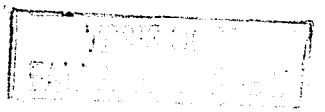
var las relaciones de vecindad para estos valores que para valores pequeños, siendo el caso extremo cuando $k = N$ [56, 52].
Puesto que queremos evaluar mapeos, incluso de naturaleza diferente, debemos tomar un valor k para ello. Como ya se mencionó, es mejor un mapeo que preserva la vecindad para valores de k pequeños, puesto esto es de mas ayuda para quien analiza la información.
Si un mapeo preserva la topología para $k = 15$, pero no lo hace para valores de k por debajo de dicho umbral, significa que si observamos un objeto i en dicho mapeo, al observar sus 15 vecinos, sabremos que en el espacio original esos 15 objetos son sus vecinos, aunque *no sabremos el orden de proximidad.*
Por otro lado, si un mapeo preserva las vecindades para valores de $k = 5$, pero no la preserva para k mayores, sabremos que si observamos a sus cinco vecinos en el espacio del mapeo, estos son en realidad sus vecinos en el espacio de características, también sin conocer el orden de proximidad.
Existen herramientas que preservan la topología adecuadamente para valores grandes de k, pero no para valores pequeños; existen también mapeos que que preservan mejor la topología para valores de k pequeños, pero no para valores de k grandes. La última conducta es mejor que la primera pues preservar vecinos para una k grande es más fácil que hacerlo para valores de k pequeños [52].
Por el motivo anteriormente expuesto, en adelante, habremos de considerar un valor de $k = 5$; esto es, cuantificaremos la preservación de la topología únicamente de los cinco primeros vecinos de cada objeto. Se eligió este valor pues el ser humano puede percibir las relaciones de distancia de hasta cinco o seis objetos simultáneamente [56]. Esta es la heurística sobre la que habremos de trabajar a fin de encontrar el mejor mapeo posible.
De esta forma, los mapeos presentados en este capítulo tienen un desempeño mostrado en la tabla 4.1.

Mapeo	Desempeño
Figura 4.1	372
Figura 4.2	417
Figura 4.8	393
Figura 4.9	606
Figura 4.10	598
Figura 4.11	348

Tabla 4.1: Desempeño de los mapeos mostrados en este capítulo. El desempeño se obtiene para el error en la confiabilidad para $k = 5$.

4.9 Discusión

Un mapeo autoorganizado es una herramienta que permite visualizar, en un espacio de baja dimensión (generalmente dos), la distribución de objetos multidimensionales. Al observar un mapeo autoorganizado, tendremos una idea aproximada de la topología que presentan los objetos en el espacio original, llamado espacio de características. Cuantificar el grado de preservación de la topología del espacio original que se encuentra reflejado en el espacio del mapeo, o malla, es necesario para identificar al mapeo con un mejor desempeño.
Diversas métricas para cuantificar la preservación de la topología fueron presentadas. Se eligió, para el problema del uso de codones, la métrica conocida como *confiabilidad*, pues los resultados que arroja se ajustan a las necesidades del problema planteado, además de permitir la comparación con mapeos de naturaleza distinta a los autoorganizados.
La métrica de error en la confiabilidad intenta cuantificar el error al contabilizar aquellos objetos que, siendo vecinos en el espacio de características, son mapeados a neuronas no vecinas en el espacio del mapeo.



Capítulo 5

Eliminación de características para el mapeo autoorganizado

5.1 Introducción

Los objetos multidimensionales presentan una cierta distribución en el espacio de características, que deseamos ver reflejada, en la medida de lo posible, en el espacio del mapeo generalmente de dimensión dos [8]. Si al eliminar un conjunto de características, el mapeo continua reflejando la topología del espacio de características original, ese conjunto de características no interviene significativamente en la definición de la topología definida en el espacio de características [56].
De manera más formal, sea Φ_M una topología definida sobre el conjunto de datos en el espacio de características de dimensión M. Si se elige un subconjunto de características $N \subset M$, de tal forma que el mapeo autoorganizado para los objetos en el espacio N presente una topología Φ_N *semejante* a Φ_M, entonces N es suficiente para describir la topología (relaciones de vecindad) de los objetos analizados.
Si el conjunto de características eliminadas no afecta el resultado (la distribución obtenida por el mapeo), puesto que dichas características no entran en la definición de topología, tampoco intervienen significativamente en el proceso del mapeo: La visualización de objetos multidimensionales que pretendemos mediante un mapeo autoorganizado no se ve afectada, pues las vecindades se preservan.
Al eliminar aquellas características que no intervienen en la definición de la topología se obtienen beneficios:

(a) El esfuerzo computacional se reduce.

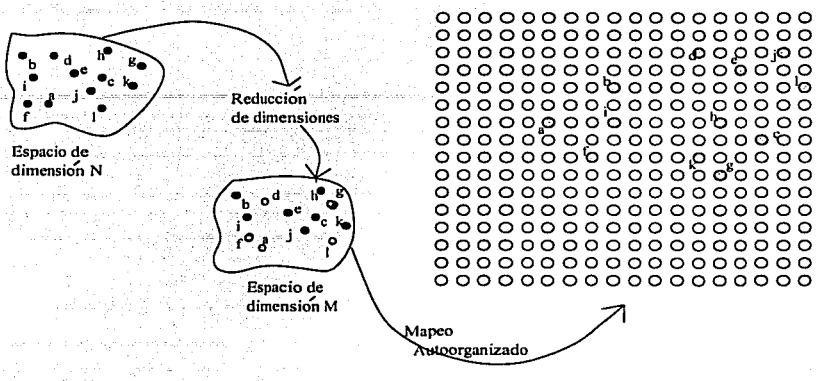
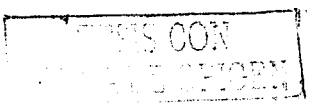


Figura 5.1: Eliminación de características sin pérdida de topología. El mapeo autoorganizado preserva la topología mostrada por los objetos en el espacio de dimensión M, que a su vez preserva la topología del espacio original, de dimensión N.

(b) El conjunto de características no eliminadas, al ser importantes para la topología definida, también lo serán para posibles explicaciones del fenómeno analizado.

En la figura 4.8 se puede observar que la neurona que mapea el uso de codones del perro se ubica a corta distancia de aquella que mapea el uso de codones del cerdo. En la figura 4.11, ambas neuronas continúan cercanas, aunque algunas de las características fueron eliminadas. Explicar esta relación de cercanía con base en las restantes características resulta más fácil al especialista.
La escencia de reducir las dimensiones puede percibirse en la figura 5.1. Cabe recordar que por topología nos estaremos refiriendo a las relaciones de vecindad entre los objetos.

Lo que queremos obtener es, a partir del espacio original de características de dimensión N y con una cierta topología Φ_N, un espacio de dimensión M, con $M < N$, y una topología Φ_M, de tal forma que Φ_M sea lo más parecida posible a Φ_N.
Encontrar un subconjunto de estas características que conserven las relaciones de vecindad entre los objetos lo mejor posible es un problema NP-



completo.
En las siguientes secciones serán analizados diversos métodos de eliminación de características, de tal forma que el mapeo represente lo mejor posible lo que ocurre en el espacio de características.

5.2 Reducción de la dimensión mediante matrices aleatorias

La esencia de la reducción de dimensión por matrices aleatorias es sustituir cada vector de características $x \in R^N$ por un vector $z \in R^M$, donde $|M| < |N|$.
Para cada vector x, se obtiene el vector z al multiplicarlo por una matriz aleatoria: $z = \mathbf{R}x$. $\mathbf{R}$ es una matriz aleatoria con N columnas y donde el número de renglones define la dimensión del vector z, M [50].
Una de las desventajas de este método es que, si bien disminuye la dimensión del vector de características, no identifica aquellas que son las más relevantes en la propia definición de la topología. Otra desventaja la constituye el hecho de que la dimensión del espacio original debe ser *considerablemente grande* (superior a 1000) [57]. La figura 5.2 muestra el mapeo autoorganizado del uso de codones de los 54 organismos referidos, donde la dimensión se ha reducido de 64 a 50. El desempeño de este mapeo para $k = 5$ es 597.

5.3 Eliminación por desviación estándar

Kohonen propone eliminar aquellas características con menor desviación estándar [61]. De ésta forma, las variables restantes continuarán presentando la misma topología, por lo que, finalmente, el mapeo preservará la topología tanto o *mas que antes*, para los objetos sobre la que fue definida.
La razón de eliminar las variables con desviación estándar pequeña es la siguiente. Imagine una variable para la cual todos los objetos tienen un mismo valor α. La desviación estándar para esta variable es 0, pues la dispersión de los valores es inexistente. Las relaciones de vecindad, en particular, y las topológicas, en general, no considerarán esta característica en su definición. La figura 5.3 muestra lo que ocurre cuando una variable tiene una desviación estándar cercana a 0.

En la figura 5.3a, se observa que la desviación estándar de la variable z es muy cercana a 0, por lo que podría eliminarse dicha variable, sin que se modifique la topología definida sobre el conjunto de objetos. Se observa esta eliminación en la figura 5.3b. La variables x y y no podrían ser eliminadas

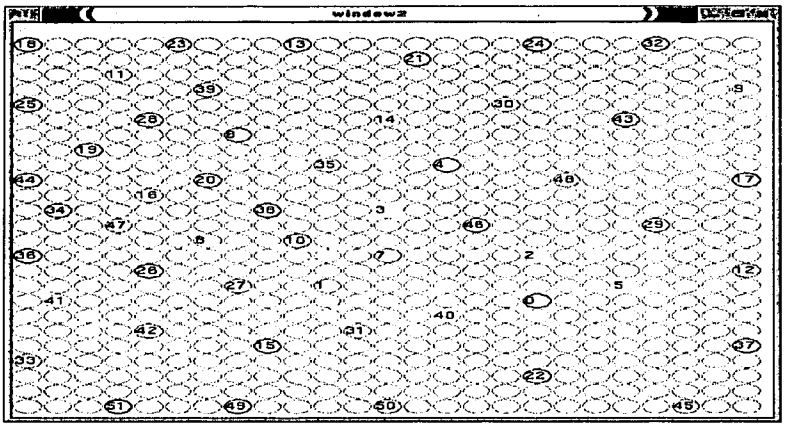


Figura 5.2: Mapeo para el conjunto de organismos analizados en donde el vector de dimensión 64 características de cada objeto ha sido sustituido por uno de dimensión 50, obtenido por medio de matrices aleatorias.

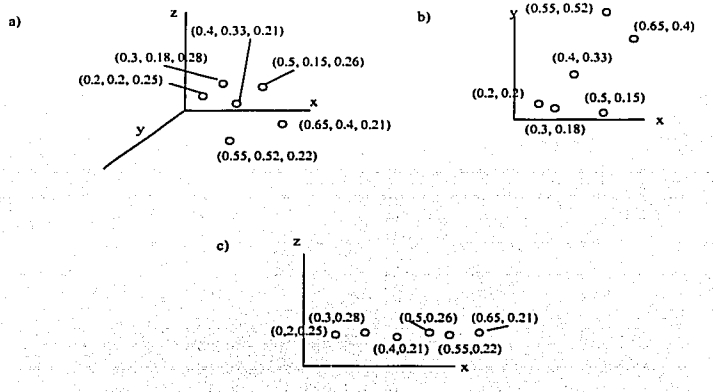


Figura 5.3: Eliminación de características con baja desviación estándar.

puesto que al hacerlo las relaciones de vecindad (la topología) no serían las mismas en el nuevo espacio. La figura 5.3c muestra lo que ocurre al eliminar la variable y: el vecino mas próximo del punto $i = (0.5, 0.15, 0.26)$ es $j = (0.4, 0.33, 0.21)$, en tanto que ahora el vecino mas próximo de i es ahora el punto $j = (0.55, 0.22)$, puesto que al eliminarse y, $i = (0.5, 0.26)$.
Para el problema del sesgo en el uso de codones, la desviación estándar para cada una de las 64 variables se muestra en la tabla 5.1. Se muestra en la tabla, en orden descendiente, el triplete y el aminoácido para el que codifica.

DS	Codón	Codón	DS	Codón	Codón	DS	Codón	Codón
0.214997418	42	AAA	0.088338386	26	CAG	0.061103994	31	CGG
0.190356567	19	CUG	0.087926938	57	GAC	0.060628864	20	CCU
0.152400715	40	AAU	0.087374292	33	AUC	0.060444256	9	UAC
0.143890531	2	UUA	0.082274447	47	AGG	0.058570818	35	AUG
0.141806445	59	GAG	0.080545251	46	AGA	0.058362805	5	UCC
0.140485343	58	GAA	0.078941355	23	CCG	0.058110984	39	ACG
0.138348259	53	GCC	0.076113472	21	CCC	0.056900062	54	GCA
0.136077068	32	AUU	0.076113309	37	ACC	0.056652363	41	AAC
0.135451054	18	CUA	0.074249781	45	AGC	0.056466068	25	CAC
0.133903614	34	AUA	0.074054905	38	ACA	0.055884677	15	UGG
0.129349133	61	GGC	0.0720688	62	GGA	0.053298167	7	UCG
0.121691221	27	CAG	0.071890028	1	UUC	0.051665713	28	CGU
0.120394554	55	GCG	0.071153043	52	GCU	0.048330492	44	AGU
0.118860429	0	UUU	0.068956485	6	UCA	0.047843166	13	UGC
0.118275881	51	GUG	0.067621237	60	GGU	0.046635195	14	UGA
0.114116875	56	GAU	0.067523633	4	UCU	0.044928455	12	UGU
0.107813679	43	AAG	0.067434405	3	UUG	0.03774358	24	CAU
0.099443349	29	CGC	0.064761639	49	GUC	0.03377279	30	CGA
0.092427011	48	GUU	0.064535788	16	CUU	0.01111337	10	UAA
0.091496503	8	UAU	0.063287007	36	ACU	0.005143231	11	UAG

Tabla 5.1: Desviación estándar del uso de codones de 54 organismos.

Si eliminamos la característica con la menor desviación estándar, triplete UAG, que es la secuencia de paro, y realizamos el mapeo autoorganizado de los objetos de dimensión 63 (64 - triplete), veremos que la confiabilidad de dicho mapeo es incluso mayor que para el mapeo de las 64 características originales. La figura 5.4 muestra el mapeo autoorganizado para 63 características.

Si eliminamos no una, sino seis características, aquellas con menor des-

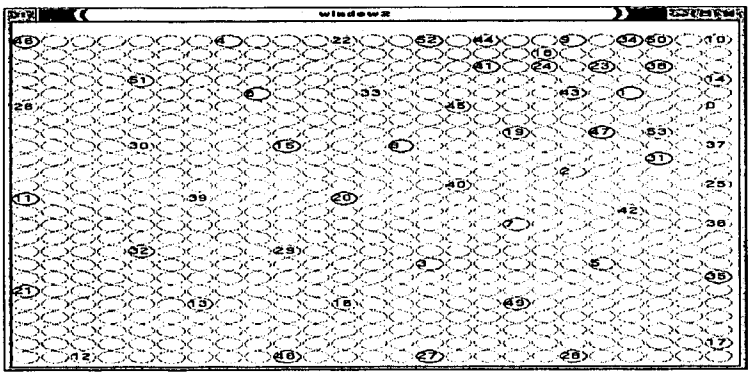


Figura 5.4: Mapeo autoorganizado para organismos con 63 características (se ha eliminado el triplete UAG).

viación estándar, obtendremos el mapeo mostrado en la figura 5.5.

Al eliminar un subconjunto de variables, el mapeo muestra una variación en su desempeño. Al mapear del espacio de características de dimensión 5 al espacio del mapeo (se eliminan 59 de las 64 características originales), el mapeo ya es notablemente peor que el mapeo original. La figura 5.6 muestra el anterior mapeo.

La tabla 5.2 muestra el desempeño de diversos mapeos para subconjuntos de las 64 variables originales. Se muestra el desempeño para los primeros cinco vecinos ($k = 5$).

Si la eliminación de un subconjunto de características resulta en un mapeo con un desempeño η, eliminar otra característica adicional, puede resultar en un mapeo con mejor o peor desempeño como se muestra en la tabla anterior. Eliminar un subconjunto s de variables con baja desviación estándar puede llevar a que el espacio, de menor dimensión que el espacio original, ya no preserve las relaciones de vecindad.

Veamos de nueva cuenta la figura 5.3. Al eliminar las variables x, z, que aunque presentan una desviación estándar baja, se traduce en la pérdida de las relaciones de vecindad previas, como se muestra en la figura 5.3c. Sin embargo, si se eliminan x y y, la relación de vecindades se mantiene, como

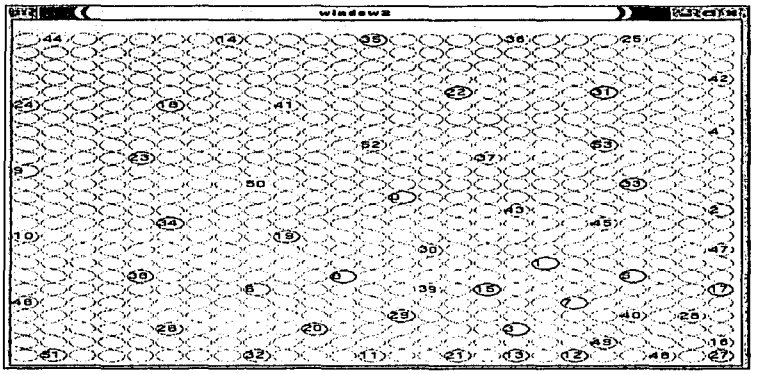


Figura 5.5: Mapeo autoorganizado para organismos con 58 características (se han descartado las seis características con menor desviación estándar). El error en la confiabilidad del mapeo es $C = 369$ para los primeros cinco vecinos ($k = 5$).

lo muestra la figura 5.3b.

Lo anterior pone de manifiesto una de las desventajas de eliminar las variables con baja desviación estándar: si se elimina un subconjunto de características, incluso si son las de más baja desviación estándar, la topología puede perderse.

Cabe mencionar que se obtuvo la matriz de correlación para los 64 tripletes, eliminando alguno de ellos cuando su correlación con otro era elevada (superior a 0.9). Por ejemplo, los tripletes 14 y 18 (UGA y CUA) tienen una correlación de 0.927. Al eliminar cualquiera de los dos, el desempeño del mapeo se ve deteriorado.

5.4 Eliminación por preservación de topología

Si deseamos eliminar aquellas características que no intervienen significativamente en el proceso del mapeo autoorganizado, ¿Por qué entonces no eliminar aquellas cuya remoción no modifique la topología que existe en el espacio original de características?

Al eliminar un subconjunto de característica de tal forma que no se modifique

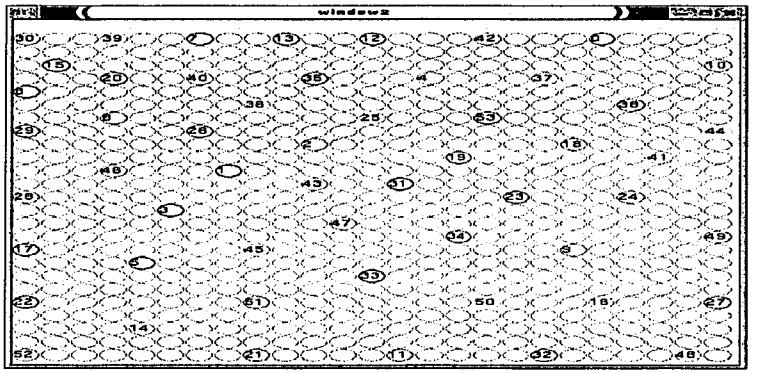


Figura 5.6: Mapeo Autoorganizado para organismos con 5 características (se han descartado 59 características). El desempeño es $C = 447$ para $k = 5$.

la topología original, aunque posiblemente si las distancias, se está simplificando el espacio. Si al mapear los objetos de este espacio simplificado mediante un mapeo autoorganizado la topología del espacio original (no el simplificado) se preserva, habremos encontrado una metodología de eliminación de características poco relevantes.
La afirmación anterior, que trataremos de probar en lo que resta de esta sección, puede ser modificada, no sin sacrificar confiabilidad. Esto es:

Al eliminar un subconjunto de características de tal forma que la topología presente en el espacio original se modifica lo menos posible, aunque las distancias no se preserven tendremos una metodología capaz de identificar aquellas características que no son significativas para la definición de la topología definida en el espacio de características y que queremos preservar en el espacio del mapeo.

De manera más formal, lo que el párrafo anterior explica es lo siguiente. Sea Φ_M una topología definida sobre los objetos analizados en el espacio de M características, esto es, de dimensión M. Si se elige un subconjunto $N \subset M$ de tal forma que la topología sobre los mismos objetos, pero considerando solamente las características en N, llamada Φ_N, y se observa que

Dimensión	Características consideradas	Desempeño
63	Todas menos 11	357
61	Todas menos 10, 11, 30	402
55	Las 55 primeras de la tabla 5.1	369
54	Las 54 primeras de la tabla 5.1	375
49	Las 49 primeras de la tabla 5.1	372
44	Las 44 primeras de la tabla 5.1	348
25	Las 25 primeras de la tabla 5.1	360
18	Las 18 primeras de la tabla 5.1	369
13	Las 13 primeras de la tabla 5.1	363
10	Las 10 primeras de la tabla 5.1	423
8	Las 8 primeras de la tabla 5.1	408
5	42, 19, 40, 2, 59	447
4	42, 19, 40, 2	468
2	42, 19	510

Tabla 5.2: Desempeño de diversos mapeos desde un espacio formado por variables seleccionadas de acuerdo a su desviación estándar.

Φ_M es *semejante* a Φ_N, habremos encontrado un conjunto de características que intervienen directamente en la definición de la topología.
Si una característica es eliminada, queremos cuantificar el impacto que esto tiene en la topología definida por el conjunto de objetos multidimensionales. La figura 5.3a muestra las relaciones de vecindades para un conjunto de objetos en el espacio n-dimensional. La figura 5.3b muestra las relaciones de vecindad si un subconjunto de variables (dimensiones) es eliminada. Puede verse que la topología no se preserva totalmente, pues los vecinos de b se han modificado. La figura 5.3c muestra las relaciones de vecindad si un subconjunto de características diferente al actual es eliminado.
Se observa que la eliminación del primer subconjunto no genera tantas violaciones a las relaciones de vecindad como el segundo subconjunto de características. De esta forma, diremos que el primer subconjunto será un mejor candidato para constituir nuestro nuevo espacio de características, de menor dimensión que el original, y a partir del cual mapearemos al espacio de baja dimensión donde visualizaremos las relaciones de los objetos que pretendemos analizar.
Definimos un error como una violación a la topología, (un incumplimiento de las relaciones de vecindad). Si en el espacio original de dimensión n cada objeto i tiene en su vecindad un conjunto de objetos i_n y al eliminar un subconjunto de características notamos que en la vecindad de i en el nuevo

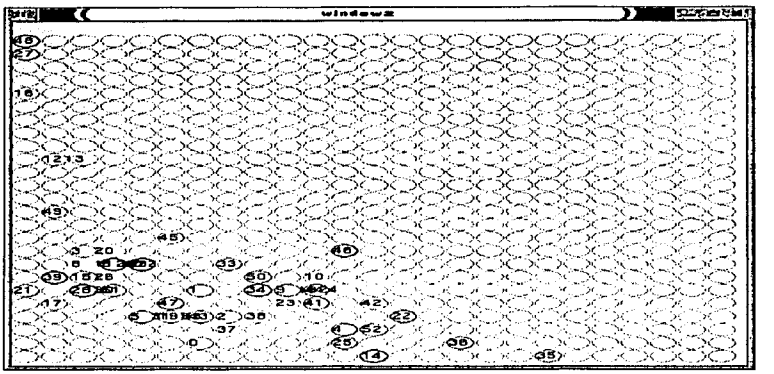


Figura 5.7: Distribución de los 54 organismos en el espacio de dimensión dos formado por las características 19 y 42, que son las que muestran la mas alta desviación estándar. El desempeño (error en la confiabilidad) es $C = 399$ para los cinco primeros vecinos ($k = 5$).

espacio se encuentra el conjunto de objetos i_m, el cual es diferente a i_n, diremos que el nuevo espacio tiene tantos errores como objetos que están en i_n no están en i_m.

Una característica α es menos importante para la topología definida para un conjunto de datos que otra característica β si al eliminar α se tienen menos errores en la preservación de la topología que cuando se elimina β.
Para el problema del sesgo en el uso de codones, se muestra en la tabla 5.3 el error que se obtiene al eliminar 63 de las 64 características que definen el uso de codones de cada organismo, dejando únicamente la característica señalada.
Para cuantificar el error, se emplea la ecuación 4.1. Esto es, para cada objeto i en el espacio de dimensión 64, se identifican al conjunto de los $k = 4$ objetos más próximos, i_k. A continuación, en el espacio definido por el subconjunto de las 64 variables, se identifica al conjunto de los $k = 4$ objetos más próximos a i, llamados i_j. Si todos los objetos $p \in i_k$ se encuentran también en i_j, esto es, $\forall p$ con $p \in i_k$ y $p \in i_j$, entonces el error en la confiabilidad para ese objeto es 0. Si un objeto se encuentra en i_k pero no en i_j, se incrementa el error en uno.

Si dos subconjuntos de variables a y b forman espacios en donde el error en la confiabilidad de éstos es a_e y b_e, con $a_e < b_e$, entonces el mapeo del espacio a presentará un mejor desempeño que el mapeo del espacio b.
El primer caso de la tabla 5.3 muestra un espacio unidimensional definido por el codón 19 (CUG). Se observa que el error en la confiabilidad para este espacio es de 1256, en tanto que el error en la confiabilidad para el mapeo del espacio definido únicamente por el triplete 11 (UAG) es 3497.
Si consideramos ahora subconjuntos de 63 variables, esto es, eliminamos solamente una de ellas, obtenemos los resultados mostrados en la tabla 5.4. Los espacios definidos en este caso son de dimensión 63 y se muestra el error en la confiabilidad al eliminar al triplete indicado. En un espacio de dimensión 63, donde el codón 10 (UAA) ha sido eliminado, el error en la confiabilidad es de 0, lo que indica que este no interviene en nuestra definición de topología. En cambio, si eliminamos la variable 58 (GAA), nos encontraremos con que el error en la confiabilidad es de 47.
Es importante notar la semejanza entre estas tablas y la tabla 5.1. Diversas variables se encuentran en la misma región, aunque no necesariamente en la misma posición. Por ejemplo, los tripletes 10 y 11 y 19 y 42 se encuentran respectivamente, al final y al principio en ambas tablas.
Al eliminar, por ejemplo, las cinco peores características, el mapeo de las características restantes identificadas por desviación estándar es mejor que el mapeo de las características restantes identificadas por la métrica actual. Sin embargo, al considerar únicamente las cinco mejores características, el resultado se invierte: El mapeo de las características indicadas por la métrica de preservación de topología es, no solamente mejor que el mapeo de las cinco características indicadas por desviación estándar, sino que es casi tan bueno como el mapeo de las 64 características originales. La figura 5.9 muestra el mapeo del espacio de dimensión 5, formado por las 5 características mas importantes (CUG, UUA, UAU, AAU, GAC).

Al igual que hicimos con la desviación estándar, eliminamos no una, sino un conjunto de variables y mapeamos de ese espacio al espacio del mapeo, bidimensional. La tabla 5.4 muestra el desempeño del mapeo que va de la dimensión indicada al espacio del mapeo.

De nuevo, tal y como ocurre con la eliminación de características por desviación estándar baja, se presenta la situación descrita en la figura 5.3: eliminar características por su buen desempeño *individual* puede resultar en la falta de preservación de las relaciones de vecindad.
Si queremos encontrar un subconjunto c de las características originales, D, de tal forma que la topología sea lo más semejante en ambos espacios y para el mismo conjunto de objetos, debemos realizar una búsqueda en el espacio de la dimensión de D. Una búsqueda exhaustiva no es viable, por lo que

Error	#Codón	Codón	Error	#Codón	Codón	Error	#Codón	Codón
1256	19	CUG	2061	62	GCA	2538	22	CCA
1336	2	UUA	2091	1	UUC	2567	41	AAC
1534	8	UAU	2110	48	GUU	2575	3	UUG
1613	40	AAU	2123	21	CCC	2632	7	UCG
1669	57	GAC	2124	6	UCA	2636	4	UCU
1697	0	UUU	2141	26	CAA	2689	16	CUU
1744	59	GAG	2149	25	CAC	2690	54	GCA
1748	61	GGC	2149	33	AUC	2719	60	GGU
1840	27	CAG	2158	31	CGG	2730	39	ACG
1843	55	GCG	2167	42	AAA	2744	20	CCU
1848	53	GCC	2259	44	AGU	2797	63	GGG
1875	32	AUU	2270	23	CCG	2807	18	CUA
1886	29	CGC	2377	34	AUA	2828	15	UGG
1897	38	ACA	2400	50	GUA	2831	28	CGU
1903	49	GUC	2426	47	AGG	3004	12	UGU
1909	51	GUG	2426	52	GCU	3079	10	UAA
1913	45	AGC	2445	58	GAA	3134	35	AUG
1916	37	ACC	2448	13	UGC	3358	14	UGA
1960	17	CUC	2466	24	CAU	3497	30	CGA
2034	56	GAU	2467	46	AGA	3497	11	UAG
2050	36	ACU	2515	9	UAC			
2056	5	UCC	2523	43	AAG			

Tabla 5.3: Preservación de la topología en espacios unidimensionales definidos por el codón señalado. Se muestra el número de errores definido en función del total de objetos que se encuentran en una vecindad $k = 4$ en el espacio original pero que no se encuentran en esa misma vecindad en el espacio unidimensional.

proponemos una búsqueda realizada por un algoritmo genético. La siguiente sección detalla esta búsqueda.

5.5 Algoritmos genéticos para la identificación de características que preserven la topología del espacio original

Sea D un conjunto de características que describen a un conjunto i de objetos, sobre el que se ha definido una topología, Φ_i. Sea $c \subset D$ un subconjunto de características en donde el conjunto de objetos i presenta ahora una topo-

Error	#Codón	Codón	Error	#Codón	Codón	Error	#Codón	Codón
0	10	UAA	12	18	CUA	21	3	UUG
0	11	UAG	12	30	CGC	21	33	AUC
0	35	AUG	13	45	AGC	21	39	ACG
0	5	UCC	13	46	AGA	21	4	UCU
4	14	UGC	13	51	GUG	21	59	GAG
4	28	CGU	13	53	GCC	21	9	UAC
4	32	AUU	16	13	UGC	24	26	CGC
4	36	ACU	16	16	CUU	24	62	GGC
4	38	ACA	16	22	CCA	25	19	CUG
8	12	UGU	16	34	AUA	25	27	CAG
8	17	CUC	16	50	GUA	25	37	AAC
8	24	CGU	16	8	UAU	27	55	GCG
8	31	CGG	17	21	CCC	28	61	GGC
8	44	AGU	17	25	CGC	29	2	UUA
8	47	AGG	17	41	AAC	32	56	GAU
8	54	GCA	17	7	UCG	33	42	AAA
8	57	GAC	20	40	AAU	34	20	CCU
8	6	UCA	20	48	GUU	34	60	GGU
8	63	GGG	20	49	GUC	36	43	AAG
9	15	UGG	20	52	GCU	47	58	GAA
11	29	CGC	21	0	UUU			
12	1	UUC	21	23	CCG			

Tabla 5.4: Preservación de la topología en espacios de dimensión 63 definidos al eliminar el codón listado. Se muestra el número de errores definido en función del total de objetos que se encuentran en una vecindad $k = 4$ en el espacio original pero que no se encuentran en esa misma vecindad en el espacio unidimensional.

logía Φ_j. Queremos encontrar c de tal forma que las diferencias entre ambas topologías sea mínima, esto es, $d(\Phi_i, \Phi_j) < \epsilon$.
Para encontrar c, es necesario considerar el conjunto potencia P de D y comparar la topología de cada conjunto de P con la topología original. La cardinalidad del conjunto potencia de un conjunto de cardinalidad c es 2^c. Para el problema del sesgo en el uso de codones, el número de características es 64, por lo que el conjunto potencia tendrá 2^{64} subconjuntos, lo que hace inviable una búsqueda exhaustiva.
La función objetivo que se pretende minimizar por medio del algoritmo genético es la siguiente:

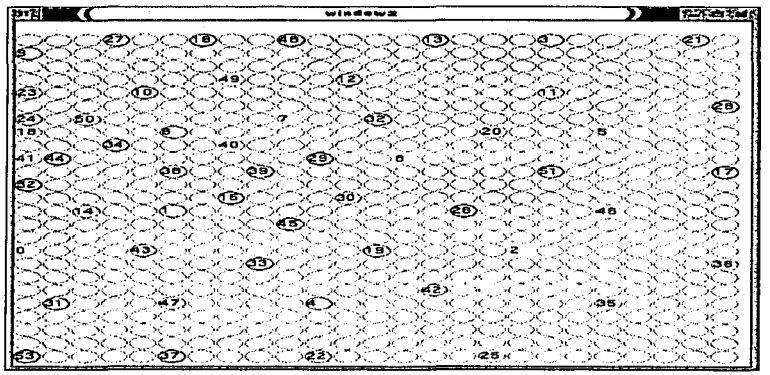


Figura 5.8: Mapeo Autoorganizado para organismos con 58 características (se han descartado las seis características con menor influencia en la preservación de la topología. Ver tabla 5.3). El desempeño del mapeo es $C = 375$ para $k = 5$

$$F_{obj} = \sum_{i=1}^{54} |G(i,k) - H(i,k)| \tag{5.1}$$

Donde $G(i,k)$ es el conjunto de los k primeros vecinos de i en el espacio de características original de dimensión 64 y $H(i,k)$ es el conjunto de los k primeros vecinos en el subespacio propuesto como un individuo por el algoritmo genético.

Para realizar búsquedas en espacios de alta dimensión, los algoritmos genéticos son una herramienta muy útil. Un algoritmo genético es un plan de búsqueda basado en el concepto de evolución biológica [42]. En éste, las posibles soluciones son codificadas en *individuos*, pertenecientes a una población, que serán capaces de intercambiar información contenida en sus genes. Los algoritmos genéticos han sido aplicados en diversos problemas y disciplinas. [24] los ha aplicado para predecir la estructura secundaria de las proteías con base en la estructura primaria (ver sección 2.5); [38] los ha utilizado para encontrar conductas adaptativas para predadores y presas que interactúan en un ambiente dinámico; Heijliger ha atacado el problema de asignaciones por medio de esta técnica [39].

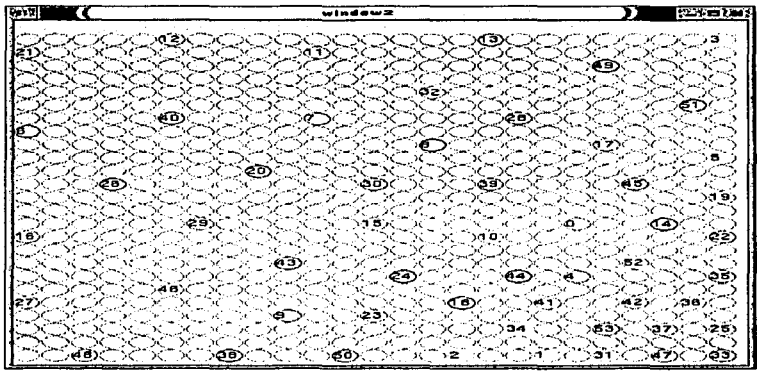


Figura 5.9: Mapeo Autoorganizado para organismos con 5 características (aquellas con mayor preservación individual de la topología. Ver tabla 5.4). El desempeño del mapeo es $C = 429$ para $k = 5$

Para el problema del sesgo en el uso de codones, en donde se tienen 64 dimensiones, los individuos cuentan con 64 genes. El gen 0 indica la presencia o ausencia de la característica 0 (triplete UUU), el gen 1 establece la ausencia o presencia de la característica 1 (UUC), etcétera. La tabla 5.5 muestra un grupo de individuos. Uno de ellos codifica un espacio de dimensión 13, pues solamente 13 genes muestran el *alelo* 1. Otro individuo en la misma tabla codifica un espacio de dimensión 54, donde no son considerados diez tripletes.
Se muestra a continuación el algoritmo genético clásico:

1. Crear población inicial.
2. Evaluar población inicial.
3. Seleccionar copias de individuos en *función* de su desempeño.
4. Cruzamiento.
5. Mutación.
6. Evaluar individuos.
7. Si aún no se cumple el criterio de terminación, ir a 3.

Dimensión	Características consideradas	Desempeño
63	Todas menos 11	357
61	Todas menos 30, 11, 14	354
55	Las 55 primeras de la tabla 5.3	399
54	Las 58 primeras de la tabla 5.3	375
49	Las 49 primeras de la tabla 5.3	387
44	Las 44 primeras de la tabla 5.3	357
25	Las 25 primeras de la tabla 5.3	384
18	Las 18 primeras de la tabla 5.3	375
13	Las 13 primeras de la tabla 5.3	411
8	Las 8 primeras de la tabla 5.3	399
5	42, 19, 40, 2, 59	429
4	19, 2, 8, 40	453
2	19, 2	468

Tabla 5.5: Desempeño de diversos mapeos desde un espacio definido por las variables elegidas por preservación de topología para los organismos analizados.

Cada individuo debe ser *evaluado*. Para ello, cabe recordar que cada objeto (organismo, en el caso que nos concierne) es descrito por un vector de características de dimensión D, donde D es el número de variables. (64 para el problema aqui presentado). Lo que el algoritmo genético pretende encontrar es un subconjunto $c \subset D$ de variables de tal forma que la topología definida por los objetos en el espacio de dimensión D, se preserve en el espacio de dimensión c.
Si un individuo, como el penúltimo individuo de la tabla 5.5, se encuentra en un espacio de dimensión 9, el vector de características que describe al conjunto de objetos será ahora de dimensión 9, y las características que forman dicho espacio son únicamente aquellas que se indican en el individuo. Es necesario ahora verificar si la topología que se definió para el espacio original, Φ, se conserva en el espacio de dimensión 9. Recordemos que por topología nos referimos, en particular, a las relaciones de vecindad: Si los objetos vecinos en el espacio de caracterísicas lo continuan siendo en el espacio del mapeo, la topología se habrá preservado.
Para evaluar una solución o individuo, es necesario obtener las relaciones de vecindad para cada objeto en el nuevo espacio y comparar dichas vecindades con las vecindades para el mismo objeto en el espacio original. Para ello, se calcula la distancia entre cada organismo y los restantes 53 para a continuación, elegir a los k más próximos. La distancia utilizada fue la de *Manhattan*.
La función objetivo es la mostrada en la ecuación 5.1, que es, en escencia,

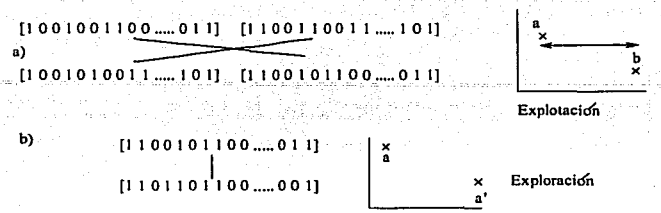


Figura 5.10: Operadores genéticos. En la figura a) se muestra el cruzamiento. Dos individuos elegidos aleatoriamente dan lugar a dos hijos. Para ello, un hijo hereda la información de uno de los padres hasta cierto gen, elegido aleatoriamente, llamado punto de cruza. De ese gen en adelante, la información la recibirá del otro padre. Con el otro hijo sucede algo semejante, solo que con el orden de los padres invertido. Para el caso mostrado, el punto de cruza es el gen seis. b) La mutación es el cambio de un gen por alguno de sus alelos (posibles valores). Uno o mas genes pueden sufrir el cambio. En el ejemplo, dos genes (el cuarto y el penúltimo) fueron mutados. La cruza se interpreta como una *explotación* de una cierta zona en el espacio de búsqueda, en tanto que la mutación es vista como una exploración, pues la búsqueda se redirige a otra región del espacio de búsqueda. [41]

un caso especial de la ecuación 4.1. Recordemos que esta función cuantifica el número de individuos que siendo vecinos de otro en el espacio orginal, ya no lo son en el nuevo espacio.
El algoritmo genético realiza la búsqueda mediante dos operadores: *cruzamiento* y *mutación*. El primer operador se define como la obtención de nuevos individuos con base en información (genes) de los padres. El segundo operador se explica como un redireccionamiento en el espacio de búsqueda y tiende a evitar que el algoritmo se estanque en soluciones localmente óptimas. La figura 5.10 muestra ambos operadores y una intrepretación de la función de cada operador. El cruzamiento explota una cierta región del espacio en tanto que la mutación explora otras regiones del espacio.

Algunos resultados obtenidos con el algoritmo genético con elitismo para el problema del sesgo en el uso de codones se muestran en la tabla 5.6. Se recurrió a una selección proporcional por ruleta, con una probabilidad de cruzamiento de 0.95 y una probabilidad de mutación de 0.05

Las características que el algoritmo indentificó como las adecuadas se muestran en la primer columna, la segunda columna muestra el número de dichas características (la dimensión del espacio propuesto por el algoritmo), la tercera columna muestra la diferencia entre la topología original y la topo-

logía presente en el nuevo espacio y la cuarta columna muestra el desempeño del mapeo autoorganizado de la dimensión mostrada en la segunda columna a la dimensión del espacio del mapeo.

Puede observarse que, en general, y tal como se esperaba, entre menor sea la diferencia entre la topología del espacio original y la topología del espacio codificado en el individuo, el desempeño del mapeo es mejor. En la sección 5.7 se profundiza sobre esta relación.

Características	Dimensión	Dif. en topología	Error en confiabilidad
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 31, 33, 34, 35, 36, 37, 38, 40, 42, 43, 44, 45, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63	53	61	339
1 2 3 4 5 6 7 8 9 11 12 13 14 19 21 23 27 28 29 31 33 34 35 38 39 40 41 42 44 46 47 48 49 50 51 52 53 55 56 57 61 62	42	151	354
14 30 33 41 2 58 45 20 48 31 4 16 1 21 29 23 55 11 6 43 24 40 32 22 27 49 0 34 44 42 54 3 12 56 59	35	66	345
10 37 52 50 9 11 2 48 59 1 43 29 57 54 19 4 8 55 44 41 18 24 58 26 40 45 13 30 31 15 20 23 5 53 17	35	107	360
53 9 25 22 11 57 6 44 18 17 48 42 52 29 40 54 23 34 58 50 7 5 20 32 27 2 13 45 10 4 55 59 49 41 0 24 43 37 47 39	40	77	363
1 42 43 56 25 14 51 50 59 24 31 27 8 21 5 41 30 19 49 22 45 7 4 53 39 38 29 11 16 46 40 20 2 26 0 13 32 35 33 15 55 37 10 34 58 54 57 52 18 28	50	61	351
	64 (Todas)	0	330 (menor 309, mayor 367)

Tabla 5.6: Soluciones encontradas por medio de un algoritmo genético. Se muestran los identificadores para los tripletes considerados, la dimensión (número de tripletes considerados), las diferencias en las relaciones de vecindad entre el espacio original de 64 codones y el espacio propuesto y el error en la confiabilidad del mapeo obtenido desde el espacio identificado por el algoritmo. Se muestra también el desempeño promedio del mapeo desde el espacio original (ver sección 5.8)

Es importante recordar que, aunque no existe una definición exacta de preservación de topología, en este trabajo estamos suponiendo que la misma se refiera la preservación de las relaciones de vecindad. Los casos en que el desempeño del mapeo autoorganizado, o error en la confiabilidad, no coincida con el error en la topología (columnas 4 y 3, respectivamente), puede ser indicativo de que la preservación de topología puede no ser la preservación

de vecindades.

5.6 Templado simulado

El templado simulado (simulated annealing) es un algoritmo general de optimización. Tiene su origen en la metalurgia, donde los metales siguen un esquema de enfriamiento, lo que les da una mayor resistencia (menores imperfecciones) que aquellos que se enfrian rápidamente. Aquel que se enfría gradualmente, siguiendo algún esquema de enfriamiento, se encontrará en un estado de *menor energía* que aquel que se enfrió sin ningún esquema [22].
Consideremos el problema de optimizar una función $f(x_1, ..., x_n)$. Sin pérdida de generalidad podemos suponer que $f \geq 0$. f es la energía que el sistema presenta cuando se encuentra en un estado particular dictado por el vector de variables. Si el sistema se encuentra en un estado con energía r, intentará moverse a algún otro estado de tal forma que la energía en el nuevo estado sea menor a la energía en el estado anterior. Esto podría guiar al sistema a enfrascarse en regiones con mínimos locales, pues las *barreras de alta energía* podrían impedirle desplazarse al óptimo global.
Consideremos ahora que el desplazamiento a un nuevo estado no será solamente considerando la diferencia de energías. Para ello, definimos una nueva variable T, la *temperatura* del sistema. El sistema se moverá de un estado con energía r y un valor determinado de las variables, $\overline{x}$, a un nuevo estado que presenta una configuración de variables $\overline{y}$ con una energía s, con una cierta *probabilidad*. Esta probabilidad está dictada por la distribución Boltzmann-Gibbs y es la siguiente:

$$P(\overline{y}) = e^{-s/kT}/Z \tag{5.2}$$

Donde s es la energía del estado, T es la temperatura del sistema y k y Z son constantes que se calculan para cada aplicación.
La distribución Boltzmann-Gibbs está dominada, a bajas temperaturas, por los estados de menor energía, que resultan ser los más probables; sin embargo, cuando la temperatura del sistema T es alta, la probabilidad de visitar un estado con mayor energía es grande en comparación a su probabilidad si la temperatura fuera menor [11].
El esquema de enfriamiento (la reducción de T) es fundamental para el correcto funcionamiento del algoritmo. Si se propone un esquema donde la temperatura T en el tiempo t esté dada por:

$$T(t) = K/log(t) \tag{5.3}$$

el algoritmo converge con una muy alta probabilidad a uno de los óptimos del sistema para algún valor constante de K. Este enfriamiento logarítmico tiene una desventaja: una fracción sifnificativa del total de estados tiene que ser visitada, lo que explica que el algoritmo encuentre al óptimo, pues, escencialmente, está realizando una búsqueda exhaustiva [35]. Un esquema de enfriamiento mas rápido, dado por:

$$T(t) = \mu T^{t-1} \tag{5.4}$$

para alguna $0 < \mu < 1$ guiará al sistema a soluciones cercanas a la óptima en menor tiempo, sacrificando con ello precisión [23].
Aplicando el endurecimiento simulado a nuestro problema, es decir, encontrar el subconjunto de tripletes que preserven lo mejor posible las relaciones de vecindad definida en el espacio original para los organismos analizados, encontramos algunas buenas soluciones. Para ello, un estado es definido por la ausencia o presencia de cada variable. Un estado puede ser, por ejemplo la ausencia de los codones $11, 14$ y 30, lo que se representa como un vector de 64 posiciones con un 1 en cada posición, salvo en las posiciones $11, 14$ y 30, en donde encontraremos un 0. Movernos de este estado, que presenta cierta energía r a otro estado, por ejemplo, aquel donde no se encuentran las variables 11 y 30, que presenta una energía s, está dada por la distribución de Boltzmann-Gibbs.
La función de energía aplicada al tempaldo simulado fue la función objetivo usada en el algoritmo genético (ver ecuación 5.1).
Se muestran a continuación dos individuos encontrados por el método de templado simulado.

Características	Desempeño
26 43 51 49 37 58 22 36 1 35 45 19 33 24 16 7 8 6 21 57 0 48 55 54 27 13 56 52 44 9 32	372
22 36 49 37 58 14 33	384

Tabla 5.7: Desempeño del mapeo para dos espacios encontrados por templado simulado.

5.7 Relación entre el desempeño de un mapeo y la semejanza en la distribución de los objetos en el espacio original y el espacio para el cual se mapea

Recordemos del capítulo cuatro que dos mapeos para un mismo conjunto de datos no son idénticos aunque si semejantes. Entonces, ¿Cómo saber si el buen desempeño de un mapeo para un subconjunto de variables que preservan la topología original es algo consistente y no resultado del azar? Esto es, queremos demostrar que el encontrar variables que generan un espacio que preserva la distribución de los objetos presente en el espacio original es en realidad un método adecuado para obtener mejores mapeos.
Por ejemplo, ¿Qué ocurriría si un conjunto de variables define un espacio para los organismos que estamos analizando en donde preserva la topología adecuadamente, y un mapeo desde este nuevo espacio, efectivamente, mostrara un buen desempeño, en tanto que otro mapeo para el mismo conjunto de datos, mostrara una preservación de la topología mucho menor a la del primer mapeo? Mostraremos a continuación que este no es el caso, pues existe una correlación positiva entre la preservación de las vecindades de un espacio definido por un subconjunto de características con respecto al espacio original y el desempeño del mapeo para los objetos en ese nuevo espacio.
Si en un espacio definido por las variables $c \in C$, donde C es el conjunto de 64 tripletes, la distribución de los objetos es *muy semejante* a aquella en el espacio original definido por C, entonces el mapeo autoorganizado para los objetos en el espacio de dimensión c deberá ser casi tan bueno como el mapeo para los mismos objetos en el espacio de dimensión C. Para saber si la distribución de objetos en dos espacios de dimensión diferente es semejante, recordemos que se comparan los k vecinos en ambos espacios para cada objeto, arrojando tantas diferencias o errores, como objetos que no aparezcan en ambos casos.
En la tabla siguiente se muestra el desempeño de diversos mapeos para dos espacios: uno definido por 53 variables y el otro por 20. El número de diferencias en la distribución de los objetos en este espacio con respecto al espacio original es de 61 y de 172, respectivamente. El valor de k para identificar esas diferencias es de $k = 4$. Recordemos que el desempeño de un mapeo está dado por el número de errores en la distribución. Por ello, entre menor sea el valor indicado como desempeño, menos errores y más *confiabilidad.* Recordemos además que un mismo conjunto de objetos puede dar lugar a mapeos que no son idénticos, pero si semejantes.

Desempeño primer espacio	Desempeño segundo espacio
348	360
369	369
375	384
354	369
363	399
339	387
359	**376**

Tabla 5.8: Desempeño de diversos mapeos para los organismos de la tabla 2.7 en dos espacios definidos por diferentes conjuntos de tripletes. La dimensión del primer espacio es 53 y no se incluyen los tripletes 10, 11, 15, 24, 30, 32, 39, 41, 46, 50 y 59, mostrando 61 diferencias en la relación de vecindad. La dimensión del segundo espacio es 20 y muestra 178 diferencias en la distribución de los organismos con respecto al espacio original.

De acuerdo a la tabla 5.6, un espacio con menos diferencias en las relaciones de vecindad que otro, esto es, aquel que mejor preserve la topología con respecto al espacio de características, dará lugar a un mapeo autoorganizado con mejor desempeño. La tabla 5.9 muestra las diferencias en las relaciones de vecindad entre el espacio indicado y el original, así como el mapeo para el espacio especificado. Entre menor sean las diferencias entre ambos espacios, el desempeño del mapeo tenderá a ser mejor, esto es, presentar menos errores.

Para seleccionar al conjunto de características que mejor preserve las relaciones de vecindad entre los objetos analizados para el espacio generado por éstas con respecto al espacio orginal, hemos considerado una vecindad de $k = 4$. En la tabla 5.10 podemos ver el motivo de ello.

La tabla 5.10 es la matriz de correlación entre el error en la distribución de los objetos (relaciones de vecindad) entre el espacio original de dimensión 64 y un espacio de dimensión $m < 64$ y el desempeño del mapeo para los 54 organismos pero considerando únicamente los m tripletes. Se observa que cuando la cuantificación del error en la distribución de los organismos se lleva a cabo tomando los $v = 4$ vecinos, la correlación con el desempeño del mapeo para $k = 5$ es la más cercana a 1.0. Recordemos que queremos evaluar un mapeo para el caso $k = 5$ pues para este caso es que se tiene una mejor apreciación de los objetos estudiados.

Existe una correlación positiva[1] (0.9032) entre las diferencias en la distribución de los organismos en dos espacios (el original y el propuesto) y el desempeño del mapeo desde ese espacio propuesto, cuando se considera una $k = 4$. Esto indica que si ese número de diferencias aumenta (o disminu-

ye), el desempeño del mapeo *tenderá* a aumentar (o disminuir. Puesto que la correlación no es máxima, esto es, no es 1.0, no en todos los casos el mapeo resultante seguirá la indicación anterior. De hecho, resulta interesante el siguiente hecho: para el espacio original de 64 tripletes, los mapeos autoorganizados presentan desempeños menores a ciertos espacios definidos por subconjuntos de esos 64 tripletes. Sobre este hecho, se profundiza en la sección siguiente.

Se presenta en la tabla 5.8, además, el desempeño para el mapeo con $k = 27$ vecinos, porque, como se observa en las figuras 4.7 y 4.12, es en las cercanías este valor que la confiabilidad del mapeo es muy baja (el error en la confiabilidad es muy elevado). Se observa que la correlación aumenta a medida que aumenta el valor de v.

5.8 ¿Cuál es el mejor método de eliminación?

Se observó que el método de eliminación de características que no son importantes en la definición de topología parece superior al método de eliminación por baja desviación estándar (tablas 5.1 y 5.3). Cabe recordar que el desempeño se cuantificó para una $k = 5$. Lo anterior, como se comenta en el capítulo seis, es porque el mapeo autoorganizado preserva los vecinos *cercanos*, en tanto que para los vecinos lejanos, la violación en la topología se incrementa. Lo que queremos es un mapeo autoorganizado que preserve la topología no solo con respecto a sus vecinos cercanos, sino también, con respecto a sus vecinos lejanos.
Se muestra en la figura 5.11 el desempeño de diversos mapeos para los mismos objetos. Se muestra un mapeo para las 64 características originales; otro para únicamente 20 tripletes, otro mas para 35, 40 y 50.

Si bien es cierto que el método de eliminación de características por preservación de topología parece ser el mejor, es necesario mostrar aquí algunos

[1] El coeficiente de correlación es un número entre -1 y 1 que mide el grado en el que dos variables x, y se encuentran linealmente relacionadas. Si existe una relación lineal perfecta con pendiente positiva, entonces la correlación es 1; si existe el mismo caso, pero con pendiente negativa, la correlación será -1. Un coeficiente de correlación 0 indica que no existe una relación lineal entre las variables. Intuitivamente, si existe una correlación positiva, quiere decir que si el valor de una variable aumenta, el valor de la otra también se incrementará y viceversa. Se obtiene mediante la ecuación siguiente:
$\rho(x,y) = (\overline{xy} - \overline{x}\ \overline{y})/[\sigma^2(x)\sigma^2(y)]^{1/2}$ donde $\overline{x}$ es el promedio de x y $\sigma^2(x) = \sqrt{\sum_{i=1}^{N}(x_i - \overline{y})/n}$ es la desviación estándar. [67]

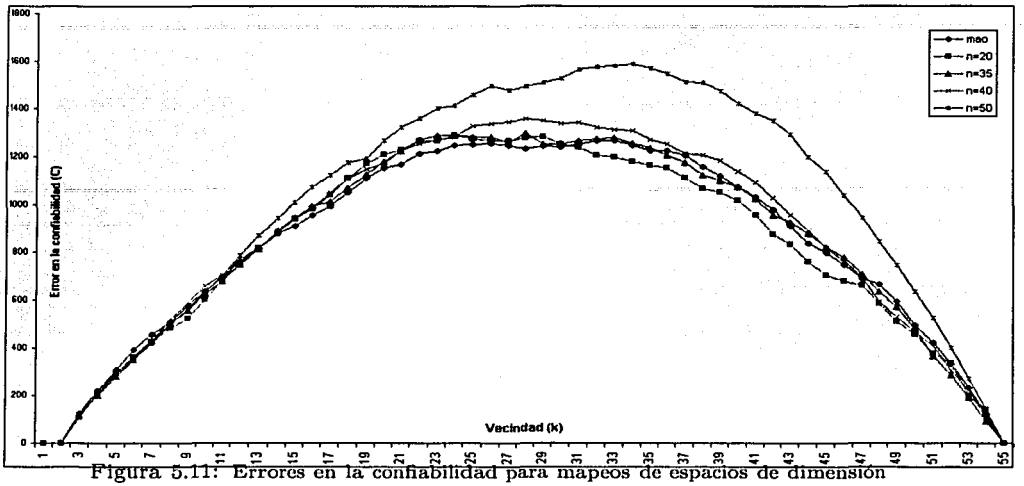


Figura 5.11: Errores en la confiabilidad para mapeos de espacios de dimensión $64(original), 20, 35, 40$ y 50. Para ciertos valores de k, el mapeo original(mao) se ve superado en el desempeño por mapeos para ciertos subconjuntos de los 64 tripletes que forman el espacio original. Los cuatro mapeos adicionales al original, fueron realizados para espacios identificados por un algoritmo genético.

detalles con respecto al mismo. Si un subespacio i del espacio orginal de características presenta una diferencia en la topología de α_i, y β_i es la confiabilidad del mapeo de i al espacio de la malla de neuronas y otro subespacio j del espacio original presenta una diferencia en la topología de α_j con una confiabilidad de su mapeo de β_j en donde $\alpha_i \leq \alpha_j$, entonces se observa una *tendencia* a cumplirse con la relación $\beta_i \leq \beta_j$, pero no en todos los casos.

5.9 Discusión

Se han mostrado diversas metodologías para descartar variables que pudieran no ser útiles para el mapeo autoorganizado. Lo que se pretende es disminuir la dimensión del vector de características tanto por fines de costo computacional como por fines explicativos.
Se compararon resultados de tres técnicas y se mostró que aquella denominada Eliminación por Preservación de Topología muestra en general mejores resultados, no solamente en cuanto al desempeño del mapeo, sino también en cuanto a la identificación de las variables relevantes para la definición de la topología.

En el problema presentado en este trabajo, la dimensión del espacio de búsqueda es de 64. Encontrar un subconjunto de estas características que conserven las relaciones de vecindad entre los objetos lo mejor posible es un problema NP-completo. Se propuso un algoritmo genético para encontrar dicho subconjunto y se mostraron diversas soluciones encontradas por éste. Así mismo, se recurrió al templado simulado como una alternativa al algoritmo genético, pero, en general, los resultados del primero fueron mejor a los del segundo.

Con el método de templado simulado se encontraron subconjuntos del total de características de tal forma que el mapeo autoorganizado desde el espacio definido por dicho subconjunto muestra desempeños adecuados; sin embargo, el resultado obtenido por este método fue inferior al obtenido por medio de algoritmos genéticos.

Se encontró también que para algunos subconjuntos de los 64 tripletes, el mapeo presenta un desempeño casi tan bueno como el desempeño del mapeo obtenido precisamente desde el espacio original de dimensión 64.

Errores	Des. del mapeo	Errores	Des. del mapeo
109	372	82	360
90	384	1212	504
364	405	175	369
142	351	511	408
432	408	690	414
94	330	143	401
555	411	389	372
187	371	1230	495
865	432	76	360
2056	507	119	342
658	408	883	456
883	456	134	366
140	348	327	381
1265	450	550	390
139	342	73	390
191	390	1340	489
223	375	355	393
631	423	1078	477
509	432	1996	519
234	390	611	432
534	420	768	423
184	372	82	384
99	385	509	393
95	405	348	372

Tabla 5.9: Relación sobre la preservación de relaciones de vecindad entre el espacio original de dimensión 64 y un espacio de dimensión menor y el desempeño del mapeo. La primer columna se refiere al número de errores en la relación de vecindad entre los dos espacios, en tanto que la segunda columna muestra el desempeño del mapeo desde el espacio de dimensión menor a 64 considerado como posible alternativa.

	k=2	k=3	k=4	k=5	k=6	k=27
v=2	0.7223	0.8167	0.8717	0.8941	0.8844	0.7035
v=3	0.712	0.818	0.874	0.9025	0.8941	0.7163
v=4	0.7174	0.8189	0.8754	0.9032	0.8961	0.7251
v=5	0.7169	0.8152	0.8721	0.8897	0.8932	0.7316
v=6	0.7083	0.8128	0.8702	0.8997	0.8947	0.7321

Tabla 5.10: Correlación entre el error en la distribución de los objetos entre el espacio original y un espacio de dimensión $m < 64$ y el desempeño del mapeo para el espacio m-dimensional. La preservación de la topología en el espacio de dimensión m se calcula para los v primeros vecinos; el desempeño del mapeo se calcula para los k primeros vecinos. Se llevaron a cabo 850 mapeos diferentes para el cálculo de la correlación.

Capítulo 6

Otros mapeos y técnicas de visualización

6.1 Introducción

Hemos visto en los capítulos anteriores que el mapeo autoorganizado es una herramienta adecuada para visualizar, en un espacio de baja dimensión (generalmente dos), objetos multidimensionales. Ahora, es preciso preguntarnos si efectivamente ese mapeo, como se manifiesta en la bibliografía [52], es una de las mejores herramientas de visualización para objetos multidimensionales.
En este capítulo mostraremos algunas técnicas de visualización de objetos multidimensionales. Se analiza el mapeo obtenido por programación genética, que minimiza el error en la confiabilidad. Se analizan también el escalamiento multidimensional tradicional y el análisis de componentes principales, dos de las técnicas más socorridas en la representación de objetos multidimensionales.
En una de las secciones, se habla de las ventajas de cada mapeo, dependiendo del número de objetos a considerar como los vecinos. Recordemos que este número esta dado por el parámetro k (sección 4.2).

6.2 El mapeo autoorganizado como herramienta de visualización

Cuando queremos conocer la distribución de objetos multidimensionales, es necesario visualizarlos en un espacio de baja dimensión, generalmente dos

[26]. Una buena herramienta de visualización presentará los objetos en el espacio de baja dimensión de tal forma que refleje lo mejor posible la distribución de los objetos en el espacio de alta dimensión.
El mapeo autoorganizado es una herramienta adecuada para la visualización pues refleja en buena medida esa distribución original de los objetos. El mapeo autoorganizado ha sido utilizado como herramienta de visualización para diversos dominios [55] [92], además de ser constantemente utilizado como herramienta de visualización de cúmulos [29].
El número de neuronas en la malla es determinante para identificar el uso que se le da a un mapeo autoorganizado: si el número es menor que el total de objetos analizados, muy probablemente se trata de una aplicación de formación de cúmulos, puesto que la idea detrás de esta herramienta es *agrupar* a objetos semejantes entre sí de tal forma que objetos que pertenezcan a grupos (cúmulos) diferentes es porque son heterogéneos [30]. El total de cúmulos es siempre menor al total de objetos, aunque el número óptimo de cúmulos para una aplicación en especial es un problema abierto [26].
Por otro lado, si el número de neuronas es mayor que el número de objetos analizados, la aplicación es de visualización [70]. El número de neuronas necesarias para una buena visualización es un problema abierto, pero [29] reporta que mas allá de un límite, el desempeño de un mapeo autoorganizado no presenta mejorías. Dicho límite esta situado entre las 400 y las 900 neuronas, dependiendo de la cantidad de datos analizados.

6.3 Mapeos que preservan la topología identificados por programación genética

La programación genética es similar al algoritmo genético, solo que en ésta, los individuos representan ecuaciones matemáticas que intentan aproximar una función desconocida [64]. La figura 6.1a muestra la codificación de individuos, el cruzamiento y la mutación.

Las funciones son representadas en forma de árbol, lo que facilita su manipulación. El algoritmo de búsqueda es el mismo que para el algoritmo genético. En la programación genética, la cruza se refiere al intercambio de subárboles entre los dos *padres* para dar lugar a dos hijos, posiblemente diferentes a los padres [89]. La figura 6.1 muestra el intercambio de subárboles entre padres para formar nuevos hijos.
La mutación muestra una mayor variedad de opciones que su homólogo en los algoritmos genéticos. En la programación genética se *modifica*, con una probabilidad baja (0.01- 0.05) [64], un subárbol de cada individuo. Si el

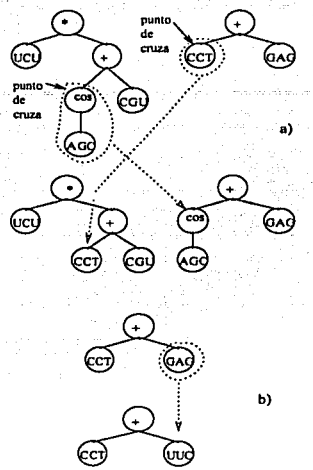


Figura 6.1: Representación en forma de árboles de la función $UCU*(cos(AGC)+CGU)$ y $CCT+GAC$. a) Para cruzar dos individuos, se elige un punto de cruza en cada uno de ellos y se intercambia el subárbol restante. b) Para mutar un individuo, se elige el punto de muta y se reliza una de varias posibilidades: cambiar la variable (GAG por UUC, por ejemplo), sustituir por un nuevo subárbol, sustituir por una constante, etcétera.

subárbol no es una hoja, se puede sustituir el operador por otro de la misma aridad (mismo número de argumentos); puede también ser sustituido el subárbol asociado al nodo de cruza seleccionado por otro subárbol creado aleatoriamente; si se trata de una hoja, esto es, una constante o una variable, se puede sustituir por otra variable o constante, o se puede sustituir por un nuevo árbol creado aleatoriamente.
La programación genética ha sido aplicada en los mas diversos problemas. Parmee [77] lo ha aplicado en la optimización de estructuras tridimensionales para minimizar la resistencia al viento; [81] ha recurrido a esta técnica para la predicción de condiciones del clima, en tanto [99] lo ha usado incluso para la creación de obras musicales.
Regresemos al problema de representar objetos multidimensionales en un espacio de dimensión dos. Hemos visto que el mapeo autoorganizado representa dichos objetos en el espacio del mapeo, bidimensional, como neuronas

activas (configuración ordenada que conservan la topología presente en el espacio original). Hemos visto también, que mediante la métrica de confiabilidad podemos calcular el desempeño de un mapeo en particular. Si fuésemos capaces de encontrar un función que mapeara del espacio de características al espacio de dimensión dos, de tal forma que la topología se preservara, tendríamos entonces un mapeo construído mediante programación genética.

Sea i un punto en el espacio de dimensión N, representado i por el vector de características $\mu_i = [C_{i0}, C_{i1}, ..., ..., C_{iN}]$ y sea j otro punto en el mismo espacio que i, pero representado por el vector de características $\mu_j = [C_{j0}, C_{j1}, ..., C_{jN}]$. Si podemos encontrar dos ecuaciones f y g de tal forma que la representación en dos dimensiones de i, $\eta_i = [x_i, y_i]$ y la representación en dos dimensiones de j, $\eta_j = [x_j, y_j]$ preserven la topología (la relación de vecindad) mostrada en el espacio de dimensión N, entonces f y g forman un mapeo que preserva la topología.

Lo anterior, más formalmente, puede expresarse como:

Sea I un conjunto de i puntos en el espacio de dimensión N:

$\mu_0 = [C_{00}, C_{01}, ..., C_{0N}]$ es el vector de características del objeto 0.
$\mu_1 = [C_{10}, C_{11}, ..., C_{1N}]$ es el vector de características del objeto 1
...
$\mu_i = [C_{i0}, C_{i1}, ..., C_{iN}]$ es el vector de características del objeto i

Y sea Φ_{IN} una topología definida sobre I (posiblemente una relación de vecindad). Si encontramos dos funciones f y g, tales que para cada punto $j \in I$:

$$\eta_j = [x_j, y_i] = (f(C_{j0}, C_{j1}, ..., C_{jN}), g(C_{j0}, C_{j1}, ..., C_{jN}))$$

y la topología inducida sobre todos los puntos $\eta_j \in I$, Φ_{I2} sea lo mas semejante a Φ_{IN}, entonces f y g forman un mapeo que preservan la topología. Las funciones f y g pueden ser encontradas por medio de programación genética. Como se explicó con anterioridad, la programación genética busca en el espacio de funciones y variables, aquellas que cumplan, lo mejor posible, con un conjunto de valores dados. De esta forma, lo que las funciones f y g deben cumplir es que los puntos (x, y) que ellas forman, cumplan con la topología que muestran los puntos en el espacio de dimensión N, aquellos que son en realidad los argumentos de f y g.
La programación genética busca minimizar el número de vecinos en el espacio de características de cada objeto que dejan de serlo en el espacio del mapeo

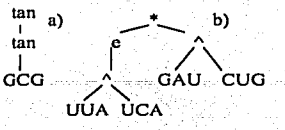


Figura 6.2: Funciones encontradas por programación genética que maximizan la topología presente en el espacio de dimensión 64. a) La función que mapea al eje x; b) La función que mapea al eje y.

(bidimensional). De esta forma, la función objetivo es la presentada en la ecuación 4.1.

En la figura 6.2 se observan las funciones f y g obtenidas por programación genética. La función que mapea al eje x es:

$$f = tan(tan(GCG))$$

en tanto que la función que mapea al eje y es:

$$g = (e^{UUA^{UCA}}) * (GAU^{CUG})$$

La distribución obtenida por dichas funciones se observa en la figura 6.3. La topología definida para el caso de la figura 6.3 es en términos de los $k = 5$ vecinos mas próximos.

6.4 Visualización por análisis de componentes principales

El análisis de componentes principales, *acp*, es una técnica estadística ubicada en el área de *análisis de factores*. El propósito del *acp* es el identificar la dependencia presente en un conjunto de datos multivariados con el objetivo de obtener una descripción reducida de los datos. Cuando existe una correlación diferente a cero entre las n variables que describen el fenómeno, el número de variables para describir ese fenómeno es de m. n es conocida como la *dimensionalidad de la superficie* en tanto que m es conocida como la *dimensionalidad intrínseca* de los datos. Entre mayor sea la correlación entre las variables, m tenderá a ser menor [1].
Las n variables observadas son representadas como funciones de las m variables, llamadas *factores*, donde $m < n$ y frecuentemente se observa que $m \ll n$. Los factores son comúnmente llamados *rasgos* y el vector que forman es el *espacio de rasgos*.

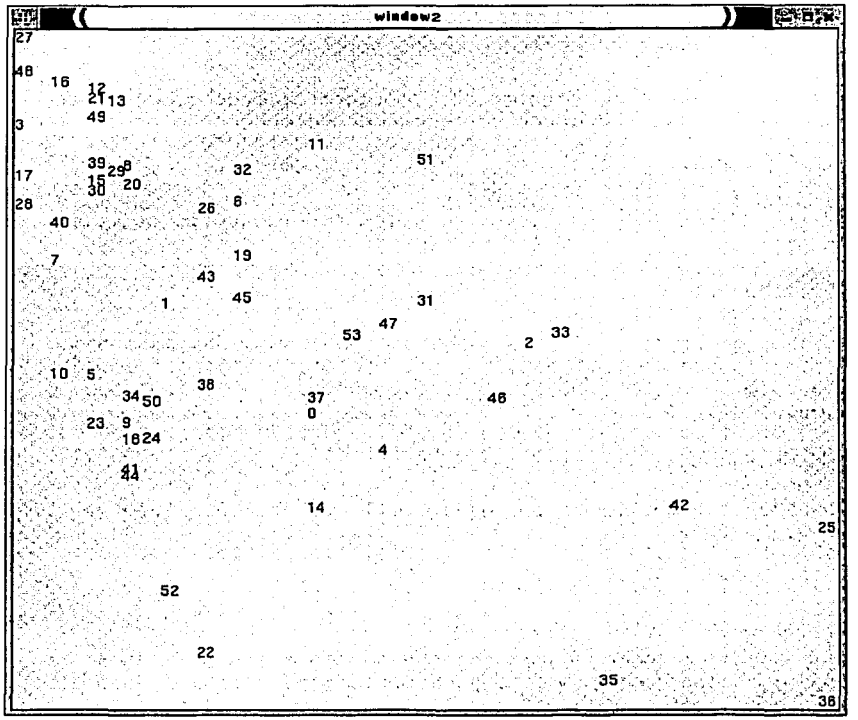


Figura 6.3: Mapeo que preserva la topología obtenido por programación genética para los 54 organismos de la tabla 2.7. El error en la confiabilidad es $C = 297$ para $k = 5$ (Ecuación 4.1).

Consideremos al vector $x = [x_1, ..., x_n]^T$ con media $Ex = 0$ y matriz de covarianza $R_x = Exx^T$. El vector de rasgos y es una transformación lineal de los datos:

$$y = Wx$$

donde las columnas de W forman una base ortonormal del subespacio L. La proyección de x en L es la reconstrucción de x desde y:

$$\hat{x} = W^T y = W^T W x$$

El objetivo de ACP es el de minimizar el error medio cuadrático de reconstrucción:

$$J_e = E||x - \hat{x}||^2$$

$Sean \lambda_1, \lambda_2, ..., \lambda_n$ los valores característicos de R_x listados en orden decreciente y sean sus vectores propios $e_1, e_2, ..., e_n$. El mínimo para el error medio cuadrático de reconstrucción, J_e, bajo la restricción de $WW^T = I$, tiene la forma:

$$W_opt = S[\pm e_1 ... \pm e_m]^T$$

Donde S es cualquier matriz ortogonal cuadrada. El error mínimo en la reconstrucción es:

$$minJ_e = \sum_{i=m+1}^{n} \lambda_i \text{ [20]}$$

Gráficamente, el ACP es descrito como una rotación de los objetos multidimensionales de tal forma que el eje con la mayor varianza sea el primer eje en el ACP, el segundo eje sea el segundo con la mayor varianza etcétera [1]. Cada componente muestra un porcentaje de la varianza total. Para algunas aplicaciones [73], los dos primeros componentes llegan a presentar hasta un 90% de la varianza total. Para los organismos analizados en este trabajo, los trece primeros ejes muestran un porcentaje de la varianza de 78.5%. Se muestra en la tabla 6.1 la varianza por cada componente hasta contar con una varianza acumulada de 90%. La figura 6.4 muestra la distribución de los organismos estudiados tomando los dos componentes principales.

Eje	Porcentaje	Porcentaje acumulado
1	36.7	36.7
2	8.1	44.9
3	7.5	52.4
4	3.2	56.6
5	3.2	58.8
6	3.1	61.9
7	3.1	65.1
8	2.7	67.7
9	2.5	70.3
10	2.4	72.7
11	2.3	75.05
12	1.7	76.7
13	1.6	78.3
14	1.5	79.8
15	1.4	81.2
16	1.4	82.6
17	1.2	83.8
18	1.1	84.9
19	1.1	86.0
20	1.0	87.0
21	0.85	87.85
22	0.8	88.65
23	0.7	89.35
24	0.6	89.95
25	0.5	90.45

Tabla 6.1: Varianza para los primeros 25 ejes obtenidos por análisis de componentes principales. Se muestra también la varianza acumulada.

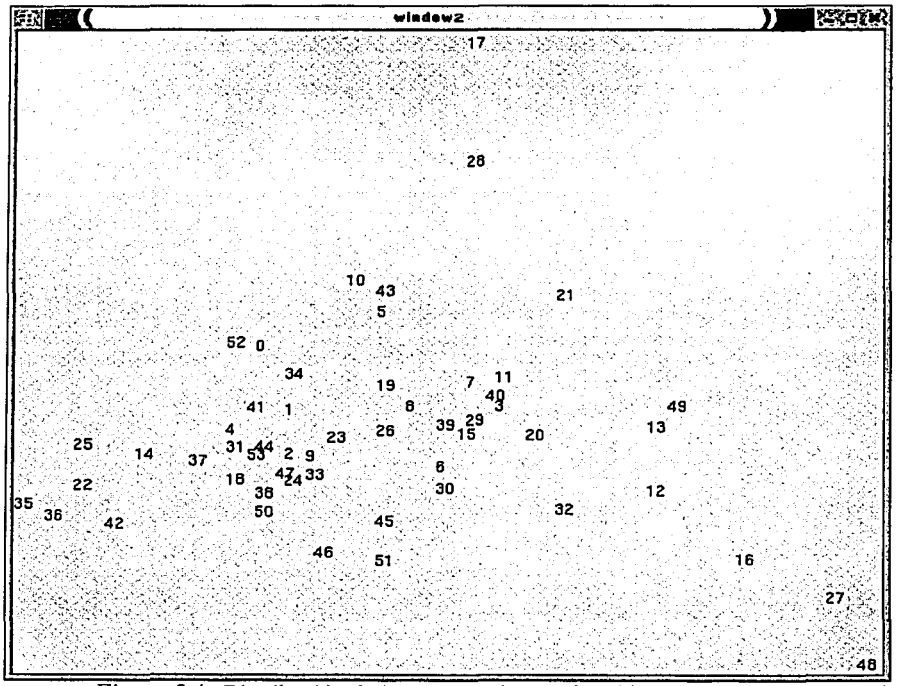


Figura 6.4: Distribución de los 54 organismos obtenida por componentes principales para los dos ejes principales. El error en la confiabilidad es $C = 393$ para $k = 5$.

6.5 Escalamiento multidimensional

El escalamiento multidimensional es una técnica de análisis multivariado en la que se representa en un espacio bidimensional a objetos multidimensionales, tomando en cuenta para ello, las relaciones de vecindad mostradas entre pares de objetos en el espacio multidimensional [19].
El escalamiento multidimensional es uno de los mapeos tradicionales más empleados para la visualización. Una de las variantes mejor estudiadas es el mapeo de Sammon, el cual intenta minimizar la expresión:

$$f \sum_{i=0}^{n-1} \sum_{j<i}^{n} (d(x_i, x_j) - \dot{d}(\dot{x}_i, \dot{x}_j))^2 / d(x_i, x_j) \tag{6.1}$$

donde

$$f = 1/(\textstyle\sum_{i=0}^{n-1} \sum_{j<i} d(x_i, x_j)) \text{ y}$$

$\dot{d}(\dot{x}_i, \dot{x}_j)$ es la distancia en el espacio del mapeo que corresponde a la distancia $d(x_i, x_j)$ en el espacio de características y n es el número de objetos analizados [30]. La distancia considerada en este trabajo fue la euclidiana.
El escalamiento multidimensional fue originalmente propuesto en estudios psicométricos, con el objetivo de entender los juicios que un cierto individuo realiza sobre diversos objetos [83]. Desde entonces, se ha convertido en una herramienta de visualización de objetos multidimensionales bastante aplicada en las disciplinas sociales, médicas y biológicas [20].
En la visualización en el ámbito de la biología molecular, [15] ha utilizado esta técnica para analizar patrones en el uso de codones en genes de un mismo organismo (*Schistosoma mansoni*). Con la distribución de los genes analizados, lograron encontrar heterogeneidad en la frecuencia de uso de los tripletes, lo que los llevó a concluir que el sesgo se debe a la composicón general del gene estudiado.
Patrones en el sesgo en uso de codones entre diferentes organismos también han sido encontrados haciendo uso del escalamiento multidimensional [16]. Se reporta en ese trabajo que cuando se compara el uso de codones entre dos especies, la herramienta produce una distribución más confiable que cuando se analizan mas especies.
En la figura 6.5 se muestra la distribución de los 54 organismos analizados obtenida por medio de escalamiento multidimensional. El escalamiento multidimensional intenta preservar la distancia entre cada par de objetos y, como se observa en la figura, las relaciones de vecindad se preservan peor que en los mapeos autoorganizado y por programación genética para valores de k (los k

primeros vecinos) pequeños. Puede observarse asi mismo, que las relaciones de vecindad se preservan notablemente mejor para valores de k grandes. Como se mencionó en la sección 4.8, preservar las relaciones de vecindad para valores de k grande, es mucho más fácil que preservarla para valores de k pequeños.

6.6 Comparación entre mapeos

Los mapeos autoorganizados preservan la topología que existe en el espacio de características de alta dimensión. La preservación de la topología, como se ha manejado en este trabajo, se refiere a que los objetos cercanos entre si en el espacio multidimensional de características serán mapeados a neuronas cercanas en el espacio del mapeo [13].
Recordemos que la métrica de preservación de topología a la que recurrimos es la de confiabilidad, analizada en la sección 4.6. En ella, el parámetro k, que indica el número de objetos que habremos de considerar como vecinos, es determinante para cuantificar el número de errores de un mapeo. La figura 6.6 muestra el error para diversas herramientas de visualización para diferentes valores de k.

Se observa que para valores pequeños de k, el mapeo autoorganizado es mejor que el escalamiento tradicional, pero a medida que k aumenta, el escalamiento multidimensional mejora notablemente y llega a ser muy superior al mapeo autoorganizado. Se muestra también en la gráfica el error para un mapeo encontrado por medio de programación genética que minimiza el error en la confiabilidad para $k = 5$. Dicho mapeo muestra consistentemente un menor error que el mapeo autoorganizado, incluso para valores de k pequeños.
Cuando el valor de k es pequeño, digamos, $5-10$ %, el mapeo autoorganizado es mejor que el escalamiento multidimensional y que la distribución por medio de los dos componentes principales [91]. Recordemos que el desempeño de un mapeo se cuantifica para los cinco primeros vecinos ($k = 5$), por lo que el mapeo autoorganizado es una mejor alternativa que los ya mencionados métodos estadísticos. El mapeo obtenido por programación genética es el que menor error presenta para ese rango de vecinos.

6.7 Discusión

Se han presentado mapeos alternativos al mapeo autoorganizado, comúnmente utilizados en la visualización de objetos multidimensionales, con la idea de

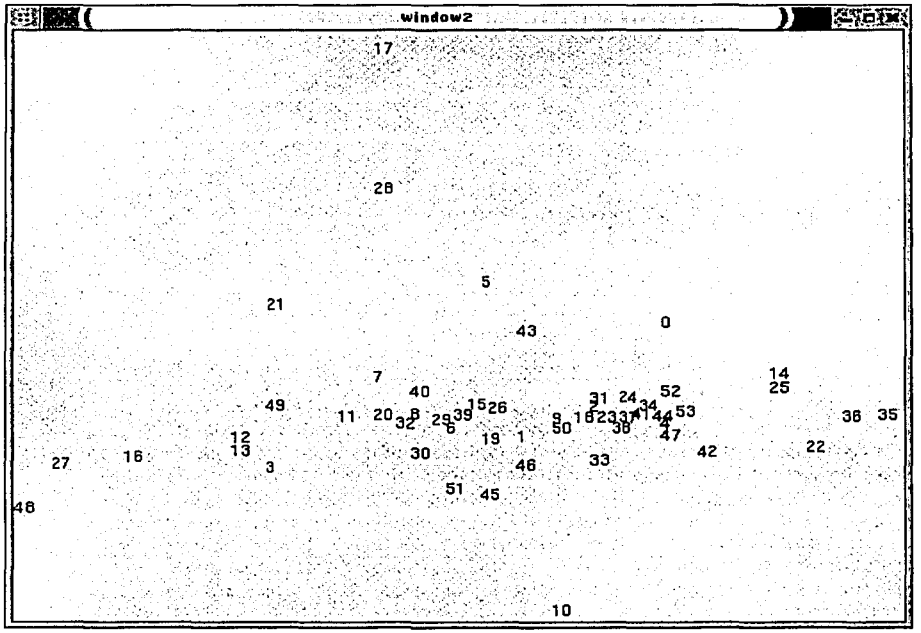


Figura 6.5: Distribución de los 54 organismos obtenida por escalamiento multidimensional. El error en la confiabilidad es C=363 para $k = 5$.

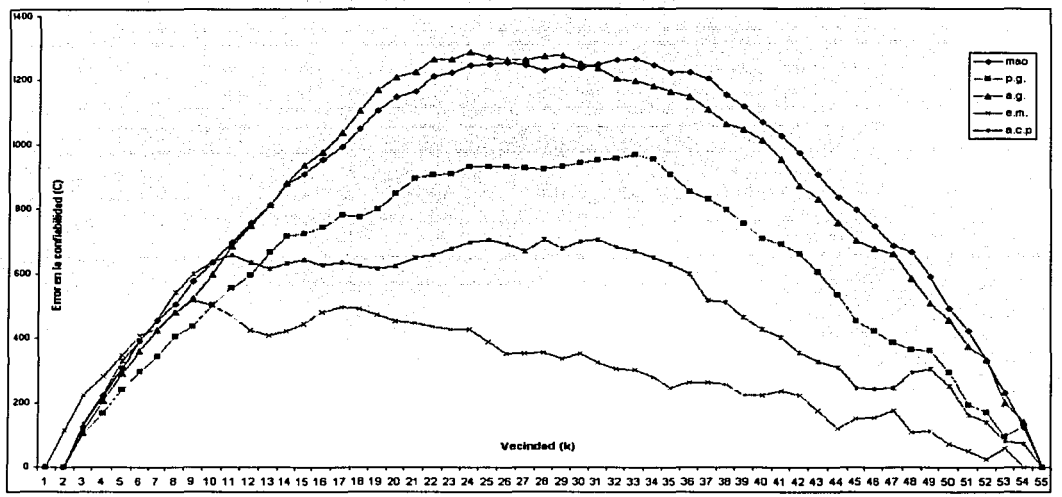


Figura 6.6: Error en la confiabilidad de diversas herramientas de visualización. El desempeño para mapeo autoorganizado para las 64 variables se indica con rombos (mao): su desempeño es menor cuando la k es aproximadamente $N/2$; Un mapeo autoorganizado para un subconjunto de tripletes (ver primer caso de tabla 5.6) es señalado por triángulos (a.g.): su desempeño es similar al del mapeo autoorganizado original; el desempeño mostrado por un mapeo obtenido por programación genética es mostrado por medio de cuadrados (p.g.); el desempeño por escalamiento multidimensional se indica por medio de cruces (color claro, e.m.); el desempeño exhibido por los dos componentes principales obtenidos por ACP se muestra con asteriscos en color obscuro (a.c.p.).

identificar aquel que muestre el menor error, es decir, la máxima confiabilidad. Se presentó el mapeo que minimiza el error en la confiabilidad obtenido por medio de programación genética, el cual tiene errores por debajo del mapeo autoorganizado tanto para primeros vecinos como para vecinos más alejados.

El escalamiento multidimensional también fue presentado, y se mostró que presenta errores mayores al mapeo autoorganizado para valores de k pequeños. El análisis de componentes principales también fue presentado y se observa que el error que muestra es mayor al error del mapeo autoorganizado para los primeros vecinos.

El mapeo autoorganizado es, pese a que presenta errores por encima del mapeo obtenido por programación genética, una buena herramienta de visualización, pues su inspiración biológica permite una mejor explicación del fenómeno analizado que la que se podría obtener por el segundo.

Capítulo 7

Resultados, conclusiones y trabajo futuro

7.1 Resultados

La visualización de las relaciones de vecindad (topología) de objetos multidimensionales puede ser realizada por medio de un mapeo autoorganizado. Se mostró en este trabajo que el mapeo autoorganizado es una buena alternativa para ello, principalmente si las vecindades son pequeñas. Se analizaron también diversas métricas para evaluar la bondad de un mapeo. Se intentó explicar con particular detalle que de todas las variables que describen a un cierto objeto, no todas intervienen por igual en la definición de topología, lo que se traduce en que no todas las variables intervienen en el mapeo autoorganizado.

Se habló también de otros mapeos, y se mostró que uno de ellos, el obtenido por programación genética, muestra una mejor confiabilidad que el mapeo autoorganizado para vecindades pequeñas.

7.1.1 El mapeo autoorganizado como herramienta de visualización

Se mostró a lo largo de este trabajo que el mapeo autoorganizado es una buena herramienta, principalmente cuando se quiere preservar las relaciones de vecindad con los primeros vecinos (k pequeño). Se mostró también que otras herramientas lo superan, incluso para el caso de k pequeño. No obstante, las herramientas que lo superan son supervisadas, es decir, tienen que ser alimentadas con una función de error que les permita saber si lo que

están haciendo se acerca o aleja de la solución ideal, en tanto que el mapeo autoorganizado no es supervisado.
El mapeo autoorganizado preserva la topología como una consecuencia de la función de modificación de los pesos, no como consecuencia de alguna función de error. Se tiene evidencia de que el proceso que lleva a cabo el mapeo autoorganizado se lleva a cabo en diferentes regiones del cerebro [80].
El mapeo autoorganizado presenta varias desventajas: carece de una función de energía, no existe un argumento sólido para la elección de los parámetros para asegurar el ordenamiento topológico y no existen pruebas de convergencia salvo para el caso unidimensional[21]. No obstante, su funcionamiento inspirado en el funcionamiento del cerebro, le ha dado un impulso en un variado y amplio número de aplicaciones de visualización, mostrando en muchos campos resultados adecuados.

7.1.2 Desempeño del mapeo con métricas más rígidas

Del capítulo cuatro, tenemos que un buen mapeo es aquel que presenta un menor error en la confiabilidad. La ecuación 4.1 define el valor C, el error en la confiabilidad de un mapeo. Recordemos que ésta cuantifica, para cada objeto, el número de objetos que se encuentran en su vecindad en el espacio de características pero no así en el espacio del mapeo. Sin embargo, esta cuantificación no es rígida, pues si un objeto no preserva el orden de vecindad con respecto a otro, pero si se encuentra dentro de los k vecinos en el mapeo, no es castigado.
Ahora, en caso de que se modifique la métrica, es decir, un objeto debe preservar el orden de vecindad, y no solamente estar dentro de los k primeros vecinos, estaremos cuantificando con mayor rigidez el desmepeño del mapeo. La figura 7.1 muestra el error en la confiabilidad para los 6 primeros vecinos.

Se observa que el mapeo autoorganizado para los 64 tripletes se encuentra por abajo del escalamiento multidimensional, los componentes principales y el mapeo autoorganizado para un subconjunto de 10 tripletes; con respecto al mapeo por programación genética, el desempeño es semejante al menos para $k < 7$. El mapeo autoorganizado para el subconjunto de 10 tripletes muestra un desempeño consistentemente peor que las otras herramientas. Esto se debe a que el algoritmo genético que identificó a esos 10 tripletes lo hizo con base en la minimización de la ecuación 4.1. Esa ecuación no contempla el que los objetos se encuentren en el mismo orden en el espacio del mapeo que en el espacio original. Lo anterior también es cierto para la programación genética.

bacillus subtilis	staphylococcus aureus
mycoplasma genitalum	virus del papiloma humano
sulfolobus solfataricus	virus A del herpes humano
mycobacterium leprae	salmonella typhi
cyanophagep60	molluscum contagiosum virus
Virus de la fiebre hemorrágica en simios	thermoplasma volcanium
pyrococcu sabyssi	pyrococcus furiosus
sulfolobus tokodaii	pyrobaculum aerophilum
bacteriófago phic 31	mycobacteriófago tm 4
Virus de la influenza (Puerto Rico)	

Tabla 7.1: Organismos pertenecientes al conjunto de prueba.

7.1.3 Capacidad de generalización

Cuando un sistema ha *aprendido* con un cierto conjunto de datos, es necesario comprobar que tan bien lo ha hecho [62]. Para ello, es necesario evaluarlo con un conjunto de datos diferente a aquel con el que fue entrenado. En el presente trabajo, el mapeo autoorganizado, así como también el mapeo obtenido por programación genética, han aprendido las relaciones de vecindad que muestran los organismos analizados; sin embargo, es necesario verificar su capacidad de *generalización* sobre un conjunto de objetos de prueba (o de control).

La tabla 7.1 muestra el nombre de los 19 organismos de prueba. Durante el proceso de autoorganización, los pesos, como se mencionó en el capítulo tres, se modificarán a fin de aproximar la distribución de los objetos en el espacio multidimensional de características. Una vez que el proceso ha terminado, el conjunto de prueba es presentado a la red. Aquellas neuronas que mejor respondan a cada uno de los nuevos objetos formarán el mapeo. Este mapeo es evaluado con respecto a la preservación de las relaciones de vecindad por medio de la ecuación 4.1.
Al igual que con el mapeo autoorganizado, el mapeo identificado por programación genética fue evaluado con el mismo conjunto de prueba y se observa en la figura 7.2 que las relaciones de vecindad se preservan mejor en este mapeo que en aquel.

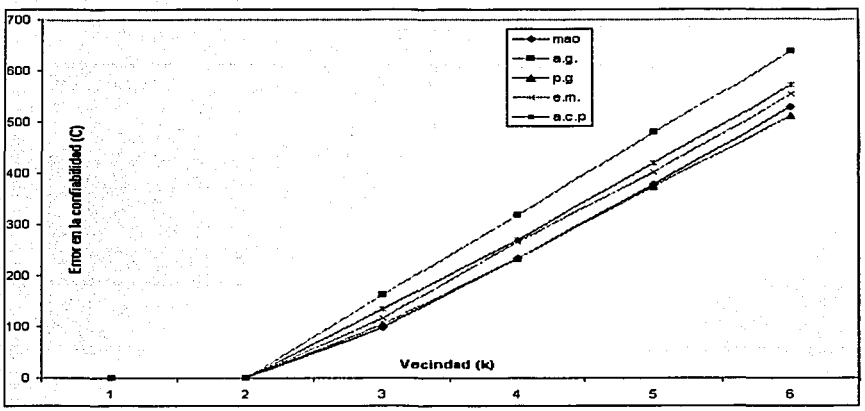


Figura 7.1: Error en la confiabilidad de diversas herramientas de visualización con la métrica mostrada en la ecuación 4.1 modificada para comparar el orden correcto de la vecindad de los objetos y no únicamente el que éstos se encuentren dentro de los k primeros vecinos en el mapeo (mao, mapeo autoorganizado; a.g., mapeo autoorganizado para 10 tripletes; p.g. mapeo obtenido por programación genética; e.m., escalamiento multidimensional; a.c.p, proyección obtenida por medio de los dos primeros componentes principales). Se muestra el desempeño para valores de $k < 7$.

7.1.4 Características eliminadas para el mapeo autoorganizado

Como se mostró en el capítulo cinco, algunos de los rasgos que describen un objeto pueden ser descartados sin que se pierda confiabilidad. Aquellas variables que no intervienen de manera significativa en la definición de topología son precisamente las que pueden descartarse.
Cuando las dimensiones del problema son *numerosas*, es deseable eliminar aquellas que no intervengan en la definición de las relaciones de vecindad, a fin de que el tiempo requerido por el proceso de autoorganización disminuya [50].
La figura 5.11 muestra que el desempeño no se ve afectado significativamente al eliminar algunas variables. Aquellas variables que son eliminadas son las que su ausencia no modifica las relaciones de vecindad (o la modifican lo menos posible) para cada objeto en el espacio original de características (de dimensión 64).
La tabla 5.10 muestra que existe una correlación positiva entre la preservación de la topología para los cuatro primeros vecinos en el espacio original y la preservación de la topología en el espacio del mapeo autoorganizado para los cinco primeros vecinos, lo que indica que eliminar rasgos que no modifican la topología formará espacios cuyo mapeos autoorganizados presentarán un desempeño semejante a aquel que presenta el espacio original.
Mediante un algoritmo genético, encontramos dichos subconjuntos. Sin embargo, continúa siendo un problema abierto el encontrar el conjunto mínimo de variables que maximice la confiabilidad. Encontrar aquellas características que intervienen de manera negativa en el proceso del mapeo autoorganizado es también una posible extensión al presente trabajo.

7.1.5 Distribución de organismos y superreino

La figura 7.3 muestra la distribución obtenida por medio de un mapeo autoorganizado de los 54 organismos analizados y cada tono de gris indica un reino taxonómico. Se observa que en general no se puede hablar de un solo cúmulo para cada especie, sino que los organismos pertenecientes a cada especie se distribuyen a lo largo de la malla.

Si el uso de codones estuviera relacionado con el reino taxonómico, los organismos semejantes entre si quedarían en la misma región, en tanto que organismos pertenecientes a diferentes reinos, serían mapeados por neuronas alejadas entre si. Por ejemplo, la *Apis mellifera*, o abeja, queda en la cercanía de *Vibrio cholerae* y de la *Escherichia coli*, organismos con los que no

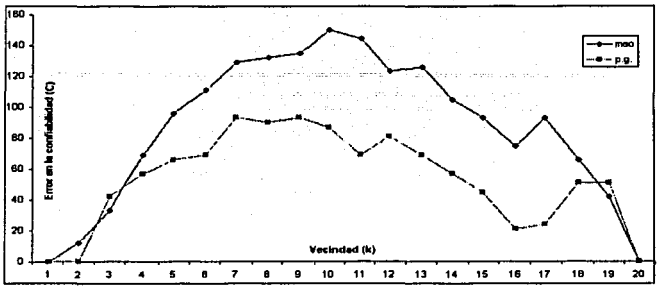


Figura 7.2: Error en la confiabilidad para el mapeo autoorganizado (rombos) y el mapeo obtenido por PG (cuadrados) para un conjunto de organismos de control como función del parámetro de vecindad (k).

comparte ni reino ni proximidad filogenética.
El mapeo obtenido por programación genética muestra una situación similar, como se muestra en la figura 7.4. El mapeo obtenido por programación genética presenta un mejor desempeño que el mapeo autoorganizado mostrado en la figura 7.3 y aquí resulta más claro que la distribución no parece estar relacionada con el reino.
Si existiese una relación entre el uso de codones y el superreino, habría al menos cuatro cúmulos en la distribución de los organismos, uno para cada superreino; sin embargo, en las dos figuras mencionadas, podemos notar que esto no es asi.

7.1.6 Distribución de organismos y contenido G+C

De la sección 2.5 recordamos que una de las teorías sobre el uso de codones es el contenido del nucleótido G o C. La figura 7.5 muestra la distribución generada por un mapeo autoorganizado e indica el contenido de $G + C$ para cada organismo. Se observa que aquellos organismos con un contenido similar de $G + C$ tienden a estar juntos. La figura 7.6 muestra la distribución de los objetos analizados obtenida mediante programación genética. En ésta, también se observa que los organismos que presentan un contenido de $G + C$ tienden a estar en una misma región.

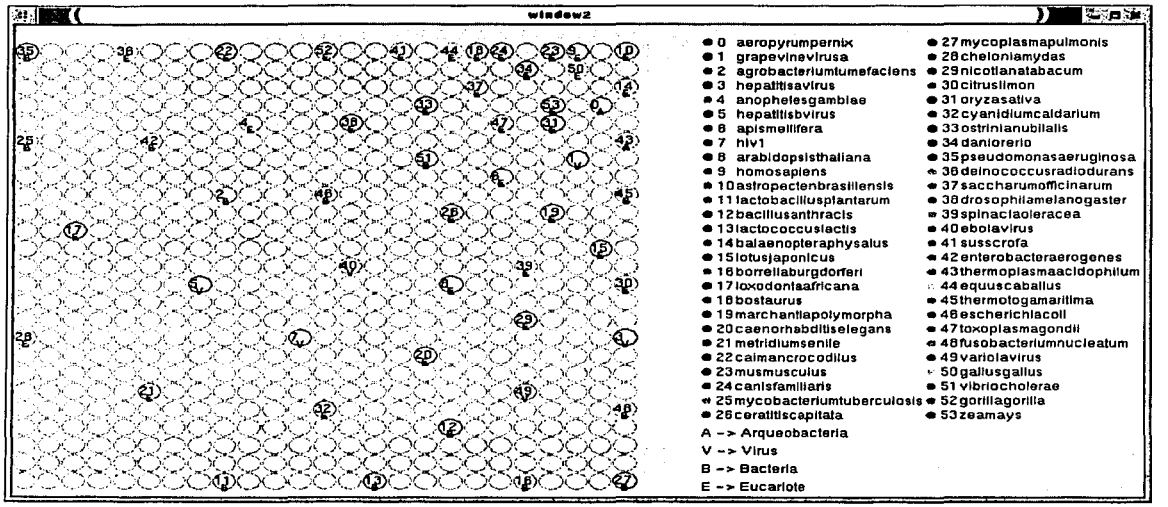


Figura 7.3: Distribución de los objetos obtenida por medio de un mapeo autoorganizado mostrando el superreino al que pertenece cada organismo.

7.2 Conclusiones

Podemos considerar conclusiones en dos ámbitos. El primero de ellos, el computacional, es mostrado en la sección siguiente, en tanto que las conclusiones en el ámbito de la biología se muestran en la sección 7.3

7.2.1 Conclusiones en el ámbito computacional

Se mostró que el mapeo autoorganizado es una herramienta de visualización de datos multidimensionales adecuada. Al comparar su desempeño con el de otras herramientas, se observó que es mejor que algunas de ellas para ciertos casos, como lo muestra la figura 6.6.
La identificación de las características que preservan las relaciones de vecindad en mayor medida, introducida en la sección 5.4, mostró ser una mejor técnica de eliminación de características que la propuesta por [61] cuando se elimina un número de variables grande, como se observa al comparar las tablas 5.2 y 5.5. Por otro lado, cuando se elimina un número pequeño de

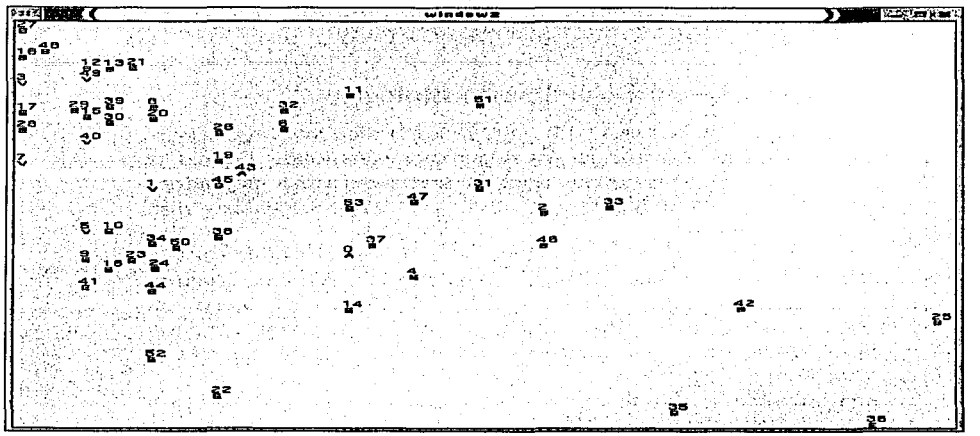


Figura 7.4: Distribución de los organismos obtenida por medio de programación genética mostrando el reino al que pertenecen.

variables, dicha técnica de eliminación también resulta ser una mejor alternativa.
La identificación de un subconjunto de las 64 características originales de tal modo que el mapeo autoorganizado para éstas mostrase un desempeño lo mas parecido al mapeo para el total de características resultó ser adecuada, lo que muestra que existen ciertas características que son más importantes que otras para el proceso de autoorganización de la red.
El algoritmo genético utilizado para identificar al subconjunto de características referido en el párrafo previo fue una alternativa adecuada, como se muestra en las tablas 5.6 y 5.7. En éstas se observa que el desempeño del mapeo aumenta a medida que las características que originan el espacio desde donde se lleva a cabo el mapeo autoorganizado mantienen una relación de vecindad para los objetos aalizados más parecida a la mostrada en el espacio original.
El mapeo obtenido por medio de programación genética introducido en la sección 6.3, mostró un mejor desempeño que el mapeo autoorganizado. Sin embargo, es importante recalcar el hecho que el mapeo autoorganizado no es supervisado, en tanto que el primero si lo es.
Adicionalmente, el hecho de ser inspirado en el funcionamiento de algunas regiones de la corteza cerebral, da al mapeo autoorganizado la posibilidad

de explicación, cuando se toma como modelo, de ciertos procesos mentales, mismos que las otras herramientas, incluida la programación genética, no pueden explicar.

7.2.2 Conclusiones en el ámbito biológico

Como se mencionó en la sección 1.2, una de las explicaciones para el uso de codones es el superreino al que los organismos pertenecen. Esto es, si dicha explicación fuese correcta, las bacterias analizadas serían mapeadas a regiones cercanas, en tanto que para los eucariontes, arqueobacterias y virus se tendría el mismo caso.
Analizando las figuras 7.3 y 7.4 podemos concluir que el uso de codones aparentemente no está relacionado con el reino taxonómico, lo que coincide con lo expresado por otros autores [72]. En dichs figuras, se observa que organismos pertenecientes a diferentes superreinos se encuentran en regiones cercanas, al mismo tiempo que organismos del mismo superreino son mapeados a regiones diferentes.
Otra de las posibles explicaciones para el uso de codones mencionada en la sección 2.5 fue el contenido de $G + C$ en el ADN. Analizando las figuras 7.5 y 7.6, podemos observar que los organismos con un contenido similar de $G + C\%$ se encuentran en regiones cercanas. De esta forma, lo que Sharp et al. [86] sostienen con referencia al contenido de $G + C$ como una posible explicación del uso de codones pudiera ser sustentada por los resultados aqui enunciados: los organismos con un contenido similar de $G + C$ hacen un uso de codones semejante, por lo que son mapeados a regiones cercanas.

7.3 Trabajo futuro

La distribución de los organismos que genera un mapeo autoorganizado podría ser mejor evaluada con base en cúmulos, en lugar de la vecindad entre los objetos más cercanos. Para ello, se podría recurrir a la identificación automática de éstos por diversas técnicas, tales como algoritmos genéticos [66]. De esta forma, la bondad de un mapeo podría ser expresada en términos de la homogeneidad de los organismos en un determinado cúmulo.
El análisis del uso de codones para *genes homólogos*[1] por medio del mapeo autoorganizado, daría a los especialistas información adicional a la que el análisis de las secuencias codificadoras completas genera.

[1] Genes homólogos son aquellos que codifican una proteína que realiza una tarea semejante en dos o más organismos.

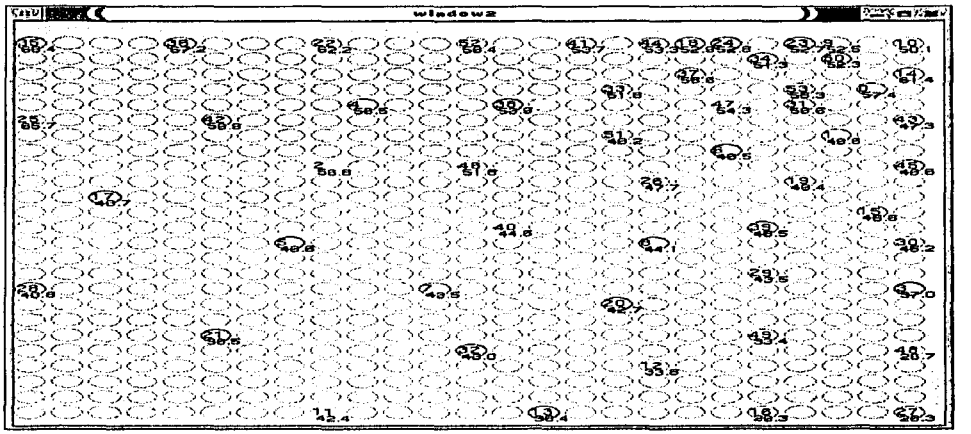


Figura 7.5: Distribución de los organismos obtenida por medio de un mapeo autoorganizado mostrando el contenido de G+C%.

Con respecto a la preservación de la topología, se ha mostrado evidencia en este trabajo que apoya la concepción de la preservación de las relaciones de vecindad como la idea principal detrás de la preservación de la topología. Realizar un análisis que defina la topología con base en las relaciones de vecindad contribuiría sin duda al mayor entendimiento de la teoría detrás del mapeo autoorganizado.

La modificación de la presentación de las neuronas en el mapeo autoorganizado propuesta por [61], conocida como *matriz-U*, otorga una mayor calidad a la visualización, pues el color que se le asigna a cada neurona da una mejor idea de la proximidad de los objetos en el espacio original y el presente trabajo puede ser extendido en esa dirección.

Como se mencionó a lo largo de este trabajo, algunas características son mas importantes que otras para el mapeo autoorganizado. Estas características pueden ser identificadas por medio de un algoritmo genético; sin embargo, el *peso informativo* de dichas características podría ser indicativo de la importancia de las mismas.

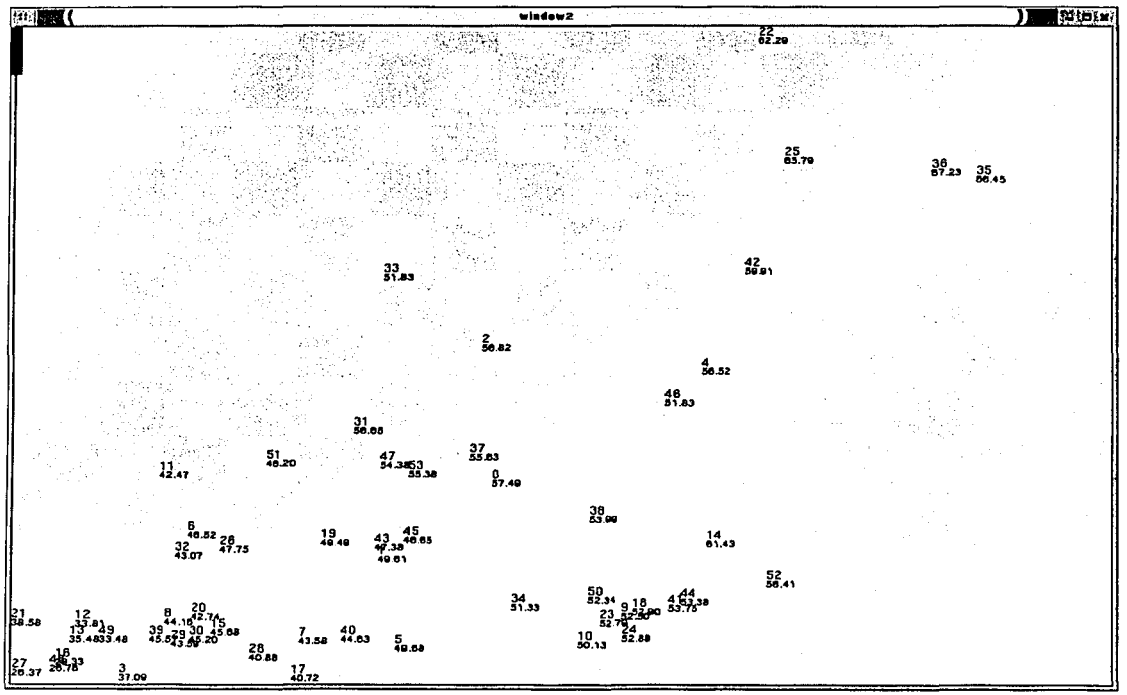


Figura 7.6: Distribución de los organismos obtenida por medio de programación genética mostrando el contenido de $G + C\%$.

Bibliografía

[1] Diamantaras, K. Principal component neural networks. Wiley. 1996.

[2] Alberts, B. et al. Essential Cell Biology. Garland Publishing. 2001.

[3] Armstrong, M. Topología Básica. Reverte. 1987.

[4] Deutsch, D. The psychology of music. Academic Press. 1982.

[5] Bentley Edward. Algebra Lineal y Ecuaciones Diferenciales. 1984.

[6] Devroye, Luc, et al. A Probabilistic Theory of Pattern Recognition. Springer. 1993.

[7] Bento Solange. DNA, Segredos e Misterios. Sarvier, 1998.

[8] Bauer H. Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps. IEEE Tran sactions on Neural Networks, Vol 3. No. 4. 1992.

[9] Brown, T. Essential Molecular Biology. Oxford University. 2000.

[10] Elliot, W. Biochemistry and Molecular Biology. Oxford University. 2001.

[11] Baldi P., Brunak S. Bioinformatics. The machine Learning Approach. MIT Press. 1998.

[12] Bystroff, C, Shao, Y. Fully automated ab initio protein structure prediction using I-SITES, HMMSTR and ROSETTA. Bioinformatics. Vol. 18 776-785. 2002.

[13] Cottrell, M., Fort. J, Pagés, G. Theoretical aspects of the SOM. Neurocomputing 21 (1998) 119-138.

[14] Bak, P. How nature works: The Science of Self-Organized Criticality. Copernicus Books. 1999.

[15] Ellis J., Morrison D. Schistosoma mansoni: patterns of codon usage and bias. Parasitology Vo. 110. 1995.

[16] Ellis J, Morrison D. Comparison of the patterns of codon usage and bias between Brugia, Echinococcus, Onchocerca and Schistosoma species. Parasitol research Vol. 81. 1995.

[17] Beantley, K. Neurocomputing. 1984.

[18] Abarbanel, H. Analysis of observed chaotic data. Springer. 1996.

[19] Cox, T. Multidimensional scaling. Chapman Hall. 2001.

[20] Groenen, B. Modern Multidimensional Scaling. Springer-Verlag. 1997.

[21] Bishop C., et al. GTM: The Generative Topographic Mapping. Neural Computing, 10, 215-234. 1998.

[22] Coveney, P. Highfield, R. Frontiers of Complexity. Faber and Faber. 1995.

[23] Coh, H. Fielding, M. Simulated annealing: searching for an optimal temperature schedule. Society for industrial and applied mathematics. Vol. 9 No.3 779-802. 1999.

[24] Currey K., Shapiro B. Secondary structure computer prediction of the poliovirus 5ñon-coding region is improved by a genetic algorithm Comput. Appl. Biosci. 1997 13: 1-12.

[25] Servicio de búsqueda del Instituto Nacional de Salud de los EE.UU. http://www.ncbi.nlm.nih.gov/Entrez/.

[26] Everitt, B. Cluster Analysis. Oxford University. 2001.

[27] Fausett, L. Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice Hall. 1995.

[28] Flanagan, J. Self-organized criticality and the self-organizing map. Physical Review E, Vol. 63, 036130.

[29] Flexer, A. On the use of self-organizing maps for clustering and visualization. Intelligent Data Analysis 5 (2001) 373-384.

[30] Flexer. A. Limitations of self-organizing maps for vector quantization and multidimensional scaling. Advances in neural information processing systems 9. MIT Press/Bradford Books. 1997

[31] Fukunawa, K. Introduction to statistical pattern recognition. Academic Press. 1993.

[32] Goldberg. Genetic algorithms in search, optimization and machine learning. Addison-Wesley. 1989.

[33] Grocock R., Sharp, P. Synonymous codon usage in Pseudomonas aeruginosa PA01. Gene. 2002. 289(1-2):131-9.

[34] Goodwin, B. Solé, R. Signs of Life. Basic Books. 2001.

[35] Hajeck, B. Cooling schedules for optimal annealing. Mathematics of operation research, Vol. 13, 311-329. 1988.

[36] Haykin, S. Neural Networks: A comprehensive foundation. Macmillan. 1995.

[37] Mehrotra, K. Elements of artificial neural Networks. MIT Press. 2001.

[38] Haynes, T. Sen. S. Evolving behavioral strategies in predators and prey. JCAI Workshop on Adaptation and Learning in Multiagent Systems. 1995.

[39] Heijliger, M. High-level synthesis scheduling and allocation using genetic algorithms. Proc. Asia and South Pacific Design Automation Conf. 1995.

[40] Hernández G, Velasco, J. El manantial escondido. Fondo de Cultura Económica.

[41] Holland, J. Emergence, from chaos to order. Helix Books. 1998.

[42] Holland, J. Adaption in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, 1992.

[43] Honkela, T. Self-Organizing Maps in Natural Language Processing. Tesis de doctorado. http://www.cis.hut.fi/ tho/thesis/

[44] Hunter, L. Artificial Intelligence and Molecular Biology. AAAI Press, 1994.

[45] Instituto Nacional de Estadística, Geografía e Informática. Distribución de cultivos en la región del Golfo de México. http:www.inegi.gob.mx.

[46] Jung, S. et al. Geometric fractal growth model for scale-free networks. Physical review letters E. Vol 65. 2002.

[47] Kanaya, S, et al. Systematization of Species-Specific diversity of Genes in Codon Usage. Japonese Society for Bioinformatics. 1996.

[48] Kanayaa, S. et al. Analysis of codon usage diversity of bacterial genes with a self-organizicharacterization of horizontally transferred genes with emphasis on the E. coli O157 genome. Gene, 276. 89 - 9.

[49] Kaski, S. SOM-based exploratory analysis of gene expression data. En N. Allinson, H. Yand J. Slack, editors, Advances in Self-Organizing Maps. Springer. 2001.

[50] Kaski, S. Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Cluestering. IEEE International Joint Conference on Neural Networks. 1998.

[51] Kaski, S. Lagus, K. Comparing Self-Organizing Maps. Proceedings of ICANN96, International Conference on Artificial Neural Networks. Alemania.

[52] Kaski, S., Kohonen, T. Statistical data Analysis by the Self-Organizing Map. Reporte quinquenal (1994-1998), Helsinki University of Technology.

[53] Kaski, S. Neighborhood Preservation in Nonlinear Projection Methods. Artificial Neural Networks, ICANN 2001, 458-491, Springer, Berlin, 2001.

[54] Base de datos de uso de codones. http://www.kazusa.or.jp/codon.

[55] Kiang, M. Extending the Kohonen self-organizing map networks for clustering analysis. Computational Statistics and Data Analysis, 2001. Vol: 38, 161-180.

[56] Kiviluoto, K. Topology Preservation in Self-Organizing Maps. IEEE Transactions on Neural Networks. 1996.

[57] Kohonen, T. Self-Organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69, 1982.

[58] Kohonen, T. Physiolocigal interpretation of the self-organizing map algorithm. Neural Networks. Vol: 6, 895-905.

[59] Kohonen, T. The self-organizing map, a possible model of brain maps. Medical Biological Engineering Computing, 34. 5-8.

[60] Kohonen, T. Hari, R. Where the abstract feature maps of the brain might come from. Trends in Neuroscience. Vol. 22, 135-139.

[61] Kohonen, T. Self-Organizing Maps. Springer Verlag. 1995.

[62] Kosko, B. Neural Networks and Fuzzy Systems. Prentice Hall, 1997.

[63] Kostiainen T, Lampinen, J. On the Generative Density Model in the Self-Organizing Map. Neurocomputing 48, 217-228.

[64] Koza, J. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press. 1992.

[65] Kuhn, T.S. La Estructura de las Revoluciones Científicas. Fondo de Cultura Económica. 1984.

[66] Kuri, A. Automatic clustering with Self-organizing Maps and genetic algorithms. MICAI 2002. Springer.

[67] Lange, K. Numerical analysis for staticians. Springer. 1999.

[68] Lewin, B. Genes VII. University of Oxford. 2000.

[69] Malsburg, von de, C. Face Recognition and Gender Determination. 1973.

[70] Mangiameli, P, et. al. A comparison of SOM neuronal network and hierarchical clustering methods. European Journal of Operation Research (1996) 402-417.

[71] Martínez, G, et al. Primitive Molecular Machine Scenario for the Origin of the Three Base Codon composition. Org. Life Evol. Bios. Vol. 29 203-214. 1999.

[72] Miramontes, P, Cedergren R. The puzzling origin of the genetic code. Trens in biochemical science. Vol. 21. 19996.

[73] MultiVariated Statistical Program. University of Sheffield. 2001.

[74] Nesti, C, et al. Phylogeny inferred from codon usage pattern in 31 organismos. Bioinformaticsi. V. 11, 167-171. 1995.

[75] Nikkila, P. et al. Analysis and visualization of gene expression data using Self-Organizing Networks, Special issue on New Developments on Self-Organizing Maps, 2002.

[76] Pandya, A. Macy, R. Pattern Recognition with Neural Networks in C++. 1999.

[77] Parmee, I. Evolutionary Design and Manufacture. Springer. 2000.

[78] Porter, T. Correlation between codon usage, regional genomic nucleotide composition, and amino acid composition in the cytochrome P-450 gene superfamily", Biochim. Biophys. Acta 1261, 394-400, 1995.

[79] Quinlan, R. Computer programs that learn. 1994.

[80] Reilly, R., Munakata, Y. Computational explorations in cognitive neuroscience. MIT Press. 2000.

[81] Rodríguez, K. Genetic programming in time series modelling: an application to meteorological data. Proceedings of the 2001 congress on evolutionary computation CEC2001.

[82] Rosenblum, L et al. Scientific Visualization. Academic Press. 1994.

[83] Schiffman, S. et. al. Introduction to Multidimensional Scaling. Academic Press. 1984.

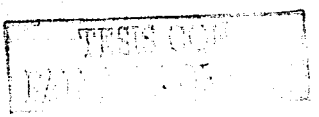
[84] Sharp, P. DNA sequence evolution: the sounds of silence. Transactiones of the Royal Society of Biological Sciences. 1995 Septiembre 29, 241-7.

[85] Andersson, S., Sharp P. Codon usage and base composition in Rickettsia prowazekii. J Mol Evol. 1996 May;42(5):525-36.

[86] Sharp, P. Matassi, G. Codon usage and genome evolution. Curr Opin Genet Dev. 1994. 851-60. Review.

[87] Siew, N. et al. MaxSub: an automated measure for the assessment of protein structure prediction quality. Bioinformatics. Vol. 16: 776-78. 2000.

[88] Smagt, van der, P. Introduction to Neural Networks. University of Amsterdam. 1993.



[89] Steeb, W. Nonlinear Workbook: Chaos, Fractals, Cellular Automata, Neural Networks, Genetic Algorithms, Gene Expression Programming, Wavelets, Fuzzy Logic - With C++, Java. World Scientific Pub Co. 2002.

[90] Teu, T, González, R. Pattern Recognition Principles. Addison-Wesley. 1984.

[91] Venna, J, Kaski, S. Neighborhood Preservation in Nonlinear Projection Methods.

[92] Vesanto, J. SOM-Based Data Visualization Methods. http://www.cis.hut.fi/projects/ide/publications/fulldetails.htmlvesanto99ida. 1999.

[93] Vesanto, J, Alhoniemi, E. Clustering of the Self-Organizing Map.

[94] Villmann T et al. Topoloy Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement. IEEE Transactions on Neural Networks. 1997.

[95] Villmann, T et al. A New Quantitative Measure of Topology Preservation in Kohonen's Feature Maps. IEEE Transactions on Neural Networks, 1994.

[96] Wagner, A. How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps. Bioinformatics. Vol. 17 1183-97. 2002.

[97] Wang, H et al. Analysis of Codon Usage Patterns of Bacterial Genomes Using the Self-Organizizing Map.

[98] Wolfram, S. Resource Library. http://mathworld.wolfram.com.

[99] Zuben, von, F. Vox populi: evolutionary computation for music evolution. http://www.ici.org.br/invencao/papers/Manzolli.htm

[100] Zurada, J. Introduction to Artificial Neural Systems. West Publishing. 1994.