Universidad Nacional Autónoma de México

Centro de Física Aplicada y Tecnología Avanzada

Métodos para el Estudio de Redes en Sistemas Biológicos

TESIS

Que para obtener el grado de

Licenciado en Tecnología

P R E S E N T A

Marcos Emmanuel González Laffitte

Directora de tesis

Dra. Maribel Hernández Rosales

Instituto de Matemáticas, UNAM

UNAM Campus Juriquilla, Querétaro, 2019

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

CENTRO DE FÍSICA APLICADA Y TECNOLOGÍA AVANZADA
FACULTAD DE ESTUDIOS SUPERIORES, CUAUTITLÁN

LICENCIATURA EN TECNOLOGÍA

Votos Aprobatorios

COMITÉ ACADÉMICO DE LA
LICENCIATURA EN TECNOLOGÍA

Presente

En cumplimiento del Artículo 26 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la Tesis de título

Métodos para el Estudio de Redes en Sistemas Biológicos

que realizó el (la) pasante

Marcos Emmanuel González Laffitte

con número de cuenta: 311260331, bajo la opción de titulación por Tesis y Examen profesional en la Licenciatura en Tecnología.

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro **VOTO APROBATORIO.**

	NOMBRE	FIRMA
PRESIDENTE	Dra. Amanda Montejano Cantoral.	
VOCAL	Dra. Cristy Leonor Azanza Ricardo.	
SECRETARIO	Dra. Beatriz Marcela Millán Malo.	
1er. SUPLENTE	Dr. Marco Tulio Angulo Ballesteros.	
2º SUPLENTE	Dra. Maribel Hernández Rosales.	

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"

UNAM, Campus Juriquilla, Qro. a 15 de febrero de 2019.

Contenido

Resumen

Una red es un modelo matemático, y la disciplina que estudia estos modelos es llamada ciencia de redes [1]. El formalismo matemático que sustenta el análisis de las redes se encuentra principalmente en la teoría de grafos [2, 3, 4]. De este modo, cada red puede ser considerada como un grafo cuyas propiedades tienen cierta interpretación, dependiente del problema a estudiar. El trabajo en esta tesis comprende el análisis de tres problemas tecnológicos relacionados con redes que modelan sistemas en biología. Primero se estudian algunas propiedades de redes ecológicas, prestando especial atención a aquellas propiedades relacionadas directamente con el agrupamiento de los organismos en estas redes y las interacciones entre ellos [5]. Posteriormente se realiza un análisis en torno al problema de detección de comunidades en grafos [6] y su aplicación sobre redes neuronales biológicas. Por último, se evalúa la implementación de sistemas de aprendizaje automático [7] en la clasificación de redes de regulación genética que modelan cáncer de mama.

Abstract

A network is a mathematical model, and the discipline that studies these models is called network science [1]. The mathematical formalism that sustains these models can be found mainly in graph theory [2, 3, 4]. In this sense, each network can be considered as a graph whose properties have some interpretation, which depends on the problem in hand. This thesis includes the analysis of three technological problems related to networks that model biological systems. First, a set of properties of ecological networks is studied, paying special attention to those properties directly related to the grouping of organisms in these networks and the interactions between them [5]. Later, an analysis is made around the problem of community detection in graphs [6] and its application on biological neural networks. Finally, we evaluate the implementation of machine learning systems [7] in the classification of gene regulatory networks that model breast cancer.

Agradecimientos

Gracias a mi familia, en especial a mi madre María Eugenia y a mi hermano Pablo,
que me han dado siempre su apoyo incondicional.

Gracias también a Ali, con quien he compartido la mayor
parte de mi tiempo y experiencias en la licenciatura.

Y gracias a todos los integrantes del grupo de Bioinformática del Instituto
de Matemáticas de la UNAM, campus Juriquilla, del que he sido
parte desde su formación.

El desarrollo de esta tesis y mi asistencia al curso:
First Course on Computational Systems Biology of Cancer 2018, Curie Institute, Paris, France,
fue posible gracias al proyecto 254206:
"DNA-mutation simulation of tumor growth and reconstruction of cancer evolution." (2016-2019),
financiado por el Fondo Ciencia Básica de CONACyT.

Para el desarrollo computacional de esta tesis se contó con acceso a servidores provistos por el
Laboratorio Nacional de Visualización Científica Avanzada,
disponiendo siempre del atento apoyo de
Luis Aguilar, Alejandro de León, Carlos S. Flores y Jair García,
personal técnico de dicha institución.

Capítulo 1

Introducción

Una red es un modelo matemático, y la disciplina que se encarga de estudiar estos modelos es llamada ciencia de redes [1]. El formalismo matemático que sustenta el análisis de las redes se encuentra principalmente en la teoría de grafos [2, 3, 4]. De este modo, cada red puede ser considerada como un grafo cuyas propiedades tienen una interpretación, dependiente del problema a estudiar. Con una red se busca fundamentalmente esquematizar dos propiedades de un sistema: los elementos que lo conforman y las interacciones que existen entre esos elementos.

Las redes tienen aplicaciones tecnológicas sobre diversos tipos de sistemas: sociales, biológicos, de información, físicos, químicos y en ingeniería [1]. En esta tesis se presenta el análisis de tres problemas tecnológicos relacionados con redes que modelan sistemas en biología. Primero se estudian algunas propiedades de redes ecológicas, prestando especial atención a aquellas características relacionadas con el agrupamiento de los organismos en estas redes y las interacciones de colaboración o exclusión entre ellos [5]. Posteriormente realizamos un análisis en torno al problema de detección de comunidades en grafos [6] y su aplicación sobre redes neuronales biológicas [8]. Por último, se evalúa la implementación de sistemas de aprendizaje automático [7] en la clasificación de redes de regulación genética que modelan cáncer de mama [9].

El estudio de tres problemas distintos permite explorar diversos métodos para el análisis de propiedades de redes, proporcionando un amplio enfoque y formación en el modelado de sistemas biológicos por medio de ellas. A continuación se describe cada una de las problemáticas abordadas en esta tesis y posteriormente se detallan las respectivas propuestas de solución o análisis desarrolladas.

1.1 Justificación

a) Análisis de propiedades de redes ecológicas: Se estudiaron las propiedades de redes de interacciones ecológicas de un tipo particular de ecosistema llamado tapete microbiano [10]. Estos son estructuras laminares compuestas por una combinación de colonias microbianas y sus respectivos sustratos. Nuestro análisis comprendió un proyecto cuyos resultados se encuentran ya publicados en [11]. Este se desarrolló en colaboración con la Dra. Valerie de Anda del Instituto de Ecología de la UNAM, quien llevó a cabo la inferencia de las redes haciendo uso del programa MetaMIS [12] sobre datos de metagenómica de estos ecosistemas. Los tapetes estudiados son originales de un estanque cercano a la Laguna Churince en la reserva ecológica de la cuenca de Cuatro Ciénegas (CCC), ubicada en el estado mexicano de Coahuila. Dicha reserva, y en particular la Laguna Churince, se han visto sujetas a desecaciones debidas a la actividad humana, esencialmente por el uso del agua en el desarrollo agropecuario de la región [13]. Como CCC se caracteriza principalmente por su alto contenido de especies endémicas, fue de nuestro interés evaluar el impacto que ha tenido el cambio en los niveles de agua sobre las colonias en los tapetes. Para analizar esto, prestamos especial atención a propiedades de redes relacionadas con el agrupamiento de los organismos, como son: densidad, grado promedio y coeficiente de agrupamiento promedio, y desarrollamos la detección de los subgrafos estadísticamente significativos llamados *motivos de red*.

b) Detección de comunidades en redes: El problema de detección de comunidades en grafos [6] conlleva particionar el conjunto de vértices de un grafo en grupos llamados comunidades, de forma que existan más aristas entre vértices de la misma comunidad que entre vértices de distintas comunidades. De esta forma, las comunidades pueden representar agrupamientos significativos dependiendo del caso de estudio [1]. Actualmente existen múltiples métodos de detección de comunidades, sin embargo ninguno de ellos ha sido ampliamente aceptado por distintos motivos, que involucran desde elevados tiempos de ejecución hasta ciertos problemas respecto a los módulos que con ellos se detectan. Uno de los métodos más usados conlleva evaluar la relevancia de una partición por medio de la medida conocida como modularidad [14], siendo que dadas múltiples particiones, aquella que maximice la modularidad representará la partición más significativa. No obstante, se ha mostrado que la evaluación de la calidad de las particiones por medio de la modularidad tiene un problema llamado límite de resolución [15]. Este implica que, dada una partición con modularidad óptima, es posible encontrar sub-grupos más significativos dentro de cada uno de los propuestos por la partición original, siendo que la partición del conjunto de vértices en estos sub-grupos proporcionaría una menor modularidad. Motivados por estudiar las alternativas al problema del límite de resolución, y apoyados en los trabajos de Newman y Girvan [14, 16], desarrollamos y evaluamos un método de detección de comunidades basado también en la optimización de la modularidad, pero relacionando esta medida con la definición de probabilidad condicional [17]. Finalmente, aplicamos nuestro análisis a la red conformada por las interacciones neuronales del sistema nervioso del nematodo *C. Elegans*, siendo que las comunidades en esta han sido ampliamente estudiadas. Así, pudimos comparar los agrupamientos detectados por nuestro método contra lo que describe la literatura, tanto en el contexto de detección de comunidades [18, 19], como en referencia a las características biológicas de dicha red [19, 20, 21].

c) Aprendizaje automático con propiedades de redes de regulación genética en cáncer de mama: Existen enfermedades propiciadas por múltiples factores tanto genéticos como ambientales. Estas son comúnmente llamadas enfermedades complejas [22]. Ejemplo conciso de estas patologías es el cáncer de mama [9]. Dado que esta afección es difícil de tratar, su detección temprana cobra importancia cuando se busca prevenir el incremento de la mortalidad debida a ella [23]. En este trabajo nos enfocamos en estudiar la capacidad de predicción que tiene un sistema de aprendizaje automático [7], en específico de redes neuronales artificiales [24], sobre la clasificación de redes de regulación genética que modelan cáncer de mama [25, 26]. Con esto, buscamos explorar las bases para el estudio de la viabilidad que tiene la aplicación de esta metodología en la detección temprana del cáncer de mama.

1.2 Resumen general de las hipótesis

a) Análisis de propiedades de redes ecológicas: Dado que una reducción en la humedad en el entorno de los tapetes produce una disminución en los nutrientes de los que estos disponen [10, 11], y dado que la pérdida de nutrientes en un ecosistema propicia interacciones de competencia entre los organismos que en él habitan [5], se espera ver en estas redes un incremento en la proporción de interacciones de competencia conforme disminuyen los niveles de humedad en las muestras de las que fueron inferidas.

b) Detección de comunidades en redes: Basaremos nuestro método en la teoría de probabilidad [17], y partiremos de los trabajos que hablan sobre la modularidad y su límite de resolución [6, 14, 15, 16, 27] para evaluar cuatro hipótesis: a) dado un grafo, existe más de una partición de sus vértices en agrupamientos que sean significativos [6], b) debe existir un camino de aristas entre toda pareja de vértices dentro de la misma comunidad, c) el número de aristas que conectan vértices dentro de una misma comunidad debe ser mayor, tanto al número de aristas que conectan vértices en distintas comunidades, como al número de aristas que conectarían vértices en una misma comunidad si el grafo en cuestión hubiera sido generado aleatoriamente preservando su secuencia de grado (hipótesis originalmente planteada en [14, 27]), y finalmente d) las aristas con mayor centralidad son aristas que unen vértices en distintas comunidades (hipótesis originalmente planteada en [16]).

c) Aprendizaje automático con propiedades de redes de regulación genética en cáncer de mama: Dado que las interacciones genéticas se ven alteradas en la presencia del cáncer [28], suponemos que las redes que modelan interacciones genéticas sujetas al cáncer de mama presentan propiedades particulares que las permiten diferenciar de redes aleatorias [9]. Para evaluar esto utilizaremos un sistema de redes neuronales artificiales, bajo la suposición de que existe una función no lineal [24] entre ciertas propiedades de las redes y su clasificación como redes relacionadas al cáncer o redes aleatorias. Aunado a esto suponemos que estas redes neuronales artificiales son capaces de aproximar dicha función al aprender a reconocer a las redes de regulación por medio de aprendizaje automático supervisado [7].

1.3 Resumen general de los objetivos

I) Objetivos generales

a) Análisis de propiedades de redes ecológicas: Analizar el efecto de la reducción de humedad sobre las redes de interacciones de colaboración y exclusión entre los organismos en tapetes microbianos originales de la reserva ecológica de la Cuenca de Cuatro Ciénengas.

b) Detección de comunidades en redes: Proponer una formulación para la modularidad por medio de la definición de probabilidad condicional, y desarrollar y evaluar un método para la detección de comunidades en grafos basado en dicha formulación. Por último, comparar las comunidades detectadas con nuestro método sobre la red de interacciones neuronales del sistema nervioso de *C. Elegans* contra las que se describen en la literatura [19, 21].

c) Aprendizaje automático con propiedades de redes de regulación genética en cáncer de mama: Evaluar la capacidad predictiva de un sistema de redes neuronales artificiales en la clasificación de redes de regulación genética, buscando diferenciar redes que modelan cáncer de mama de una colección de redes generadas aleatoriamente.

II) Objetivos específicos

a) Análisis de propiedades de redes ecológicas:
1. Implementar en el lenguaje de programación Python, un programa que nos permita obtener propiedades de grafos como son: densidad, coeficiente de agrupamiento promedio y grado promedio.
2. Determinar el valor de dichas propiedades sobre las redes de tapetes microbianos de CCC.
3. Obtener los motivos de red por medio del software MFinder.
4. Comparar los resultados obtenidos para tres tipos de subredes: que contengan todas las interacciones, tanto de exclusión como colaboración, aquellas conformadas solo por interacciones de exclusión y aquellas formadas únicamente por interacciones de colaboración.

b) Detección de comunidades en redes:
1. Definir a la modularidad usando la definición de probabilidad condicional, y desarrollar un método de detección de comunidades en redes basado en la optimización de la modularidad.
2. Evaluar nuestro método de detección de comunidades sobre las redes aleatorias llamadas modelos-LFR [29] y Planted-Graphs [14, 6], ya implementados en la biblioteca NetworkX [30] del lenguaje Python.
3. Analizar los agrupamientos detectados con dicho método al aplicarlo a la red de amistad del Club de Karate de Zachary [14], comparando nuestros resultados contra la información empírica de esta red [31].
4. Desarrollar una comparación de comunidades en la red neuronal de *C. Elegans*.

c) Aprendizaje automático con propiedades de redes de regulación genética en cáncer de mama:
1. Establecer la definición de una red neuronal artificial con respecto a la definición de digrafo acíclico.
2. Construir un programa de creación de redes neuronales artificiales basado únicamente en la paquetería básica del lenguaje Python, y por medio de este entrenar a una red en la reproducción de la función booleana XOR.

3. Desarrollar el entrenamiento de redes neuronales artificiales de la biblioteca SKLearn [32], en la clasificación de redes de regulación genética relacionadas al cáncer de mama y de redes aleatorias.
4. Evaluar el aprendizaje de estas redes neuronales artificiales por medio de la proporción de sus aciertos positivos en la clasificación de las redes de regulación genética y redes aleatorias.

1.4 Resumen general de la metodología

a) Análisis de propiedades de redes ecológicas: Para este análisis se trabajó con redes proporcionadas por la Dra. Valerie de Anda del instituto de Ecología de la UNAM [11]. En estas, cada interacción se encuentra etiquetada con alguno de dos signos, positivo (+) o negativo (-), que indican si la interacción ecológica es de cooperación o exclusión respectivamente. Se cuenta originalmente con 12 redes, llamadas consenso, simples y con signos, una por cada nivel taxonómico: familia, orden, clase, y filo, inferidas a partir de datos de metagenómica de tres sitios de muestreo dentro del estanque cercano a la laguna Churince [13]. De cada red consenso se extrajeron otras dos, una compuesta puramente por relaciones positivas y otra por relaciones negativas. Además, para cada nivel taxonómico se constituyó una red llamada global, correspondiente a la unión de las redes consenso a un mismo nivel en los tres sitios de muestreo, que a su vez fue nuevamente separada en una red global de exclusión y otra de colaboración. Con todo lo anterior, se obtuvieron 48 redes que formaron la base para el análisis de redes presentado en [11]. Para esta tesis se estudiaron primeramente las proporciones de interacciones de exclusión y colaboración, respecto a la cantidad de interacciones totales, sobre las 12 redes consenso. Luego se determinaron sobre estas mismas 12 redes los valores para las propiedades de densidad, coeficiente de agrupamiento y grado promedio [1]. En seguida comparamos estos valores contra el promedio de sus equivalentes en 100 redes aleatorias con el mismo número de vértices y aristas, generadas por medio de la biblioteca NetworkX, con propósito de evaluar la relevancia de estas propiedades respecto a un modelo nulo [1]. Se recurrió por último al concepto de motivo de red [33, 34] para analizar la presencia de subredes estadísiticamente significativas sobre estos sistemas, determinando la existencia de estos motivos por medio del programa MFinder [35]. Con esto, se estudió la formación de patrones recurrentes constituidos por interacciones de exclusión y colaboración en las 48 redes.

b) Detección de comunidades en redes: En esta sección se formuló la medida llamada modularidad por medio de la definición de probabilidad condicional [17]. Luego se desarrolló un método de detección de comunidades [6] basado en dicha formulación [14], definiendo sobre este una alternativa al límite de resolución [15]. Se evaluó este método utilizando los grafos aleatorios de prueba (Benchmark Graphs) llamados Planted-Graphs [14] y modelos-LFR [29], comúnmente utilizados para evaluar trabajos de detección de comunidades en redes, y actualmente implementados en la biblioteca NetworkX [30] del lenguaje de programación Python. Comparamos en ambos casos los módulos detectados por nuestro algoritmo contra los detectados por medio del algoritmo Clauset-Newman-Moore [36], mientras que para los modelos-LFR se compararon nuestros resultados también contra la partición ideal que estos modelos proporcionan [29, 30]. Como medida de similitud entre particiones se utilizó la información mutua normalizada, siguiendo lo planteado en [29]. Después se aplicó este procedimiento a una red de relaciones de amistad, ampliamente utilizada en el contexto de detección de comunidades, conocida como El Club de Karate de Zachary [14], comparando nuestros resultados contra la información empírica que se posee respecto a esta [31]. Por último se aplicó también nuestro análisis a la red conformada por las interacciones neuronales del sistema nervioso del nematodo *C. Elegans*, comparando los agrupamientos detectados por nuestro método contra lo que describe la literatura, tanto en el contexto de detección de comunidades [18, 19], como en referencia a las características biológicas propias de esta red [19, 20, 21], nuevamente haciendo uso de la información mutua normalizada.

c) Aprendizaje automático con propiedades de redes de regulación genética en cáncer de mama: Inicialmente se estudiaron los principios del aprendizaje automático [7], con lo que luego se desarrollaron definiciones propias de lo que se entiende por una red neuronal artificial capaz de aprender por medio del algoritmo de retropropagación. Dicho planteamiento esta basado en el trabajo de Rojas [24], y lo desarrollamos respecto a la definición de grafo acíclico dirigido.

Por medio de nuestras definiciones, se programó un sistema de construcción de redes neuronales utilizando la paquetería básica del lenguaje Python, poniéndolo a prueba al entrenar una red para aprender la función booleana XOR. Posteriormente, se desarrolló un análisis de clasificación de 48 redes dirigidas, no pesadas y sin lazos ni multiaristas, de regulación genética en humano relacionadas al cáncer de mama. Estas fueron obtenidas a partir de la base de datos The Cancer Network Galaxy [37], desarrollada y mantenida por la Universidad de Tokio. Para cada una de estas redes se formularon otras dos aleatorias. Una de ellas preservando el orden y tamaño de su contraparte real, mientras que la otra preservó la secuencia de grado de la red que modela cáncer, generada a partir del modelo Havel Hakimi para redes dirigidas, también existente ya en la biblioteca NetworkX. Así, se buscó clasificar estas 144 redes por medio de un sistema de redes neuronales artificiales, generadas con la biblioteca SKLearn del lenguaje Python. Las parejas de entrenamiento y prueba [7, 24] de estos sistemas consistieron en propiedades de redes como entrada y un identificador binario como objetivo, que permite distinguir entre redes de regulación y redes aleatorias. Finalmente, evaluamos el aprendizaje de estos sistemas computacionales por medio de la exactitud de la clasificación [7], definida como la proporción de aciertos en una colección de pares de prueba.

El desarrollo computacional de toda la tesis se llevó a cabo en lenguaje de programación Python, por medio de la paquetería básica de dicho lenguaje y haciendo particular uso de las bibliotecas: NetworkX [30], Matplotlib [38], Numpy [39], MPMath [40] y SKLearn [32]. Para ejecutar el análisis se contó con acceso a los Clusters de Laboratorio Nacional de Visualización Científica (LAVIS) [41], del Instituto de Neurobiología la UNAM, campus Juriquilla. Gracias a dichos servidores, se ejecutaron múltiples procesos al mismo tiempo, lo que fue de utilidad para ejecutar de forma eficiente el análisis de los tres problemas tecnológicos planteados.

1.5 Organización de esta tesis

El contenido de esta tesis se encuentra condensado en cuatro capítulos, además de esta introducción y una conclusión general. En el primero de ellos se tratan los conceptos referentes a la teoría de grafos, que es fundamento general de los posteriores tres capítulos. Después de esto, presentamos el análisis hecho sobre redes de interacciones ecológicas de la Cuenca de Cuatro Ciénegas, seguido por el estudio de detección de comunidades por optimización de la modularidad y su aplicación a la red neuronal de *C. Elegans*. Se presenta por último la investigación hecha sobre redes de regulación genética relacionadas al cáncer de mama.

Cada capítulo se desarrolla planteando el problema tecnológico a estudiar, acompañado por la solución o aproximación propuesta para este. Posteriormente, se da una introducción al tema teórico sobre el que se desarrolla cada capítulo de manera particular. Después se muestra un marco teórico sobre los conceptos en torno a los que gira el análisis, para finalmente condensar cada proyecto en las secciones de metodología, resultados, discusión y recomendaciones.

1.6 Artículo publicado como parte del trabajo de esta tesis

De Anda Valerie, Zapata-Peñasco Icoquih, Blaz Jazmín, Poot-Hernández Augusto Cesar, Contreras-Moreira Bruno, González-Laffitte Marcos, Gámez-Tamariz Niza, Hernández-Rosales Maribel, Eguiarte Luis E., Souza Valeria. *Understanding the Mechanisms Behind the Response to Environmental Perturbation in Microbial Mats: A Metagenomic-Network Based Approach*. Frontiers in Microbiology, vol. 9, 2018. `https://www.frontiersin.org/article/10.3389/fmicb.2018.02606`.

Capítulo 2

Teoría de grafos

En este capítulo se presentan los conceptos matemáticos [2, 3, 4] relacionados con los problemas tecnológicos planteados en esta tesis. Iniciamos estableciendo las definiciones de grafo no dirigido y de grafo dirigido. Posteriormente se presentan las propiedades que sobre estos se pueden analizar.

2.1 Definiciones básicas de teoría de grafos

Definición 2.1.1. *Un **grafo no dirigido** se define como una pareja ordenada de conjuntos $G = (V, E)$, tales que $V \neq \varnothing$ aunque E si puede ser vacío, donde los elementos del conjunto V son llamados **vértices**, mientras que el conjunto E se encuentra conformado por parejas no ordenadas de vértices distintos entre sí, llamadas **aristas**. Comúnmente se denota V = V(G) y E = E(G) cuando se habla de más de un grafo al mismo tiempo. En ocasiones y mientras se tenga un solo grafo G en contexto, nos referiremos simplemente como V al conjunto de vértices de G y como E al conjunto de sus aristas.*

Dado un grafo no dirigido $G = (V, E)$, se dice que dos vértices $u, v \in V$ son **adyacentes** si y solo si $\{u, v\} \in E$, y a la vez la arista $\{u, v\}$ se dice **incidente** tanto a u como a v. Cuando dos vértices son incidentes también podemos decir que estos se encuentran relacionados, unidos o asociados. En el caso en que dos vértices no estén conectados o unidos por una arista entonces se dirá que esos vértices son **independientes** uno de otro.

En vez de enumerar cada elemento de un grafo para hablar de él, es común dar una representación visual para este. En dicha representación, los vértices se muestran como puntos, y las aristas como lineas que unen puntos (fig. 2.1). Recurriremos a tal representación a lo largo de este trabajo para esclarecer definiciones y esquematizar conceptos.

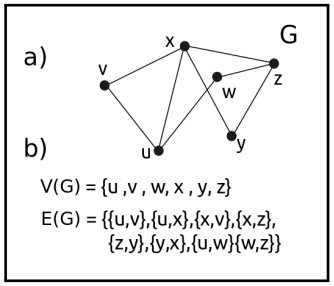

Figura 2.1: a) Representación visual y b) enumeración de los elementos de un grafo no dirigido G.

Por otro lado, un grafo dirigido se diferencía de un grafo no dirigido únicamente en que el conjunto de aristas es sustituido por un conjunto de parejas ordenadas de vértices, quedando definido como sigue.

Definición 2.1.2. *Un **grafo dirigido** o **digrafo** se define como una pareja ordenada de conjuntos $G = (V, A)$, tales que $V \neq \varnothing$ pero A si puede ser vacío, donde los elementos del conjunto V son llamados **vértices**, mientras que el conjunto A se encuentra conformado por parejas ordenadas de vértices distintos entre sí, llamadas **arcos** o **flechas**. Al igual que con grafos no dirigidos, se puede denotar como $A(G)$ al conjunto de las flechas de un grafo dirigido G y como $V(G)$ al de sus vértices, si es que se habla de más de un digrafo al mismo tiempo.*

En el caso de los grafos dirigidos se debe prestar especial atención al orden en que se presentan los vértices en cada flecha del digrafo. Por ejemplo, dados dos vértices distintos u, v en un grafo dirigido, se tiene que $(u, v) \neq (v, u)$. Es decir que hay dos flechas diferentes que pueden formarse entre los vértices u y v. Debido a lo anterior, para toda flecha $a = (u, v)$ en un digrafo, se dice que el vértice u es adyacente hacia el vértice v, mientras que v se distinguirá como adyacente desde u, y al mismo tiempo se dirá que la flecha a entra o apunta hacia v y sale o apunta desde u. Por esta misma razón, las parejas ordenadas de vértices en un digrafo se representan visualmente como *flechas* (fig. 2.2) que apuntan desde un vértice hacia otro.

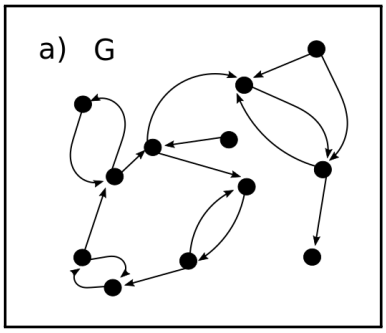

Figura 2.2: Representación visual de un digrafo G.

Cabe mencionar que es posible modificar las definiciones anteriores para permitir la existencia de aristas o flechas conectando a un vértice consigo mismo. Cuando esto sucede, se dice que tal grafo posee **lazos**. De la misma forma, también se pueden construir grafos donde se repitan aristas o flechas entre un mismo par de vértices. En este último caso se dice que el grafo tiene **multiaristas** (fig. 2.3). Así, un grafo que no tiene lazos ni multiaristas es comúnmente llamado **grafo simple**. En este trabajo trataremos únicamente con grafos sin lazos ni multiaristas, por lo que se deberá asumir en todo momento que las definiciones en él refieren a grafos simples, a no ser que se mencione explícitamente lo contrario.

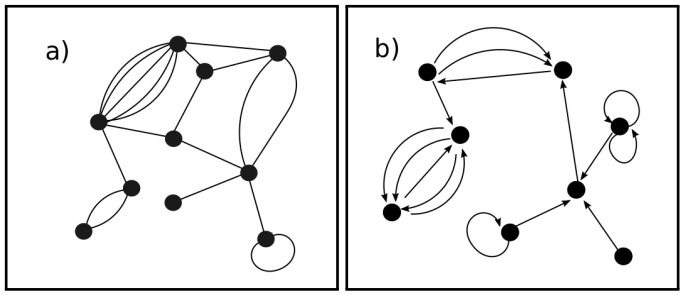

Figura 2.3: Ejemplos de a) un grafo no dirigido y b) un digrafo, ambos con lazos y múltiples aristas.

En ocasiones, será de nuestro interés suprimir la dirección de las flechas sobre un grafo dirigido G. En otras palabras, buscaremos asociar a este un grafo no dirigido cuyas aristas representen a todos los pares de vértices entre los que existe por lo menos una flecha en G. Esto queda plasmado en la siguiente definición.

Definición 2.1.3. *Dado un digrafo G con conjunto de vértices V, se define como **grafo subyacente** de G al grafo no dirigido G_{sub}, con conjunto de vértices también en V, y conjunto de aristas $E(G_{sub})$ tal que la arista $\{u, v\}$ pertenece a $E(G_{sub})$ si y solo si alguna o ambas flechas (u, v), (v, u) están en $E(G)$, para todo par de vértices $u, v \in V$, (fig. 2.4).*

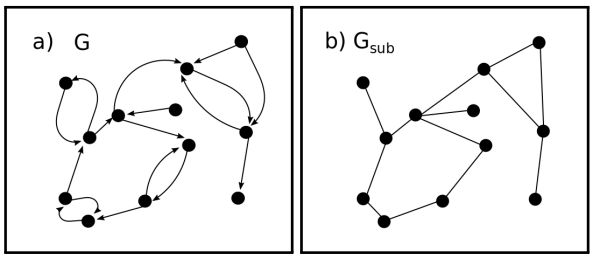

Figura 2.4: a) Un digrafo G y b) su correspondiente grafo subyacente G_{sub}.

Dados cualesquiera dos grafos G y H, si es posible establecer un mapeo biyectivo $\phi : V(G) \to V(H)$, tal que bajo ϕ se preserve en H la relación de adyacencia de los vértices de G, es decir, denotando por uv a una arista o flecha en G, si se cumple que $uv \in E(G) \iff \phi(u)\phi(v) \in E(H)$ (fig. 2.5), entonces se dice que G y H son **isomorfos**, y a la vez ϕ es conocido como un **isomorfismo** entre G y H.

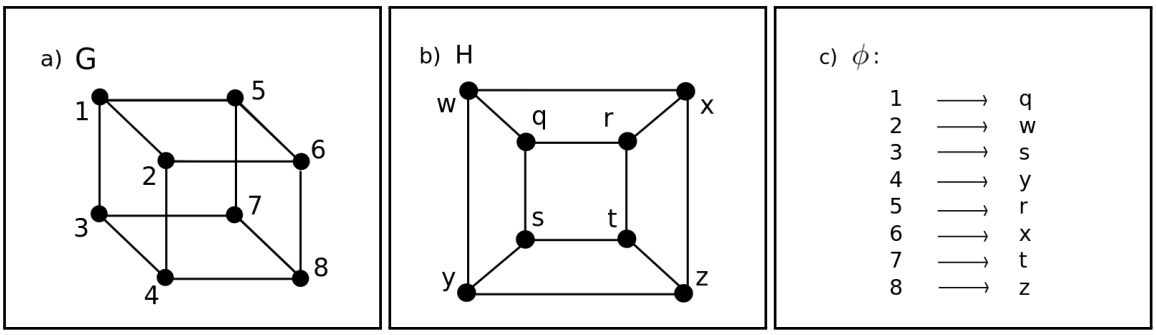

Figura 2.5: a) Un grafo no dirigido G, b) otro grafo no dirigo H, y c) un isomorfismo entre G y H .

2.2 Subgrafos y conectividad

A continuación se presentan las definiciones correspondientes a subgrafos y caminos, y con la ayuda de estas se desarrolla la definición de grafo no dirigido conexo. Después de esto se analizan las definiciones relativas a la conectividad en grafos dirigidos. Dado que existen algunas definiciones similares entre grafos no dirigidos y dirigidos, para evitar repetir información innecesariamente, en ocaciones se denotará como $G = (V, E)$ a un grafo cualquiera, remarcando el uso de esta notación en su momento.

Definición 2.2.1. *Dado un grafo G = (V, E) (dirigido o no dirigido), se llama* ***subgrafo*** *de G a cualquier grafo* $G' = (V', E')$ *tal que* $V' \subset V$ *y* $E' \subset E$.

Definición 2.2.2. *Sean G = (V, E) un grafo y* $G' = (V', E')$ *un subgrafo de G (ambos dirigidos o no dirigidos), G' es llamado* ***subgrafo inducido*** *de G por V' si* $uv \in E \iff uv \in E' \quad \forall u, v \in V'$, *y es denotado como* $G' = G[V']$.

Un importante concepto relacionado al de subgrafo, surge cuando buscamos conocer si un grafo se puede representar visualmente como una única componente, o si tiene que ser dibujado en partes no conectadas entre sí (fig. 2.6). Para comprender esto mejor, primero daremos la definición de camino en un grafo no dirigido y con esto estableceremos lo que se entiende por partes conectadas. Posteriormente estableceremos los conceptos análogos para grafos dirigidos.

Definición 2.2.3. *Sea G = (V, E) un grafo no dirigido y sea* $v_1, v_2, v_3, ..., v_n$ *una sucesión S de* n *vértices de G diferentes entre sí. Si respecto a S ocurre que* $e_i = \{v_i, v_{i+1}\} \in E$ *para* $i = 1, 2, 3, ..., n-1$, *entonces nos referiremos a S como un* ***camino*** *con longitud* $n - 1$, *formado entre los vértices* v_1 *y* v_n. *Si* $v_1 = v_n$, *pero todos los demás vértices en S siguen siendo distintos entre sí, entonces S se conoce como* ***ciclo***.

Definición 2.2.4. *Dado un grafo no dirigido G = (V, E), si existe al menos un camino entre toda pareja de vértices en V, entonces G se dice* ***conexo*** *(fig. 2.6). De otra forma, se dice que G es* ***no conexo*** *o que es* ***disconexo***.

Definición 2.2.5. *Sea G = (V, E) un grafo no dirigido y conexo, se define como la* ***distancia*** $d(u, v)$ *entre cualesquiera dos vértices* u *y* v *en G, a la menor longitud de entre todos los caminos existentes entre* u *y* v. *Se puede verificar que esta distancia constituye efectivamente una métrica sobre el conjunto de vértices del grafo [2].*

Definición 2.2.6. *Sea G = (V, E) un grafo no dirigido, se define como* ***componente conexa*** *de G a todo subgrafo inducido G' = (V', E') tal que G' es conexo, pero* $G[V' \cup \{w\}]$ *no es conexo para cualquier elección de* $w \in V - V'$. *Se puede determinar la distancia entre dos vértices presentes en la misma componente conexa como mencionado en la definición anterior, pero entre vértices en distinta componente dicha métrica queda indefinida. En especial, un grafo con un único vértice es considerado como una sola componente conexa.*

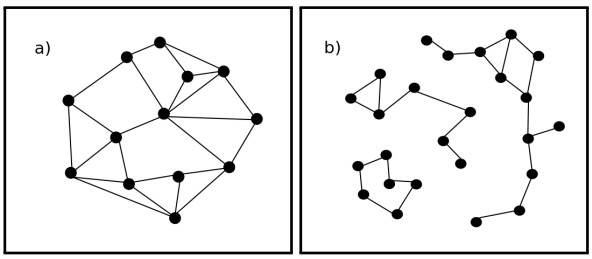

Figura 2.6: a) Un grafo no dirigido conexo y b) un grafo no dirigido disconexo.

Así, se dirá que todo grafo no dirigido conexo está compuesto por una única componente conexa. Por otro lado, para los grafos dirigidos la noción de conectividad es más amplia ya que un camino dirigido debe comprender la dirección de las aristas, aspecto que se trata en las siguientes definiciones.

Definición 2.2.7. *Sea G = (V, A) un grafo dirigido. Se dice que una sucesión $v_1, v_2, v_3, ..., v_n$ de n vértices de G, diferentes entre si, es un **camino dirigido** $P(v_1, v_n)$ de longitud $n-1$ desde v_1 hacia v_n, si y solo si $a_i = (v_i, v_{i+1}) \in A$ para $i = 1, 2, 3, ..., n-1$. Además, si $v_1 = v_n$ pero todos los demás vértices siguen siendo distintos entre sí, entonces tal sucesión se conoce como **ciclo dirigido**.*

Definición 2.2.8. *Se dice que un digrafo G es **débilmente conexo** si su grafo subyacente G_{sub} es conexo.*

Definición 2.2.9. *Un digrafo G = (V, A) se dice **unilateralmente conexo** si para toda pareja de vértices $u, v \in V$ existe, ya sea un camino dirigido $P(u, v)$, o bien, un camino $P(v, u)$.*

Definición 2.2.10. *Un digrafo G = (V, A) es **fuertemente conexo** siempre que para toda pareja de vértices $u, v \in V$ exista, tanto un camino dirigido desde u hacia v, como un camino dirigido desde v hacia u.*

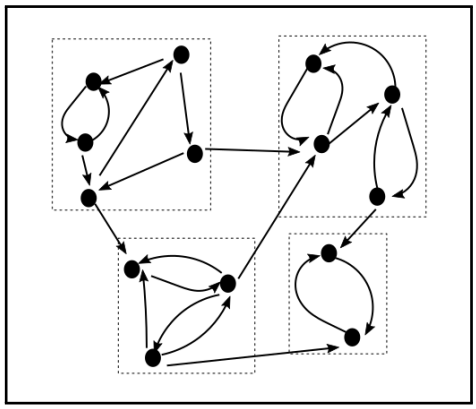

Figura 2.7: Un digrafo donde las componentes fuertemente conexas estan encerradas en cuadros punteados.

Definición 2.2.11. *Sea G = (V, A) un grafo dirigido, se define como **componente fuertemente conexa** (fig. 2.7) de G a todo subgrafo inducido G' = (V', A') tal que G' es fuertemente conexo, pero $G[V' \cup \{w\}]$ no es fuertemente conexo para cualquier $w \in V - V'$. Un grafo dirigido con un único vértice es considerado a su vez como una componente fuertemente conexa.*

Definición 2.2.12. *Un grafo dirigido que no contiene ciclos dirigidos es llamado **digrafo acíclico**, o **DAG**, por sus siglas en inglés (Directed Acyclic Graph, fig. 2.8).*

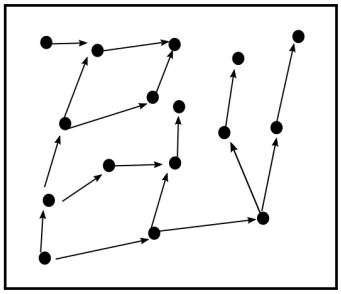

Figura 2.8: Ejemplo de un digrafo sin ciclos dirigidos, o DAG.

2.3 Propiedades de grafos

Para facilitar el estudio de las propiedades que se pueden determinar sobre un grafo [1], en esta sección las presentaremos categorizándolas de la siguiente forma: primero se incluyen algunas propiedades definidas para cada uno de los vértices y aristas de un grafo dado, en seguida se definirán aquellas propiedades y valores asociados directamente al grafo, siendo estos últimos comúnmente conocidos como invariantes por isomorfismos del grafo [2], y finalmente se describirán algunos agrupamientos de vértices que permiten definir subgrafos con características particulares.

Propiedades asociadas a vértices y aristas

Comenzaremos por describir los agrupamientos de vértices adyacentes a un vértice particular, y después las propiedades que de este conjunto se derivan.

Definición 2.3.1. *Sea G = (V, E) un grafo no dirigido, y sea v cualquier vértice de G, se define como **vecindario**, o conjunto de vecinos de v en G al conjunto*

$$N_v = \{u \in V | \{v, u\} \in E\}$$

Definición 2.3.2. *Sea G = (V, A) un grafo dirigido, y sea v cualquier vértice de G, se define como **vecindario de entrada**, o conjunto de vecinos de entrada de v en G al conjunto*

$$N_v^- = \{u \in V | (u, v) \in A\},$$

*y como **vecindario de salida**, o correspondientemente conjunto de vecinos de salida de v en G al conjunto*

$$N_v^+ = \{u \in V | (v, u) \in A\}$$

A partir de la definición de vecindario de un vértice v, se puede estudiar la cantidad de vértices relacionados con v. Si el grafo a tratar es no dirigido entonce se llamará **grado** de v a la cardinalidad de su vecindario, que en símbolos es $k_v = |N_v|$.

Por otro lado, si el grafo en contexto es dirigido entonces tendremos para cada vértice v en él un **grado de entrada** $k_v^- = |N_v^-|$ y un **grado de salida** $k_v^+ = |N_v^+|$. De esta forma, tanto para grafos dirigidos como para no dirigidos, a los vértices con mayor grado (de entrada o salida) se les llama **hubs** [1].

Por otro lado, recordando la definición de distancia entre dos vértices (def. 2.2.5), resulta posible evaluar que tan alejado se encuentra un vértice de todos los otros en un grafo no dirigido, concepto que se estudia con la siguiente definición.

Definición 2.3.3. *Sea G = (V, E) un grafo no dirigido, se define como la **excentricidad** de un vértice $v \in V$ a la máxima de las distancias $d(v, u)$ entre v y todo otro vértice $u \in V$, quedando denotada como*

$$ex(v) := max\{d(v, u) | u \in V\}$$

Dado un grafo no dirigido, es común referirse al grado de un vértice como su **centralidad de grado**. Se especifica que este valor es una centralidad *de grado* ya que que existen definidas otros tipos de centralidades, de las que se habla a continuación.

Definición 2.3.4. *Sea G = (V, E) un grafo no dirigido y conexo, con n vértices, y sea l_v la distancia promedio de un vértice $v \in V$ a todo otro vértice en G, dada por*

$$l_v = \frac{1}{n} \sum_{u \in V} d(v, u),$$

se define como ***centralidad de cercanía*** *de v al inverso de su distancia promedio a otros vértices*

$$C_v = \frac{1}{l_v}$$

Definición 2.3.5. *Dado un grafo no dirigido y conexo G = (V, E), y dado un vértice $v \in V$, denotamos por q^v_{uw} a la cantidad de caminos de menor longitud, tales que contienen a v, entre alguna pareja de vértices u y w distintos de v, y a la vez denotamos como p_{uw} a la cantidad total de caminos de menor longitud entre los vértices u y w. Entonces, se define como* ***centralidad de intermediación*** *del vértice v a la suma sobre todo par de vértices u y w*

$$\zeta_v = \sum_{u,w \in V \setminus \{v\}} \frac{q^v_{uw}}{p_{uw}}$$

Una arista también puede tener centralidad. La centralidad de intermediación para cada arista, descrita a continuación, depende también de la cantidad de caminos más cortos a los que pertenece [16].

Definición 2.3.6. *Sea G = (V, E) un grafo no dirigido. Dada una arista $e \in E$, denotamos como η^e_{uv} a la cantidad de caminos de menor longitud entre los vértices u y v, tales que e relaciona a alguna pareja de vértices en esos caminos, y también denotamos por κ_{uv} a la cantidad total de caminos de menor longitud entre los vértices u y v. De esta forma, se define como* ***centralidad de intermediación*** *de e, o simplemente centralidad de e, a la suma sobre todo par de vértices u y v*

$$c_e = \sum_{u,v \in V} \frac{\eta^e_{uv}}{\kappa_{uv}}$$

Por otro lado, dado un vértice v, es posible evaluar en que medida se encuentran agrupados los vecinos de v como se muestra a continuación [1].

Definición 2.3.7. *Sea G = (V, E) un grafo (no dirigido o dirigido), y sea v un vértice en V, denotamos como f_v a la cantidad de aristas (o flechas) entre los vecinos de v, y denotamos por g_v a la cantidad total de parejas (no ordenadas u ordenadas, según sea el caso), sin repetición, de vecinos de v. Se define así, como* ***coeficiente de agrupamiento local*** *de v, a la proporción*

$$\alpha_v = \frac{f_v}{g_v}$$

Para concluir con las propiedades asociadas a los vértices y aristas de un grafo, daremos las definiciones correspondientes a aquellos vértices y aristas cuya supresión provoca que el grafo no dirigido al que pertenecen se rompa en varias componentes. Esto queda definido como sigue:

Definición 2.3.8. *Sea G = (V, E) un grafo no dirigido, con n componentes conexas, y sea v un vértice en V. Si el grafo inducido $G[V - \{v\}]$ tiene $m > n$ componentes conexas, entonces v es llamado **vértice de corte** de G.*

Definición 2.3.9. *Sea G = (V, E) un grafo no dirigido, con n componentes conexas, y sea e una arista en E. Si el grafo $G' = (V, E - \{e\})$ tiene $m > n$ componentes conexas, entonces e es llamada **arista puente** de G.*

Propiedades asociadas a un grafo

Las primeras dos propiedades a tratar en este apartado nos hablan sobre la cantidad de elementos que componen a un grafo.

Definición 2.3.10. *Dado un grafo G = (V, E), sea este dirigido o no dirigido, se define como **orden** de G a la cantidad de vértices que este tiene, y como su **tamaño** a la cardinalidad de su conjunto de aristas (o flechas).*

Considerando que trataremos únicamente con grafos con un número finito de vértices y aristas, se aprecia que la máxima cantidad de parejas de vértices distintos, ordenadas o no ordenadas, que puede tener un grafo, está limitada a su vez por el orden que este tenga.

Observación 2.3.1. *Sea G = (V, E) un grafo no dirigido con orden n, el máximo número de aristas que G puede tener está dado por*

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$$

Lo anterior se sigue inmediatamente de la definición de grafo no dirigido, ya que el conjunto de las aristas comprende a todas las parejas no ordenadas de vértices distintos, o combinaciones sin repetición de los vértices tomados de dos en dos.

Además, un grafo dirigido G de orden n tendrá máximo $n(n-1)$ flechas, ya que se pueden asociar exactamente dos direcciones a cada pareja no ordenada de los vértices de G, y recordando que G es simple, entonces no deben existir flechas apuntando desde un vértice hacia si mismo ni adyacencias múltiples. Con estas propiedades se formula la siguiente definición.

Definición 2.3.11. *Dado un grafo no dirigido G = (V, E), de orden n y con tamaño m, se define como **densidad** de G a la proporción de las aristas que tiene, entre el máximo de aristas que pueden formarse con sus n vértices, esto es*

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}$$

Mientras que un grafo dirigido $G = (V, A)$, también de orden n y tamaño m tendrá una densidad dada por

$$\rho = \frac{m}{n(n-1)}$$

En general, todo grafo que cumple tener a todas sus posibles aristas o flechas, y consecuentemente tener densidad igual a 1, es llamado **grafo completo**. A su vez, la menor densidad que puede tener un grafo será 0, propia de un grafo sin aristas.

Es importante notar que aunque la densidad de un grafo sea cercana a 0, los vértices en él pueden aún encontrarse agrupados fuertemente. Para medir esto, recuperando la definición de coeficiente de agrupamiento local de un vértice (def. 2.3.7), podemos definir para un grafo G, dirigido o no dirigido, con orden n, un **coeficiente de agrupamiento promedio**, dado por

$$C = \frac{1}{n} \sum_{v \in V(G)} \alpha_v$$

Además, dada una partición del conjunto de vértices de un grafo, es posible evaluar por medio de la medida llamada modularidad, descrita a continuación, la similitud de tal partición respecto a la formación de los grupos particulares de vértices, conocidos como comunidades. Dicha medida depende de la noción de que debe existir un mayor número de aristas conectando vértices dentro de la misma comunidad, que conectando vértices en diferentes comunidades.

Definición 2.3.12. *Dado un grafo no dirigido $G = (V, E)$ de tamaño m y una partición $\tilde{P}$ de los vértices de G, se define como **modularidad de G respecto a $\tilde{P}$**, a la suma sobre toda pareja ordenada de vértices*

$$Q = \frac{1}{2m} \sum_{(u,v) \in V \times V} [A_{uv} - \frac{k_u k_v}{2m}] \delta(C_u, C_v)$$

donde $V \times V$ es el producto cartesiano de V consigo mismo, A_{uv} es 1 si $\{u, v\} \in E$ y 0 de otro modo, k_u y k_v son los grados de dos vértices u y v respectivamente, C_u y C_v representan las celdas de la partición $\tilde{P}$ a las que pertenecen u y v, y $\delta(C_u, C_v)$ es una función que vale 1 si $C_u = C_v$ y 0 cuando $C_u \neq C_v$.

La modularidad es una medida que toma valores en el intervalo $[-1, 1]$, de modo que una modularidad cercana a 1 representa una partición en grupos muy parecidos a una comunidad, mientras que valores cercanos a -1 indican que tal partición propicia la existencia de un mayor número de aristas entre vértices pertenecientes a distintas comunidades. Se debe mencionar que la medida de modularidad forma parte del problema teórico de detección de comunidades en grafos, que a la fecha no se encuentra rigurosamente definido [6], y sobre el cual se hablará más a detalle en el capítulo 4 de esta tesis.

Por otro lado, buscaremos evaluar la cantidad promedio de aristas que existen unidas a cada vértice. Para esto, se debe notar que la suma de los grados de todos los vértices es igual a dos veces la cantidad de aristas en el grafo, ya que cada arista se estaría contando exactamente dos veces en dicha suma, aspecto condensado en el llamado **Teorema del doble conteo** [2]. Así, dado un grafo no dirigido G con orden n y tamaño m, se define al **grado promedio** de los vértices en G como

$$< k >= \frac{1}{n} \sum_{v \in V(G)} k_v = \frac{2m}{n}$$

Sin embargo, en el caso de los grafos dirigidos se tiene que la suma de los grados de entrada es igual a la suma de los grados de salida, y ambas sumas son iguales al tamaño del grafo (ya que toda flecha que sale de un vértice es también una flecha que entra a otro), con lo que el **grado promedio** de un digrafo G se puede definir como el tamaño m del digrafo dividido entre su orden n

$$< k >= \frac{m}{n}$$

Para complementar estas propiedades, se tienen también las siguientes medidas sobre un grafo no dirigido, relacionadas con la excentricidad (def. 2.3.3) de sus vértices.

Definición 2.3.13. *Dado un grafo simple no dirigido G, se define como* ***diámetro*** *de G a la máxima de las excentricidades entre sus vértices, y como* ***radio*** *de G a la mínima de estas excentricidades.*

En cuanto a las propiedades derivadas de la definición de grado de un vértice, dado un grafo no dirigido G, se presta especial atención al máximo grado y al mínimo grado, tomados de la colección de los grados de todos los vértices de G. Denotando estos por k_{max} y k_{min} respectivamente, se tiene que si $k_{max} = k_{min} = k$, entonces G es llamado k**-regular**.

De manera semejante se pueden definir grados máximos y mínimos para grafos dirigidos, pero esta vez se tendrán cuatro posibles definiciones, específicamente, grados máximos de entrada y salida, así como grados mínimos de entrada y salida. Otra propiedad importante relacionada con los grados de los vértices en un grafo es la siguiente.

Definición 2.3.14. *Sea $G = (V, E)$ un grafo no dirigido de orden n. Se define como* ***sucesión de grado*** *de G, a la secuencia $k_1, k_2, k_3, ..., k_n$, conformada por los n grados tomados a partir de cada uno de sus vértices, de tal forma que $k_i \leq k_{i+1}$ para $i = 1, 2, 3, ..., n - 1$.*

Definición 2.3.15. *Sea G un grafo no dirigido con orden n, y sea n_k la cantidad de vértices de G con grado k. Se define como* ***distribución de grado*** *de G, respecto a todo entero no negativo k, a la colección de las proporciones*

$$P(k) := \frac{n_k}{n}$$

Tanto la secuencia de grados de un grafo no dirigido, como su distribución de grado, tienen su análogo para grafos dirigidos. Ya que ambas definiciones dependen de la definición de grado de un vértice, debemos recordar que en grafos dirigidos cada vértice tiene asociado dos tipos de grados, uno respecto a su vecindario de salida y otro respecto a su vecindario de entrada. Así, para un grafo dirigido se tendrán dos secuencias de grado, una secuencia de grados de entrada y una secuencia de grados de salida. Esto también sucederá para la distribución de grado, por lo que dado un grafo dirigido G, se podrá hablar de la fracción de vértices que tiene cierto grado, de entrada o de salida, respecto al orden de G.

Agrupamientos de vértices

Definición 2.3.16. *Sea G = (V, E) un grafo no dirigido y conexo, se define como* ***centro*** *de G al conjunto $Z(G)$ conformado por todos los vértices de G cuya excentricidad sea igual al radio de G.*

Definición 2.3.17. *Sea G = (V, E) un grafo, dirigido o no dirigido, se define como* ***clique*** *(o* ***clan****) a cualquier conjunto $V' \subseteq V$ que induce un subgrafo $G' = G[V']$ tal que G' es un grafo completo. Si G' es completo pero $G[V' \cup \{w\}]$ no es completo para cualquier $w \in V - V'$, entonces V' es llamado* ***clique maximal*** *(fig. 2.9).*

Definición 2.3.18. *Sea G = (V, E) un grafo, dirigido o no dirigido, se define como* ***conjunto independiente*** *de vértices a cualquier conjunto $V' \subseteq V$ que induce un subgrafo $G' = G[V'] = (V', E')$ tal que $E' = \varnothing$.*

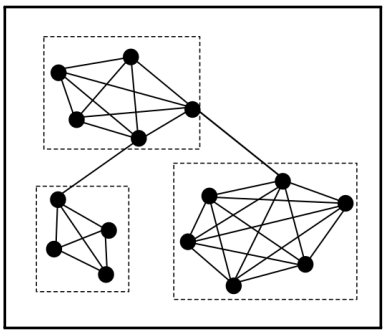

Figura 2.9: Un grafo simple donde se encierran en cuadros punteados algunos cliques maximales.

Definición 2.3.19. *Dado un grafo no dirigido G = (V, E), y denotando por V^c al conjunto de todas las posibles colecciones de vértices de corte en G, se define como* ***conjunto de vértices de corte mínimo*** *a todo conjunto con cardinalidad mínima de entre los elementos de V^c.*

Definición 2.3.20. *Dado un grafo no dirigido G = (V, E), y denotando por E^p al conjunto de todas las posibles colecciones de puentes en G, se define como* ***conjunto de aristas de corte mínimo*** *a todo conjunto con cardinalidad mínima de entre los elementos de E^p.*

Capítulo 3

Análisis de propiedades de redes ecológicas

Un ecosistema se encuentra constituido tanto por una cierta colección de organismos y factores abióticos, como por las interacciones que entre estos se desarrollan [5]. El presente trabajo comprende el análisis de las redes [1] de interacciones ecológicas existentes en un tipo particular de ecosistema, conocido como tapete microbiano [10]. Estos son estructuras laminares (fig. 3.1) compuestas por una combinación de colonias microbianas y sus respectivos sustratos.

3.1 Problema tecnológico: análisis de propiedades de redes ecológicas en tapetes microbianos de la reserva ecológica de Cuatro Ciénegas, Coahuila

En este capítulo se muestra el análisis y los resultados obtenidos del estudio de tapetes microbianos, originados en un estanque cercano a la Laguna Churince de la reserva ecológica de la cuenca de Cuatro Ciénegas (CCC), ubicada en el estado mexicano de Coahuila (fig. 3.2). Este trabajo consistió en el estudio de las redes de interacciones ecológicas presentes a diferentes niveles taxonómicos de los organismos habitantes de estos tapetes.

Dicha reserva, y en particular la Laguna Churince, se han visto sujetas a desecaciones debidas a la actividad humana, esencialmente por el uso del agua en el desarrollo agropecuario de la región [13]. Dado que CCC se caracteriza principalmente por su alto contenido de especies endémicas, es de importancia evaluar el impacto que ha tenido el cambio en los niveles de agua sobre las colonias celulares que conforman los tapetes.

Figura 3.1: Tapete microbiano.
(Imagen tomada de [10])

Motivados por evaluar la respuesta de estos organismos ante las perturbaciones en los niveles de agua en CCC, se desarrolló un análisis de las propiedades de redes ecológicas proporcionadas por la Dra. Valerie de Anda del Instituto de Ecología de la UNAM, quien llevo a cabo su inferencia haciendo uso del software MetaMIS [12], sobre datos de metagenómica de los tapetes. Ella y su equipo extrajeron dichos datos a partir de muestras que recolectaron a lo largo de dos años (Otoño del 2012 a Primavera del 2014), tomándolas en tres sitios del estanque llamado *la Lagunita*, durante cuatro tiempos distintos, caracterizados por diferentes niveles de agua.

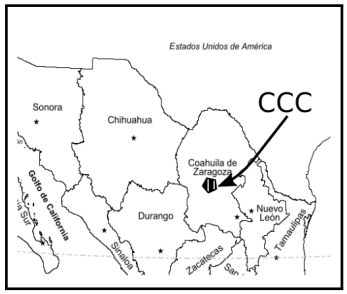

Figura 3.2: Localización de CCC.
(Imagen tomada y adaptada de http://cuentame.inegi.org.mx/mapas).

En colaboración con la Dra. de Anda se han evaluado diversas propiedades de estas redes ecológicas, concluyendo con la publicación de los resultados [11]. En esta tesis se presenta un análisis que complementa a aquel desarrollado junto con la Dra. de Anda. Con ambos estudios se buscó analizar principalmente cuatro aspectos de los tapetes: la capacidad de los organismos para desarrollar agrupamientos, determinar las diferencias entre las proporciones de interacciones ecológicas de colaboración y exclusión [5] respecto a las condiciones en los niveles de agua, detectar aquellos microorganismos con mayor número de interacciones, y finalmente determinar patrones de interacción en las redes, conocidos como motivos de red [33, 34].

Este capítulo se desarrolla estableciendo primero un panorama general de la disciplina llamada ciencia de redes [1]. Seguido a esto se proporciona un marco teórico respectivo a la clasificación taxonómica, interacciones ecológicas, tapetes microbianos, y los conceptos de análisis de redes básicos para este capítulo. Posteriormente se aborda el tema de los motivos de red, seguidos por la metodología empleada para este análisis. Finalmente se presentan y discuten los resultados de este trabajo.

3.2 Introducción a la ciencia de redes

Una red es un modelo matemático, y la disciplina que se encarga de estudiar a estos modelos es llamada ciencia de redes. El formalismo matemático que sustenta a estos modelos se encuentra principalmente en la teoría de grafos [2, 3, 4], de modo que cada red puede ser considerada como un grafo cuyas propiedades tienen cierta interpretación, dependiente del problema a estudiar. Con las redes se busca fundamentalmente esquematizar dos características de un sistema dado: los elementos que lo conforman y las interacciones que existen entre estos elementos.

Para hablar de los componentes en una red se hace uso de la misma nomenclatura de teoría de grafos, siendo que los vértices de un grafo se corresponden con los elementos de la red, y las aristas (o flechas) con las interacciones entre esos elementos. Es habitual que a los vértices en una red se les llame nodos, sin embargo, en este trabajo nos restringiremos a llamarlos vértices para resumir el uso del lenguaje.

Aspectos generales del análisis de redes

Las redes tienen aplicaciones tecnológicas en diversos tipos de sistemas [1, 2, 3]: sociales, biológicos, de información, físicos, químicos y en ingeniería. En general, pueden ser utilizadas para analizar problemas donde se presenten componentes o agentes que se encuentren conectados entre sí, o interactuando bajo ciertas reglas.

A su vez, un mismo sistema puede tener asociadas más de una red. Con esto se busca poder representar todos los estados, configuraciones, o facetas que este puede tener estando sujeto a diferentes tipos de restricciones. De esta forma, al igual que con otros modelos matemáticos, las propiedades de redes son características que pueden explicar fenómenos o procesos de interés sobre el sistema que modelan.

El análisis de redes no se limita únicamente a aplicar los conceptos de la teoría de grafos para desarrollar sus modelos. En este también se recurre a otras especialidades, como la estadística, teoría de probabilidad, ciencias de la computación, dinámica de sistemas y teoría de control, para enriquecer la información en una red y complementar el análisis que sobre estas se hace.

En este trabajo nos enfocaremos en el estudio de los cambios por los que pasan algunas redes sometidas a distintas condiciones. A lo largo de esta tesis, recurriremos a la aplicación de conceptos de estadística, teoría de la probabilidad, y ciencias de la computación para sentar las bases de los diversos métodos de análisis a aplicar sobre redes que modelan sistemas biológicos.

Análisis de redes en biología

La mayoría de los sistemas biológicos presentan cierta complejidad en su estructura y dinámica. Por complejidad en este caso se debe entender la existencia de las llamadas propiedades emergentes de un sistema [22], que no se explican por las características de sus elementos, sino por las interacciones que entre ellos se dan. Es por esto que el análisis de las redes en biología ha permitido comprender fenómenos surgidos de dichas relaciones complejas.

Por esto, se puede encontrar una amplia colección de aplicaciones del análisis de redes en biología. Ejemplos de los problemas que con estas se busca estudiar, abarcan desde la descripción de la estructura anatómica de ciertos organismos y los canales de comunicación entre los componentes de cada órgano y tejido, hasta el estudio de procesos biológicos, como rutas metabólicas, mecanismos bioquímicos y propagación de enfermedades.

Además, es común encontrar ejemplos tanto de redes modelando interacciones dirigidas como no dirigidas. En el caso de interacciones dirigidas se tienen ejemplos como: redes tróficas (fig. 3.3), donde se desarrolla una estructura jerárquica entre organismos sujetos a interacciones de depredación [34], o redes de regulación genética, que pueden a su vez ser usados para, o bien mostrar las relaciones entre genes y sus factores de transcripción [26, 42], o visualizar cascadas de señalización, útiles para describir los mecanismos de respuesta de una célula ante cambios en su entorno [43].

En lo que respecta a las redes no dirigidas, se pueden estudiar problemas relacionados con: redes que muestran interacciones de amistad o reproducción entre organismos en una población dada [14], conexiones anatómicas en circuitos neuronales [19], y redes que describen la estructura de compuestos como proteínas y ácidos nucléicos, utilizadas para el estudio de su composición y plegamiento.

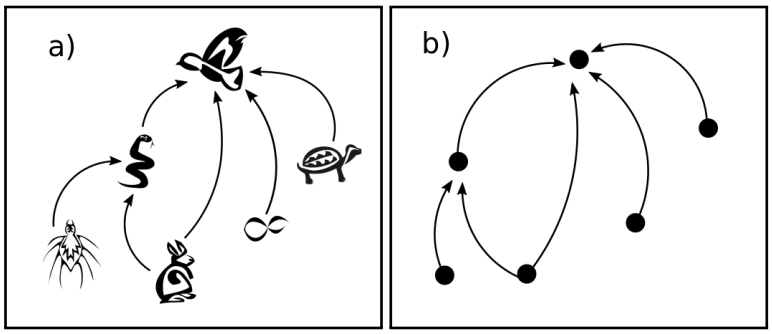

Figura 3.3: a) Caricatura de una red trófica y b) el grafo dirigido que la constituye.

Propiedades de redes

En el estudio de las propiedades de redes biológicas, se pueden encontrar algunas características representativas como: la cantidad de elementos e interacciones que conforman a la red, elementos con un mayor número de interacciones, e incluso elementos cuya remoción de la red provoca que esta quede desconectada. Todos estos son ejemplos de características de grafos que tienen a su vez alguna interpretación sobre una red.

En primera instancia, todos los atributos asociados a grafos son candidatos para ser analizados como propiedades de redes. Para facilitar el estudio de estas propiedades, podemos agruparlas en tres categorías: características correspondientes a cada uno de los vértices o aristas de la red, valores que caracterizan a la red en sí misma, y colecciones de vértices o aristas determinados por poseer propiedades en común. Como ejemplo de las primeras, se pueden inferir valores de centralidad para cada vértice, de modo que con estos se indique la relevancia de cada elemento en su sistema. Por otro lado, para una red se puede determinar un valor de densidad, correspondiente a la proporción de interacciones existentes entre sus elementos, contra la cantidad máxima de interacciones que estos pueden llegar a tener. Y en lo que concierne a los agrupamientos de vértices, se pueden estudiar por ejemplo, aquellas colecciones de elementos tales que ninguno de ellos se encuentre interactuando con otro en su colección, es decir, conjuntos independientes de vértices.

Es importante recalcar que las propiedades que aquí se mencionan son valores derivados de las propiedades matemáticas de la red con la que se cuente. Es decir que para realizar el cálculo de estas características, no es necesario contar *a priori* con ningún dato empírico ni información particular sobre el sistema a analizar, más que la red misma.

Sin embargo, es posible enriquecer el análisis de las propiedades de redes al contar con datos experimentales, o que nos puedan hablar sobre aspectos específicos del sistema, como: coloreados sobre los vértices (fig. 3.4) que permitan estudiar agrupamientos en redes, coeficientes de correlación de Pearson como indicadores de la fuerza de cada interacción entre los elementos de la red, y los gradientes químicos o físicos asociados a cada flecha en una red de transporte o señalización.

Lo anterior son ejemplos de cómo toda información empírica es relevante para favorecer y complementar el análisis de redes. A continuación se describirá el uso de esta información experimental para poder, primeramente, obtener una red que modele un sistema de nuestro interés.

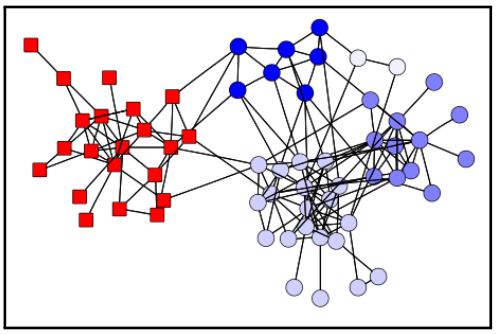

Figura 3.4: Una red de relaciones sociales entre delfines, coloreados por comunidades detectadas en [14].

Construcción e inferencia de redes

En ocasiones el sistema a investigar se encuentra definido en términos de una red. Muestra de esto son las redes a nivel de conexiones físicas en Internet [1]. Más aún, también es posible encontrar este tipo de redes en la biología, siendo un ejemplo concreto de esto las redes neuronales (fig. 3.5) en el sistema nervioso central de algún organismo [8, 20]. Sin embargo, no todo sistema se puede encontrar ya estructurado de esta manera, por lo que comúnmente se debe llevar a cabo una investigación respecto a las interacciones que hay en él.

Es posible construir redes únicamente por medio de la recopilación de información bibliográfica de un sistema. De este modo, se puede enumerar cada uno de los elementos en el sistema y las relaciones entre ellos, argumentando la existencia de cada componente dentro la red. Un ejemplo de esto consiste en esquematizar las cascadas de señalización entre genes y sus factores de transcripción para modelar los fenotipos de una célula al pasar por cambios en su entorno [43]. Debido al trabajo que esto conlleva, este método resulta práctico para redes pequeñas ($\sim$ 20 vértices , $\sim$ 30 aristas).

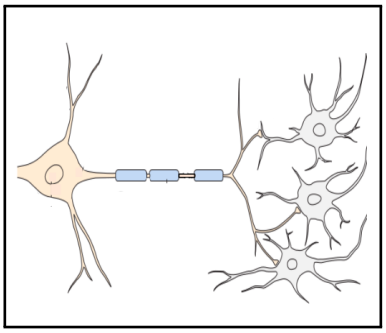

Figura 3.5: Células en una red neuronal biológica.
(Imagen tomada y adaptada de [8])

Sin embargo existen sistemas, particularmente en biología, que cuentan con una gran cantidad de componentes [1], tanto de elementos como de interacciones entre ellos. Para construir redes en este caso, resulta necesario recurrir a métodos estadísticos o probabilísticos de construcción de redes. A este tipo de métodos en particular se les conoce como métodos de inferencia de redes [44].

Un método comúnmente utilizado para inferir una red a partir de información empírica, requiere que dada una colección de muestras de componentes biológicos, se calcule para cada pareja de ellos alguna medida estadística que se pueda interpretar como grado o fuerza de interacción. De esta forma, se puede llegar a afirmar la existencia de una arista en la red si tal medida adquiere valores mayores a un cierto umbral [44]. Es posible pensar en el valor absoluto de un coeficiente de correlación lineal como una medida de interacción [45]. No obstante, dicho valor describe una correlación lineal, y dado que no toda correlación implica una verdadera interacción, o que no toda interacción puede ser caracterizada como lineal, se advierte que las redes inferidas con este coeficiente pueden llegar a describir interacciones de una forma inconsistente con el sistema que se esté analizando. Además, el hecho de establecer un umbral sobre una medida de interacción conlleva dos problemas principales [44]. Uno de ellos es que en general no se sabrá cuantas aristas serán falsos positivos para cada posible umbral. Aunado a esto, es posible que el valor de este umbral varíe dependiendo del sistema a analizar, lo que puede llegar a restar objetividad a la inferencia. Debido a esto, es preferible realizar pruebas de hipótesis [45] sobre la distribución de la medida de interacción en una serie de datos [44, 46], lo que atribuye a cada red un valor de incertidumbre que puede ser reducido al estudiar más muestras.

Por otro lado, es posible recurrir a métodos o sistemas computacionales que contemplen interacciones más complejas. Como ejemplo de esto se tiene el sistema MetaMIS (Metagenomic Microbial Interaction Simulator) [12], utilizado para inferir las redes ecológicas de los tapetes microbianos analizados en este capítulo. Este software realiza su inferencia comparando datos de metagenómica contra modelos Lotka-Volterra (LV), usados en el estudio de la dinámica depredador-presa. Sin embargo, al igual que con otros modelos matemáticos, las redes también deben estar sujetas a validación y corrección siempre que sea posible. De esta manera, se debe contar con información empírica que permita validar las interacciones inferidas. Esto resulta particularmente complicado al tratar con interacciones microbianas, ya que a diferencia de sistemas macroscópicos donde las interacciones se pueden observar directamente, no se dispone comúnmente de información experimental u observacional sobre toda interacción microscópica [46]. Lo anterior dificulta afirmar que incluso sistemas como MetaMIS reproduzcan fielmente las interacciones en todo tipo de entorno microbiano [47].

Además, se debe advertir que recientemente se demostró cómo es que en general, varias redes pueden tener asociada una misma dinámica [48], o evolución temporal como la descrita por los modelos LV. Esto indica que a partir de una misma colección de datos empíricos se pueden llegar a inferir múltiples redes distintas entre sí, todas igualmente válidas, y de manera independiente al método de inferencia utilizado. Aunado a esto, se ha mostrado como el uso de la llamada *abundancia relativa* (cantidad de organismos de una misma especie, o taxon, normalizada respecto a la cantidad total de organismos en un ecosistema), puede tener un impacto perjudicial en la inferencia de redes basada en los modelos LV, ya que por medio de esta no se puede conocer a la población total del sistema, reduciendo así la capacidad de los LV para reproducir las poblaciones originales de los organismos estudiados [47].

Ignorando dichas limitaciones, MetaMIS proporciona una única red, llamada *consenso*, para representar a un sistema microbiano. Más aún, sus autores recomiendan usar datos de abundancia relativa para realizar la inferencia [12], sin hacer mención de estas problemáticas. Sin embargo, el programa admite el uso de *abundancia absoluta* (cantidad sin normalizar de organismos de un mismo taxon), por lo que previendo los inconvenientes relacionados con el uso de los datos normalizados, la Dra. de Anda y su equipo utilizaron abundancia absoluta para inferir las redes ecológicas de los tapetes microbianos [11]. No obstante, se recomienda tener precaución al utilizar MetaMIS, y se reconoce que es de importancia estudiar en un futuro el impacto de tales condiciones sobre la inferencia de redes con él.

Todo esto permite apreciar que la inferencia de redes conlleva un problema de investigación en sí mismo, por lo que se recomienda siempre desarrollar un estudio respecto al estado del arte en la inferencia de redes sobre el sistema biológico que se quiera analizar. Finalmente, es importante mencionar que las redes sobre las que se desarrolla el análisis de cada uno de los capítulos siguientes provienen de bases de datos públicas [21, 37], lo que nos permite saber que ya han sido curadas o estudiadas previamente.

3.3 Marco teórico

Para estudiar las interacciones entre los organismos presentes en los tapetes microbianos de CCC, se infirieron redes a diferentes niveles taxonómicos. La Dra. de Anda y su equipo llevaron a cabo un análisis metagenómico de varias muestras de estos tapetes, proporcionando después estos datos al programa MetaMIS. Las redes inferidas por medio de dicho software son grafos dirigidos donde cada flecha se encuentra etiquetada con alguno de dos signos, positivo (+) o negativo (-), que indican si la interacción ecológica es de cooperación o exclusión respectivamente. Sobre estas se desarrolló el análisis de múltiples propiedades de redes, entre las que destacan algunas que hablan sobre la distribución de las interacciones entre los organismos, como: densidad de interacciones de exclusión, densidad de interacciones de colaboración, densidad total de la red, grado promedio y coeficiente de agrupamiento promedio. Finalmente, para enriquecer el análisis, se estudió la presencia de los subgrafos estadísticamente significativos llamados motivos de red, buscando determinar patrones significativos de interacción entre organismos. En esta sección presentamos el marco teórico relacionado a todos estos conceptos.

Clasificación taxonómica

La organización taxonómica, originalmente propuesta por el científico sueco Carlos Linneo (1707 - 1778), es un sistema de clasificación jerárquico usado para agrupar a los seres vivos. Consiste principalmente de ocho niveles [5], donde cada nivel, o taxon, describe una colección de varias instancias de su nivel inmediato inferior (fig. 3.6).

Los niveles que este sistema contempla actualmente, ordenados comenzando por el más bajo, son: especie, género, familia, orden, clase, filum, reino y dominio. En ocasiones se incluyen los grupos de subclase, subfilo y subreino, aunque el modelo más común contempla solo los 8 niveles mencionados. De esta forma, una colección de especies conforma un género y una colección de géneros conforma una familia, y así sucesivamente.

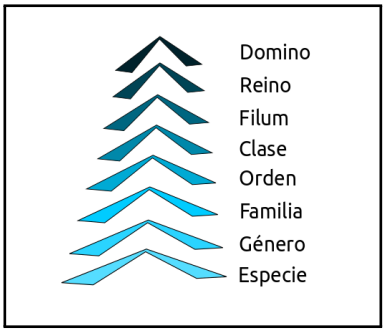

Figura 3.6: Niveles taxonómicos.

Dada cierta especie, entonces se puede mencionar a que género, familia, orden, y otros grupos pertenece. Por ejemplo, el hongo de la levadura de cerveza, o especie *Saccharomyces cerevisiae*, pertenece también a la familia *Saccharomycetaceae*, orden *Saccharomycetales* y clase *Saccharomycetes*.

Las definiciones relacionadas con la taxonomía son establecidas y reguladas por organizaciones como el Sistema Integrado de Información Taxonómica, ITIS por sus siglas en inglés (Integrated Taxonomic Information System) [49]. Este sistema de clasificación permite recuperar información relevante de cada organismo, y dado que es estandarizado por la ITIS, facilita la construcción y mantenimiento de bases de datos de taxonomía, de las cuales es ejemplo One-Codex [50], usada para estudiar la clasificación de los organismos en los tapetes microbianos de CCC.

Tapetes microbianos

Los tapetes microbianos son estructuras laminares formadas por microorganismos que se adhieren a superficies, en lagunas y mares, donde pueden establecer colonias para poder subsistir en ambientes sujetos a distintos factores abióticos [10].

Es importante mencionar que estos tapetes no son completamente planos. Debido a que más microorganismos se pueden adherir a un tapete ya formado, comúnmente estos se encuentran constituidos por capas de colonias acomodadas una sobre otra. Esto es posible ya que dentro de un mismo tapete se dan gradientes de diversos factores químicos, produciendo sustancias que fungen como substrato para las nuevas capas (fig. 3.7). Sin embargo estos gradientes dependen a su vez del entorno de los tapetes, por lo que variaciones en la salinidad, temperatura, o profundidad del cuerpo de agua en el que se encuentran, representa para ellos una perturbación en los nutrientes de los que disponen.

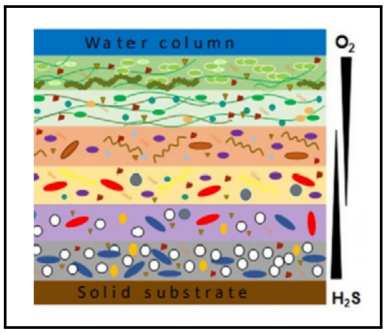

Figura 3.7: Capas en los tapetes microbianos.
(Imagen tomada y adaptada de [10])

La importancia del estudio y conservación de los tapetes microbianos, radica en ciertas características que estos presentan y sus potenciales aplicaciones tecnológicas. Por ejemplo, se ha encontrado que los tapetes más antiguos han existido desde hace un poco más de 3 mil millones de años (3.4 a 3.7 x 10^9 años). Además, algunos tapetes se han desarrollado en ambientes con condiciones extremas de salinidad o temperatura, por lo que podrían llegar a dar indicios del surgimiento de la vida en la tierra.

Por otro lado, dado que los tapetes microbianos llevan a cabo un intercambio de proteínas con su entorno, estos podrían ser utilizados en la producción industrial de fármacos, catalizadores, y otras enzimas, a cambio de ser provistos con algún substrato. Sin embargo, no solo ofrecen ventajas como productores de químicos, sino también como degradadores, con utilidad por ejemplo en sitios marítimos contaminados por petróleo.

Finalmente, también se debe considerar que la principal fuente de alimento de la mayoría de estos tapetes es la fotosíntesis. De esta forma, los tapetes representan maquinarias bioquímicas que actualmente limpian el aire, e incluso se considera que pueden ser aprovechados para degradar algunos contaminantes causantes del calentamiento global.

Análisis metagenómico

El estudio de un metagenoma comprende la recolección y clasificación de material genético a partir de muestras tomadas directamente del medio ambiente [51]. Este conlleva un análisis multidisciplinario, al relacionar áreas como la microbiología, ecología, secuenciación genética, bioinformática y genómica. Con él se busca conocer toda la diversidad microbiana en un ambiente o ecosistema específico.

Originalmente, el análisis de metagenómica se basó en el cultivo de los organismos en las muestras del entorno a analizar, y la medición de la abundancia o cantidad de organismos en estos cultivos. Sin embargo, nuevos avances en las técnicas de procesamiento de las muestras y secuenciación del ADN en ellas, han permitido reconocer que no todos los organismos que realmente existen en esas muestras sean aptos para cultivo, implicando que es posible que un análisis por cultivo pueda resultar incompleto en comparación con métodos actuales.

A la fecha, uno de los métodos más usados para estudiar este tipo de muestras es la *Secuenciación Shotgun, o Escopeta*, donde el ADN de cada organismo u OTU (Operational Taxonomic Unit), es separado en segmentos para posteriormente realizar su secuenciación por medio de programas y equipo de *Secuenciación de Nueva Generación*, o NGS (Next Generation Sequencing) [51].

Por último, para conocer la taxonomía relacionada a las secuencias derivadas de las muestras, se recurre a otros programas computacionales, como One-Codex [50], que poseen bases de datos para comparar estas secuencias y reconocer la taxonomía de lo organismos a los que pertenecen.

Generalidades sobre el software MetaMIS

MetaMIS es un software de inferencia de redes basado en el análisis de datos de metagenómica de un ecosistema [12]. Está programado usando la paquetería del software MATLAB R2015b (The MathWorks, Inc., Natick, Massachusetts, United States), y se encuentra disponible para sistemas operativos Mac y Windows.

Al proporcionarle datos de abundancia (absoluta o relativa) de N organismos, este programa desarrolla un preprocesamiento, seleccionando una cantidad N_{HA} de OTU's con alta abundancia, o abundancia promedio mayor a 1%, y otra cantidad LA de OTU's con baja abundancia, o abundancia promedio menor a 0.1%. Luego, MetaMIS construye un modelo Lotka-Volterra discreto para cada par de OTU's en los datos. Basándose en los modelos individuales, el sistema produce un modelo Lotka-Volterra generalizado (GLV) para evaluar el comportamiento acoplado de toda pareja.

Usando la *disimilaridad de Bray-Curtis*, MetaMIS evalúa que el modelo GLV reproduzca la abundancia de cada OTU en los datos originales. Cuando los modelos aproximan correctamente a los datos empíricos, el sistema construye una red dirigida simple llamada *red de interacción*, cuyas flechas presentan dos signos, $(+)$ para interacciones de colaboración y $(-)$ de exclusión . De manera predeterminada, este proceso se repite $IN = N - N_{HA} + 1$ veces, con lo que se obtiene una cantidad IN de redes de interacción. No obstante, dependiendo de la cantidad LA de organismos de baja abundancia, los parámetros internos de MetaMIS se pueden ver modificados para producir una cantidad IN de redes de interacción menor a la predeterminada.

Tomando en cuenta la dirección de cada flecha y su respectivo signo entre todas las redes de interacción, MetaMIS considera cuatro posibles flechas por cada par de OTU's. Para esto, el sistema construye dos proporciones $n^+_{ab}/(n^+_{ab} + n^-_{ab})$ y $n^-_{ab}/(n^+_{ab} + n^-_{ab})$ por cada par ordenado (a, b) de organismos, donde n^+_{ab} representa la cantidad de redes de interacción en las que existe una flecha de a hacia b etiquetada con signo $(+)$, y lo mismo para n^-_{ab}. Seguido a esto, se lleva a cabo una prueba de hipótesis sobre la distribución de dichas proporciones respecto a todas las redes de interacción, con lo que se determinan como válidas las flechas que tienen mayor presencia en el conjunto de las redes de interacción.

Se debe señalar que la bibliografía no menciona explícitamente la solución al caso en el que se puedan determinar como válidas las dos flechas, positiva y negativa, sobre la misma pareja ordenada (a, b), ni tampoco si este caso es teóricamente imposible. Sin embargo, por medio de la prueba de hipótesis sobre dichas proporciones, este programa es capaz de devolver una única red dirigida simple, llamada *red consenso*. Más información al respecto se puede encontrar tanto en el artículo que describe al programa [12], así como en la guía de usuario que forma parte de su material suplementario.

Interacciones ecológicas

Es necesario comprender que los organismos en la naturaleza comúnmente se encuentran bajo cierta presión ambiental, como disponibilidad de recursos, lo que deriva en el concepto de competencia por la supervivencia entre especies. Con esto, las interacciones ecológicas entre organismos vivos pueden ser estudiadas dependiendo del efecto que tenga una población, o incluso un taxon en específico, sobre los demás en un ecosistema [5]. Así, los mecanismos de interacción pueden ser clasificados al saber si son benéficos, perjudiciales, o inclusive, si no tienen efecto sobre alguna de las especies interactuando. Algunos tipos de interacciones ampliamente conocidas, llamadas relaciones interespecíficas [5], son:

1. Competencia: Es descrita como una interacción perjudicial para las dos poblaciones involucradas, en especial si el recurso por el que compiten es limitado. Ejemplo: dos herbívoros compitiendo por el mismo pasto.
2. Depredación: Una interacción donde una de las especies mata y consume a otra. La interacción es benéfica para la especie superviviente y perjudicial para la otra.
3. Simbiosis: Se da cuando dos organismos se encuentran en un contacto directo. Aunque puede representar un efecto negativo sobre una de las especies, esta relación es comúnmente usada como sinónimo de mutualismo, que es un tipo de simbiosis donde ambas especies se benefician.
4. Facilitación: Son interacciones benéficas para ambas especies, o donde una de ellas no percibe efecto pero la otra se beneficia.

Se debe considerar también que el efecto que tiene un organismo sobre otro no es necesariamente recíproco o bilateral, por lo que se dice que estas interacciones son dirigidas. Por esta razón, recurriremos en este trabajo a la aplicación de grafos dirigidos para su estudio. De esta forma, se tienen dos tipos de especies en cada interacción ecológica, una primera especie que desarrolla una actividad de supervivencia, y una segunda especie que percibe el efecto de la actividad de la primera.

En general, dada una interacción que tiene un efecto perjudicial para la especie receptora, diremos que esta representa una interacción de exclusión, mientras que si la interacción representa un beneficio para esta misma especie, entonces se dirá que esta es una interacción de colaboración.

Debe señalarse que aunque estas asociaciones son comúnmente desarrolladas entre especies, también es posible extender a otros niveles taxonómicos los principios que con ellas se describen, de esta forma, podemos construir redes de interacciones ecológicas donde cada uno de los vértices de la red representa, ya sea una especie, o bien, colecciones con definiciones más amplias como familias o clases [11].

Redes de interacciones ecológicas

En una red de interacciones ecológicas, es posible representar cada interacción como una flecha saliendo de una especie y apuntando a otra, para describir el efecto que tiene la primera sobre la segunda (fig. 3.8). Con esto, se representa el efecto sobre la especie a la que apunta la flecha al asignar a esta última uno de dos símbolos: $(+)$ si la relación es favorable para dicha especie, y $(-)$ si le es perjudicial.

Al definir las redes ecológicas de esta forma, dos interacciones distintas (e.j: competencia o depredación) pueden estar representadas por flechas con un mismo signo. Dadas redes que son inferidas por medio de datos experimentales, cada signo representa indicios de una interacción ecológica que posteriormente puede ser estudiada a detalle si existe interés en determinar el mecanismo ecológico detrás de ella.

En particular, en este trabajo no estudiaremos interacciones sin efecto, por lo que nos restringiremos a estudiar relaciones cuyas flechas tienen todas alguno de los signos ya mencionados. Además, analizaremos propiedades asociadas a la red en su totalidad, más que interacciones ecológicas particulares.

Aunado a esto, debe notarse que dada una colección de flechas de interacciones entre especies en el mismo ecosistema, podemos conectar tales interacciones por medio de las especies que tienen en común, construyendo así una red con signos de las interacciones ecológicas en este ecosistema. Este procedimiento se puede desarrollar de manera recursiva para unir entre sí redes previamente formadas, o inclusive, se pueden llegar a separar las interacciones de una red en otros dos o más modelos, dependiendo por ejemplo del tipo de relación que se busque estudiar (+ o -).

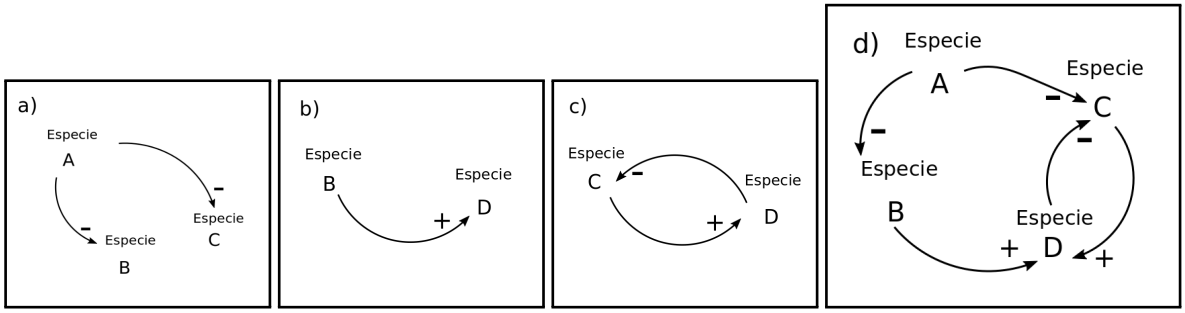

Figura 3.8: Se muestran múltiples relaciones entre especies: a) dos interacciones que pueden ser de depredación o competencia, b) una relación que puede ser de facilitación o simbiosis, c) dos relaciones donde una especie percibe un efecto benéfico pero la otra se ve perjudicada, y d) una red con signos y dirigida, formada por todas las interacciones anteriores.

Propiedades a estudiar sobre las redes de CCC

En este trabajo se analizaron múltiples propiedades de redes, que retomando nuestra clasificación de propiedades, son de dos tipos: aquellas que caracterizan a la red globalmente y otras que representan conjuntos de vértices con propiedades particulares. En [11] se reportan todas aquellas propiedades que fueron significativas para el estudio de los tapetes de CCC. Inicialmente se evaluaron las siguientes propiedades:

a) Propiedades para la red: orden, tamaño, diámetro, radio, densidad, grado promedio, grado máximo, coeficiente de agrupamiento promedio, modularidad y distribución de grado.

b) Colecciones de vértices: hubs de grado máximo, vértices de corte, componentes conexas, componentes fuertemente conexas, cliques maximales, conjuntos independientes de vértices, y comunidades detectadas por el algoritmo Louvain [52].

En esta tesis presentamos un análisis que complementa a aquel desarrollado en [11], prestando especial atención a algunas propiedades que describen la forma en la que las interacciones se encuentran distribuidas entre los OTU's, estas son: densidad total de la red, densidad de interacciones de exclusión, densidad de interacciones de colaboración, grado promedio y coeficiente de agrupamiento promedio.

Para retomar la definición de **densidad** de una red, se debe recordar primero que por **orden** de la red se entiende la cantidad de vértices que la forman, y como **tamaño** a la cantidad de interacciones existentes en ella. En general, las redes analizadas son **grafos dirigidos**, que hemos definido como grafos sin lazos ni flechas múltiples, es decir, pueden existir solo dos flechas entre un mismo par de especies, una apuntando en cada sentido, y no pueden existir flechas saliendo y entrando a una misma especie o taxon. Dada una red de este tipo y de orden n, se tiene que puede tener un tamaño máximo de $n(n-1)$, que se corresponden con las parejas ordenadas y sin repetición de los n vértices. De esta forma, se tiene que la **densidad** de una red se puede escribir como

$$\rho = \frac{m}{n(n-1)} \tag{3.1}$$

La densidad de una red podrá tomar valores en el intervalo $[0, 1]$, de forma que para una red ecológica con densidad cercana a 1, se puede intuir que aproximadamente todos los organismos u OTU's se encuentran interactuando entre sí.

En particular, para el análisis en esta tesis se definieron dos variaciones de la densidad de una red, que llamaremos por común **densidades signadas**. Aquí definimos como **densidad de exclusión**, a la proporción de flechas etiquetadas con signo menos (-) respecto al tamaño de la red. Respectivamente definimos como **densidad de colaboración** a la proporción de interacciones positivas (+) respecto al tamaño de la red. Dado que ambas densidades signadas se encuentran normalizadas respecto al tamaño de la red, entonces la suma de ambas debe ser igual a 1. Esto nos ayudará a comparar redes sujetas a distintos niveles de humedad sobre un plano de densidad de colaboración vs densidad de exclusión, y en particular sobre la recta $x+y = 1$, expresión en la que se considera la suma de ambas densidades signadas.

Otra propiedad relacionada con las proporciones de interacciones que existen en un red ecológica, es la cantidad de interacciones promedio que tiene cada especie, u OTU en general. Con esta medida, se busca evaluar el nivel o magnitud con el que interactúan los organismo de la red. Esto se puede estudiar con el concepto de **grado promedio** $< k >$ de una red, que se puede escribir como

$$< k >= \frac{m}{n} \tag{3.2}$$

Si el grado promedio es cercano a $n - 1$, entonces, de manera similar a la densidad, podemos prever que existe un gran cantidad de interacciones entre los organismos de la red.

Por último, buscaremos también medir, dado un vértice v de nuestra red, qué tantas interacciones se desarrollan entre los vértices conectados a v. Esto es evaluado por medio del **coeficiente de agrupamiento promedio** de la red, definido como la media aritmética del **coeficiente de agrupamiento local** de los vértices en la red. Ambos coeficientes tienen valores también entre 0 y 1, de modo que pueden representar la probabilidad de que para dos vértices tomados al azar en la red, ambos tengan un efecto o sean influenciados por un tercero al mismo tiempo. Para estudiar que tan significativas son todas estas medidas en las redes de interacciones ecológicas proporcionadas por la Dra. de Anda, compararemos estos resultados con sus equivalentes evaluados en redes generadas aleatoriamente.

Aleatorización de redes

Dada cualquier red, es posible intercambiar las conexiones que existen en esta. En esencia, el proceso básico de aleatorización de redes se puede resumir en dos pasos: primero se escoge y borra aleatoriamente una relación ya existente en la red, y posteriormente se procede a añadir otra relación que no existía antes en ella. En resumen, esto consiste en substituir de manera aleatoria algunas relaciones en la red.

La importancia de procedimientos como este reside en que, al calcular las propiedades de una red es posible comparar, por ejemplo, por medio de una prueba de hipótesis [45], que tan estadísticamente significativas son estas respecto a las calculadas sobre las redes que fueron generadas con el proceso de aleatorización. De este modo, se puede evaluar la relevancia de aquellas propiedades de redes que modelan sistemas reales, en caso de que se sospeche que cierta propiedad no es producto del azar y que puede llegar a tener una interpretación derivada de las propiedades del sistema estudiado.

Es importante señalar que el proceso antes mencionado no es la única forma de aleatorizar redes. En este procedimiento, se estableció como constante al número total de interacciones, aunque estas fueran modificadas. En general, una **red aleatoria** es aquella para la cual, algunas de sus propiedades permanecen constantes bajo procesos de aleatorización de otras de sus características. Existen muchos procedimientos tanto para aleatorizar redes, como para producir redes aleatorias, y todos estos conforman un área de investigación por sí mismos [1].

Motivos de redes

Los motivos de redes, o *network motifs*, definidos originalmente en [42], son patrones de conexiones, que aparecen reiteradamente en una misma red. Sin embargo, no todo patrón repetido en una red califica como motivo. Para esto, estas configuraciones de relaciones deben estar presentes en una cantidad significativamente mayor que en redes aleatorias, generadas a partir de la red que estemos estudiando. Lo antes mencionado conlleva desarrollar una prueba de hipótesis [45] sobre la abundancia de los motivos en una red. De este modo, para determinar si un cierto patrón cumple ser un motivo, será necesario evaluar si la presencia de este en una colección de redes aleatorias es estadísticamente significativa.

En concreto, un **motivo de red** es un subgrafo inducido conexo sobre el grafo que constituye nuestra red, que a su vez resulta ser estadísticamente significativo en comparación con redes aleatorias generadas con el mismo número de vértices que la red original. En general, no existen reglas respecto a la cantidad de vértices que deban componer a un motivo, ni como deban desarrollarse las conexiones entre estos vértices. Es decir que para detectar un motivo por fuerza bruta en una red cualquiera, sería necesario llevar a cabo la búsqueda de todos los posibles subgrafos inducidos de la red, lo que deriva en dos problemas complejos. El primero de estos problemas, comprende calcular todas las posibles colecciones de vértices de una red, proceso que consume mucha memoria computacionalmente, ya que depende del conjunto potencia de los vértices de nuestra red. Posteriormente, para clasificar a los subgrafos como cierto tipo de motivo, es necesario evaluar si existe un isomorfismo entre cada subgrafo inducido y todos los otros que tengan el mismo número de vértices, lo que además de memoria, requiere de tiempo, ya que conlleva muchas operaciones computacionales.

Debido a lo anterior, los estudios donde se analizan motivos se desarrollan entorno a subgrafos inducidos pequeños (3-5 vértices), o a algunos patrones específicos que ya han sido reconocidos como motivos en redes biológicas [26, 33, 53]. En estos estudios se ha encontrado que ciertos motivos, por ejemplo en redes de regulación genética de la bacteria E. Coli (fig. 3.9), optimizan el uso de la energía en los procesos químicos, por lo que su existencia en estas redes se explica como una ventaja evolutiva para la célula.

Al mismo tiempo, otros motivos encontrados también en E. Coli, dan razón de la capacidad de una célula a reaccionar a estímulos externos, ya que se encuentran compuestos por un gen cuya autorregulación determina la expresión de otra colección de genes. Sin embargo, en lo que a redes de interacciones ecológicas concierne, únicamente se ha logrado encontrar en nuestra revisión bibliográfica, investigaciones sobre redes tróficas con un enfoque principalmente teórico [34].

El mismo grupo de investigación que definió a los motivos [42], desarrolló también un software llamado MFinder [35], que permite llevar a cabo la búsqueda de motivos compuestos desde dos, y hasta seis vértices. Junto con este, se encuentra también disponible al público, un catálogo de algunos de los subgrafos dirigidos de tres y cuatro vértices, clasificados por un *id* que los relaciona con el programa MFinder. Dicho programa fue utilizado en el presente trabajo para analizar los motivos con mayor presencia en las redes ecológicas de tapetes microbianos, permitiéndonos llevar a cabo su posterior interpretación biológica.

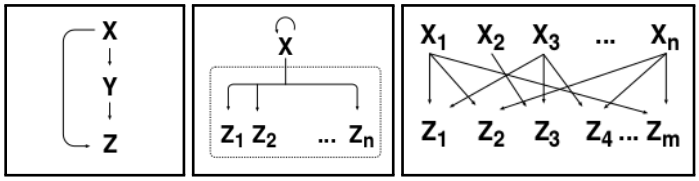

Figura 3.9: Motivos de red ampliamente estudiados en E. Coli
(Imagen tomada y adaptada de [42])

3.4 Metodología

Este trabajo consistió en el estudio de las propiedades y motivos de redes ecológicas, inferidas de tapetes microbianos originados en la reserva ecológica de la cuenca de Cuatro Ciénegas (CCC), Coahuila [11]. Para describir el contexto dentro del cual se analizaron las redes, se menciona a continuación el trabajo desarrollado por la Dra. de Anda y su equipo durante la recolección y procesamiento de las muestras biológicas. Seguido a esto se habla sobre el análisis computacional de las propiedades de redes.

Recopilación y procesamiento de las muestras de los tapetes micribianos

Se recolectaron muestras en un estanque llamado *La Lagunita*, adyacente a la Laguna Churince en CCC. Dicho estanque presenta distintos niveles de profundidad, no mayores a 0.42cm, en condiciones normales de abundancia de agua. En él se definieron 3 sitios de muestreo cercanos entre sí, un sitio seco (sitio A) y dos sitios húmedos (sitio B y sitio C).

Se llevaron a cabo cuatro muestreos para cada sitio, correspondientes cada uno a cuatro tiempos entre los años 2012 y 2014. Estos cuatro tiempos fueron seleccionados por la Dra. de Anda y su equipo, buscando estudiar diferentes niveles de agua. La primera muestra fue tomada en Noviembre de 2012, la segunda en Mayo de 2013, la tercera en Octubre de 2013, y la última en Mayo de 2014. Después de esto, las 12 muestras fueron procesadas por la Dra. de Anda. Posteriormente, el ADN genómico extraído de ellas, fue secuenciado en el CINVESTAV-LANGEBIO, en Irapuato, Guanajuato, por medio de secuenciación Shotgun. Finalmente, se utilizó el software One Codex [50] para determinar la clasificación taxonómica de las secuencias.

Inferencia de redes con MetaMIS

La inferencia de las redes de interacciones ecológicas fue hecha por la Dra. de Anda por medio del software MetaMIS [12]. A dicho programa le fueron proporcionadas series de tiempo de la abundancia absoluta de los organismos u OTU's (Operative Taxonomic Unit) en los tapetes, cantidad que depende de las repeticiones que tiene cierta secuencia de ADN genómico en una misma muestra. De este procedimiento se obtuvieron 12 redes consenso, dirigidas, sin lazos y con signos (interacciones de colaboración $+$, e interacciones de exclusión $-$), una por cada nivel taxonómico: familia, orden, clase, y filo, a partir de cada sitio de muestreo. En la figura 3.10 se muestra la cantidad de redes de interacción que MetaMIS generó para desarrollar la inferencia de estas 12 redes.

Posteriormente, de cada red consenso se extrajeron otras dos, una compuesta puramente por relaciones positivas (+) y otra por relaciones negativas (-). Además, para cada nivel taxonómico se constituyó una red llamada global, correspondiente a la unión de las redes consenso a un mismo nivel en los tres sitios de muestreo, que a su vez fue nuevamente separada en una red global positiva y otra negativa. Con todo lo anterior, se obtuvieron finalmente 48 redes que formaron la base para el análisis de propiedades de redes presentado en [11].

Análisis de propiedades de redes

Implementamos un programa en el lenguaje de programación Python, que por medio de las bibliotecas NetworkX y Python-Louvain [30, 54], llevó a cabo el análisis de algunas propiedades para redes dirigidas, y algunas otras para redes no dirigidas evaluadas sobre el grafo subyacente de cada red dirigida. Estas propiedades son:

1. Propiedades para la red: orden, tamaño, diámetro, radio, densidad, grado promedio, grado máximo, coeficiente de agrupamiento promedio, modularidad y distribución de grado.
2. Colecciones de vértices: hubs de grado máximo, vértices de corte, componentes conexas, componentes fuertemente conexas, cliques maximales, conjuntos independientes de vértices, y comunidades detectadas por el algoritmo Louvain [52].

Sitio	Taxon	Redes de Interacción
A	Filum	78
	Clase	130
	Orden	144
	Familia	169
B	Filum	86
	Clase	138
	Orden	154
	Familia	161
C	Filum	86
	Clase	146
	Orden	143
	Familia	165

Figura 3.10: Cantidad de redes de interacción generadas por cada red consenso.
(Información del material suplementario de [11])

Además de evaluar las propiedades ya mencionadas, el programa genera 100 redes aleatorias por cada red que se le proporcione, preservando su cantidad de vértices y flechas. Así, devuelve también los valores promedio de algunas propiedades evaluadas sobre las redes producidas aleatoriamente, de modo que estas pueden ser contrastadas con las propiedades de las redes originales.

Por medio de este programa, se desarrolló el análisis de todas las propiedades antes mencionadas para cada una de las 48 redes. La ejecución de este programa se llevó a cabo en un servidor provisto por el Laboratorio de Visualización Científica (LAVIS) [41], del Instituto de Neurobiología de la UNAM, campus Juriquilla. Tanto el programa desarrollado, bajo el nombre *NetworkAnalysis.py*, así como ejemplos e instrucciones para ejecutarlo, pueden ser encontrados en el repositorio público: `https://github.com/valdeanda/NetAn`, que es administrado por la Dra. de Anda.

Análisis de motivos de redes

De igual manera, la detección de los motivos de red se realizó haciendo uso del servidor provisto por el LAVIS. Esto se hizo para cada una de las 48 redes, por medio del programa MFinder [35], usando opciones default de ejecución y sin considerar los signos asociados a las interacciones ecológicas, ya que actualmente dicho software no contempla el uso de pesos ni etiquetados en su análisis.

Dicho programa es capaz de detectar motivos compuestos por máximo 6 vértices, sin embargo, se desarrolló la detección solo para motivos de 3 vértices, dado que los de mayor orden requerían un tiempo de ejecución que hacia imposible la conclusión del análisis.

3.5 Resultados

Con propósito de describir el contexto del análisis, se resumen algunas de las observaciones realizadas por la Dra. de Anda y su equipo respecto a la diversidad biológica de los tapetes, así como los resultados relativos a los motivos de redes. Posteriormente, se incluyen los datos utilizados para el análisis en esta tesis, seguidos por una discusión en torno a ellos.

Entorno microbiano de los tapetes

Tomando en cuenta a los tres sitios, se detectaron en total: 539 familias, 302 ordenes, 168 clases y 100 phyla. En particular, se observó que el sitio B resulto ser menos diverso que los otros dos, específicamente durante los tiempos caracterizados por un menor nivel de agua. Por otro lado, el sitio A, el más seco, presentó una mayor diversidad taxonómica. Otra diferencia estadística entre los sitios indica que en aquellos con más humedad (B y C), existe una mayor cantidad de organismos anaeróbicos, aunado a que los organismos en el sitio B resultan ser principalmente fotosintéticos y heterótrofos.

Motivos de las redes de tapetes microbianos

De entre las 48 redes[35], se prestó especial interés en los motivos presentes en cada uno de los dos grupos de 16 redes con un mismo signo, obtenidas al separar las interacciones positivas de las negativas en las doce redes consenso y las cuatro globales, para todo nivel taxonómico. De esta forma se obtienen motivos constituidos, o bien por interacciones positivas, o por interacciones negativas, pero no por relaciones de ambos tipos. En la figura 3.11 se muestran 4 de estos motivos, que en la nomenclatura del programa MFinder son los que tienen *id*: 6, 36, 12, y 46, a los que se les dio una interpretación ecológica junto con la Dra. Niza Gámez Tamariz, colaboradora de la Dra. de Anda, y también perteneciente al Instituto de Ecología de la UNAM.

Los primeros dos (6 y 36), se encontraron en las redes con signo positivo, lo que sugiere que están compuestos por interacciones de facilitación. Mientras, los últimos dos (12 y 46), se detectaron para las redes de interacciones con signo negativo, lo que nos permite formular para ellos una interpretación como patrones de interacciones de competencia.

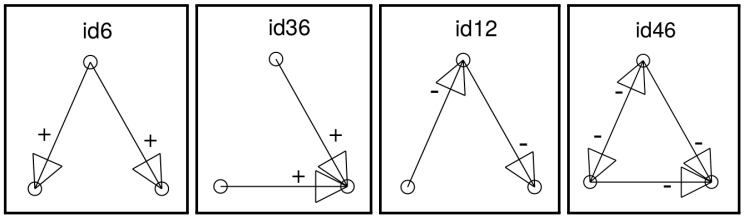

Figura 3.11: Motivos presentes en los tres sitios y en todo nivel taxonómico. Los motivos (6) y (36) sugieren interacciones de facilitación, mientras que los motivos (12) y (46) refieren a relaciones tróficas y/o de competencia [11]. *(Imagenes tomadas y adaptadas de [35])*

Organismos con un mayor número de interacciones (hubs)

De entre todos los organismos, se analizaron para las 48 redes, aquellos que tuvieron grado máximo de entrada o salida, llamados hubs (fig. 3.12). Ya que estos organismos poseen un gran número de interacciones, es posible que sus características tengan una fuerte influencia en la dinámica de toda la red. Se encontró por ejemplo, que el filo *Verrucomicrobia* es un hub para el sitio C. Dicho filo es conocido por desarrollar relaciones de mutualismo [55, 11], lo que favorece a la demás phyla en su red.

Propiedades analizadas en esta tesis

En esta tesis se llevó a cabo un análisis que complementa a aquel desarrollado junto con la Dra. de Anda, prestando especial atención a algunas propiedades que hablan sobre la distribución de las interacciones ecológicas entre los organismos de las 12 redes consenso, estas son: densidad total de la red, densidad de interacciones de exclusión, densidad de interacciones de colaboración, grado promedio y coeficiente de agrupamiento promedio.

Los valores de estas medidas para las redes consenso se encuentran resumidas en las tablas de las figuras 3.13 y 3.14, redondeadas a 5 decimales. Dado que para todos los niveles taxonómicos, la red en el sitio C fue la que presentó mayor grado promedio, se decidió normalizar para cada taxon el grado promedio de cada sitio respecto al del sitio C. De esta forma, se presenta más adelante un análisis comparativo entre los grados promedios normalizados de las 12 redes consenso.

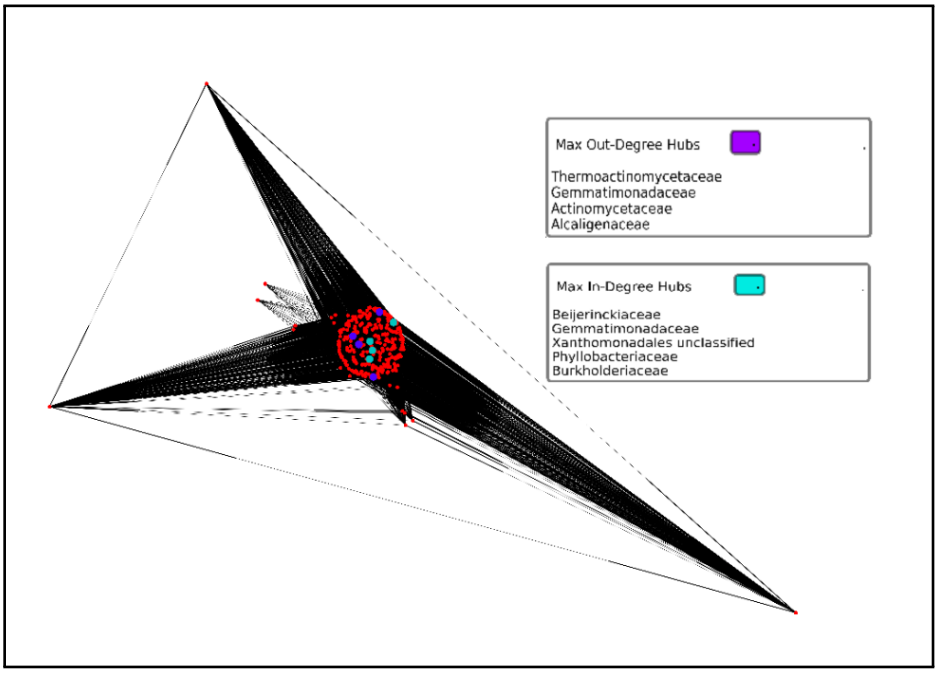

Figura 3.12: Red consenso del sitio B a nivel Familia. Se resalta en morado los hubs para grado de salida, y en azul los de grado de entrada.

3.6 Discusión

En las imágenes 3.13 y 3.14, que condensan los valores referentes a algunos atributos que describen la distribución de las interacciones en las redes consenso, se incluyen también los valores de coeficiente de agrupamiento promedio obtenidos para las 100 redes aleatorias generadas con el script *NetworkAnalysis.py*. Se debe recordar que estas fueron producidas preservando la cantidad de vértices e interacciones entre ellos, con lo que la densidad y el grado promedio se preserva de igual manera.

Para las redes reales, se aprecian valores altos de agrupamiento promedio ($\sim$0.95), que además resultan ser semejantes al promedio calculado para redes aleatorias ($\sim$0.90), aunque las reales son ligeramente mayores en promedio. Los altos valores para la densidad ($\sim$0.90) y agrupamiento promedio, así como el parentesco de este último con su semejante en redes aleatorias, sugieren que las interacciones en las redes consenso no cuentan con ninguna característica en particular que las pueda volver frágiles ante cambios menores en su entorno, como podría ser quedar desconectadas ante la pérdida de un único organismo. Esto permite intuir que estas redes son robustas ante pequeñas perturbaciones.

Para poder estudiar más a fondo el efecto de los niveles de agua sobre la redes, se pueden analizar las relaciones de los valores en la tabla 3.14, al construir una gráfica de densidad de exclusión contra densidad de colaboración, mostradas en las figuras 3.15 para las 12 redes consenso coloreadas por grupo taxonómico, y 3.16 para estas mismas 12 redes coloreadas por sitio de muestreo.

Grupo Taxonómico	Sitio	Orden	Tamaño	Densidad Total	Grado Promedio	Grado Promedio Norm.	Coeficiente de Agrupamiento Promedio	
							Reales	Aleatorias
Familia	A	354	108439	0.86778	306.32486	0.99254	0.93706	0.86778
	B	332	98640	0.89761	297.10843	0.96268	0.95441	0.89761
	C	354	109254	0.87430	308.62712	1.00000	0.94514	0.87430
Orden	A	207	38543	0.90387	186.19807	0.98501	0.94467	0.90387
	B	205	38399	0.91820	187.31220	0.99090	0.95099	0.91820
	C	219	41398	0.86712	189.03196	1.00000	0.94474	0.86712
Clase	A	91	7888	0.96313	86.68132	0.75431	0.98529	0.96313
	B	123	13688	0.91217	111.28455	0.96841	0.94619	0.91217
	C	129	14824	0.89777	114.91473	1.00000	0.95081	0.89775
Filo	A	61	3560	0.97268	58.36066	0.92402	0.98131	0.97266
	B	68	4167	0.91462	61.27941	0.97023	0.95158	0.91463
	C	69	4358	0.92882	63.15942	1.00000	0.95974	0.92882
							Media: 0.95433	**Media:** 0.90984

Figura 3.13: Resultados para las propiedades de: orden, tamaño, densidad, grado promedio y coeficiente de agrupamiento promedio. Se incluyen también los valores de grado promedio normalizado respecto al del sitio C para cada taxon.

Nivel Taxonómico	Sitio	Tamaño	Int. Pos	Int. Neg	Densidad Positiva	Densidad Negativa
Familia	A	108439	39888	68551	0.36784	0.63216
	B	98640	33385	65255	0.33845	0.66155
	C	109254	51325	57929	0.46978	0.53022
Orden	A	38543	14648	23895	0.38004	0.61996
	B	38399	13775	24624	0.35873	0.64127
	C	41398	20573	20825	0.49696	0.50304
Clase	A	7888	2931	4957	0.37158	0.62842
	B	13688	4950	8738	0.36163	0.63837
	C	14824	7669	7155	0.51734	0.48266
Filum	A	3560	1424	2136	0.40000	0.60000
	B	4167	1417	2750	0.34005	0.65995
	C	4358	2238	2120	0.51354	0.48646

Figura 3.14: Resultados para las mediciones de densidades signadas.

En estas gráficas se aprecian doce puntos, uno por red sobre la recta $x + y = 1$, para x e y ambas en $[0, 1]$, donde x es la densidad de exclusión de cada red, mientras que y es la densidad de colaboración para estas. Al comparar ambas gráficas, se observa que, de forma independiente al grupo taxonómico, se presenta un orden concreto entre los sitios de muestreo.

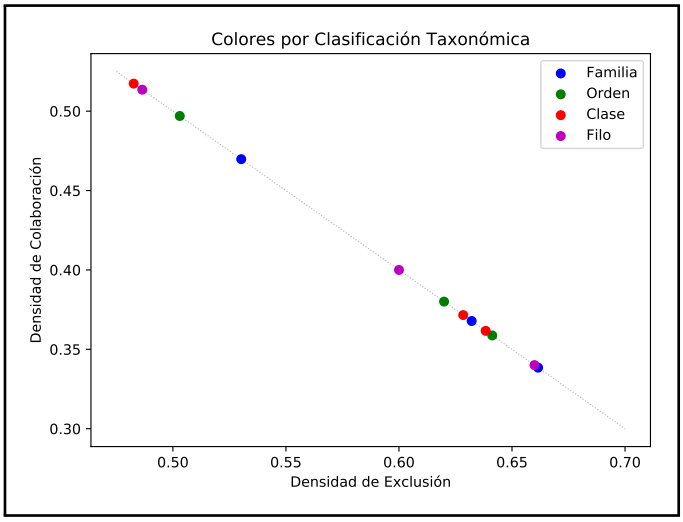

Figura 3.15: Rectas para densidades signadas. Cada punto representa una red consenso, coloreadas por nivel taxonómico.

Con estas se aprecia que para el sitio C, uno de los sitios con humedad, las interacciones positivas se encuentran equilibradas (∼0.5/∼0.5) con las interacciones negativas, mientras que para el sitio A, que es el sitio seco, se aprecia un incremento en la cantidad de interacciones negativas (competencia o depredación). Sin embargo, más adelante sobre la recta se encuentran las redes del sitio B, que presenta la mayor proporción de interacciones negativas a pesar de ser el otro sitio húmedo. Para aclarar este resultado es necesario estudiar la cantidad de interacciones que tienen los organismos en promedio.

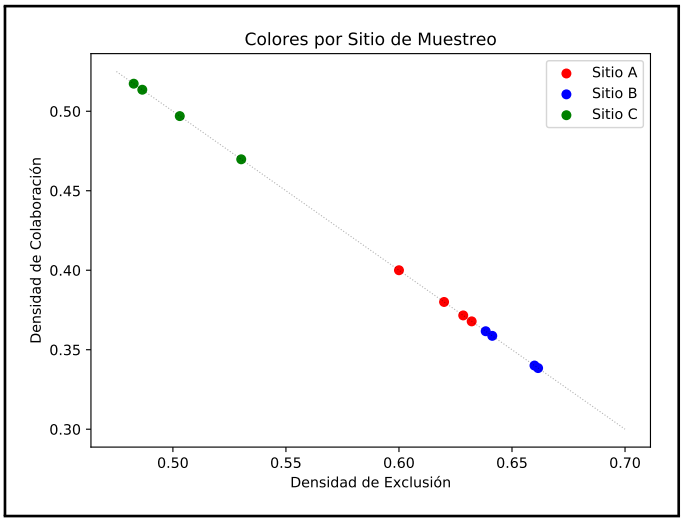

Figura 3.16: Rectas para densidades signadas. Cada punto representa una red consenso, coloreadas por sitio de muestreo.

Graficando los valores de grado promedio normalizado (fig. 3.17), encontrados en la tabla de la figura 3.13, tomados para cada nivel taxonómico, se observa que en todos, a excepción del nivel familia, el sitio B tiene un grado promedio mayor al sitio A.

De esta forma, y de igual manera que con las propiedades de densidad y agrupamiento promedio, que muestran como es posible que las redes sean robustas a perturbaciones, un mayor grado promedio puede indicar una mayor capacidad de responder a abundantes interacciones negativas, al regular o atenuar el efecto de cada una de ellas sobre cada organismo.

Recordando que algunos de los organismos en el sitio B producen su alimento por medio de fotosíntesis, mientras que otros son heterótrofos, se sugiere que este sitio presenta, debido a su particular diversidad taxonómica, una natural tendencia a formar interacciones de competencia o depredación, siendo a la vez capaz de soportarlas a largo plazo.

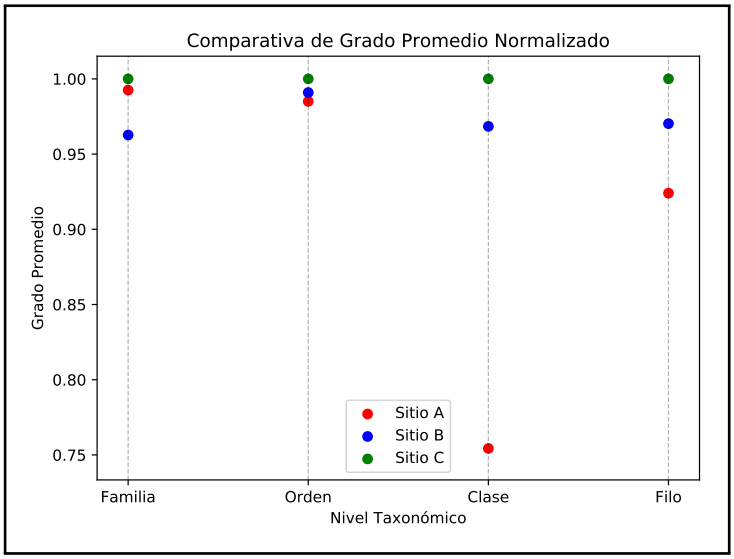

Figura 3.17: Grado promedio normalizado en todo nivel taxonómico.

Más aún, el sitio A presenta el menor grado promedio de los tres sitios y en todos los niveles taxonómicos, a excepción del nivel de Familia. Junto con la proporción de relaciones negativas que se desarrollan en este sitio, se sugiere que los organismos que habitan en este tapete se pueden encontrar bajo un estrés ecológico mayor, comparado con los organismos en los otros sitios.

Teniendo en mente que el sitio C, siendo un sitio húmedo, presenta un balance entre las densidades de relaciones positivas y relaciones negativas, se intuye que el estrés al que se puede encontrar sometido el tapete en el sitio A, es debido a la escasez de nutrientes provocada por los bajos niveles de agua.

Estos resultados no nos permiten refutar la hipótesis presentada en la introducción de esta tesis. Sin embargo, para poder afirmar que existe una presión ambiental sobre el tapete del sitio A, y esclarecer si el estrés ecológico al que se ven sometidas estas redes se debe estrictamente a la reducción en los niveles de agua, es necesario contar con un mayor número de muestras o redes consenso por nivel taxonómico, y en una mayor cantidad de sitios de muestreo.

3.7 Recomendaciones

En este capítulo se exhibieron los detalles relacionados con el análisis de redes. En particular, se comentó que uno de los aspectos más difíciles es llevar a cabo la inferencia de la red. Esto es un problema tecnológico y un área de investigación en sí mismo, por lo que es recomendable siempre desarrollar una búsqueda respecto al estado del arte en la inferencia de redes sobre el sistema biológico específico que se quiera analizar.

En particular, se recomienda tener precaución al utilizar el sistema MetaMIS para inferir redes microbianas, ya que a la fecha este programa no contempla las limitaciones descritas respecto a la inferencia por medio de los modelos Lotka-Volterra sobre datos de abundancia relativa [47], ni respecto a la posibilidad de inferir múltiples redes igualmente válidas a partir de una serie de tiempo [48]. Por esto, se considera que es importante estudiar a detalle el impacto de tales problemáticas sobre la inferencia de redes con MetaMIS.

Por otro lado, se recomienda buscar la posibilidad de realizar la detección de motivos en redes tomando en cuenta el tipo de interacción ecológica (+ o -). Con esto se podrían estudiar motivos compuestos tanto por interacciones de colaboración como de exclusión, haciendo más preciso el análisis e interpretación de estos subgrafos.

Actualmente el lenguaje de programación Python representa una gran ventaja en comparación con otros lenguajes, ya que este está constituido por instrucciones con un alto nivel de abstracción, es decir, próximas al lenguaje humano. De esta forma, es posible desarrollar un análisis en Python, enfocándose más en el estudio por sí mismo que en la estructura del código a implementar, a diferencia de lenguajes donde se debe procurar en todo momento la cantidad de memoria disponible.

Junto con lo anterior, se debe considerar también que en dicho lenguaje se encuentran desarrolladas ya muchas bibliotecas para investigaciones muy específicas, como lo es el análisis de redes, por lo que se cuenta ya con bases informáticas para iniciar una investigación.

Se debe considerar también que al analizar redes, es necesario contar con una solida capacidad de procesamiento, preferiblemente proporcional al número de vértices que constituyan nuestras redes. Esto se debe a que, aunque muchos problemas en redes sí presentan soluciones concretas, estas pueden depender de una cantidad considerable de tiempo, prolongándose así la obtención de resultados si el equipo de computo no es adecuado.

Finalmente, en este capítulo se analizaron los efectos de la disminución de los nutrientes de los que disponen los organismos en redes de interacciones ecológicas, al relacionarlos con la cantidad de interacciones de colaboración y competencia en ellas. Sin embargo, se recomienda que para poder extraer conclusiones concretas a partir de las propiedades de una red, siempre se busque contar con una alta cantidad de ejemplares.

Capítulo 4

Detección de comunidades en redes

El problema de detección de comunidades en grafos [6] conlleva particionar el conjunto de vértices de un grafo en grupos llamados comunidades (fig. 4.1), de forma que existan más aristas entre vértices de la misma comunidad que entre vértices de distinta comunidad. Lo anterior se ve motivado por la idea de que las comunidades pueden representar agrupamientos significativos bajo la interpretación que se le de al grafo como una red [1, 16].

4.1 Problema tecnológico: detección de comunidades en grafos por optimización de la modularidad, y su aplicación a la red neuronal del nematodo *C. Elegans*

La detección de comunidades es un problema que a la fecha no se encuentra rigurosamente definido [6] y su solución por fuerza bruta es computacionalmente impráctica [56], ya que requeriría evaluar qué tan significativas son todas y cada una de las posibles particiones del conjunto de vértices de un grafo.

Actualmente existen múltiples métodos de detección de comunidades [6, 57], sin embargo ninguno de ellos ha sido ampliamente aceptado por distintos motivos, que involucran desde elevados tiempos de ejecución, hasta ciertos problemas respecto a los agrupamientos que con ellos se detectan.

Uno de los métodos más usados para detectar comunidades en grafos, conlleva evaluar la relevancia de una partición de los vértices del grafo por medio de la medida conocida como modularidad [14], siendo que dadas múltiples particiones, aquella que maximice la modularidad representará la partición más significativa.

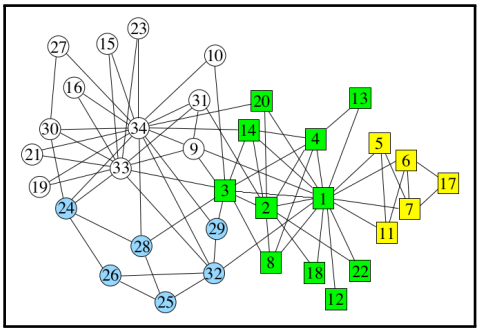

Figura 4.1: Ejemplo de un grafo de relaciones sociales particionado en comunidades diferenciadas por colores.
(Imagen tomada y adaptada de [6]).

No obstante, se ha mostrado que la evaluación de las particiones por medio de la modularidad tiene un problema llamado límite de resolución [15]. Este implica que, dada una partición con modularidad óptima, es posible encontrar sub-grupos más significativos dentro de cada uno de los propuestos por la partición original, siendo que la partición del conjunto de vértices en estos sub-grupos proporcionaría una menor modularidad.

Apoyados en los trabajos de Newman y Girvan [14, 16], y basados en la definición de probabilidad condicional [17], desarrollamos en este trabajo una formulación para la medida de modularidad. Se debe mencionar que en la revisión bibliográfica no se encontró referencia a una formulación semejante. Posteriormente, motivados por estudiar las posibles alternativas al problema del límite de resolución de la modularidad, y basados sobre nuestra formulación para esta medida, desarrollamos y evaluamos un método de detección de comunidades en grafos no dirigidos y conexos.

Llevamos a cabo la evaluación de este método por medio de los grafos aleatorios de prueba (Benchmark Graphs) llamados Planted-Graphs [14] y modelos-LFR [29]. Ambos modelos son comúnmente utilizados para evaluar trabajos de detección de comunidades en redes [6] y actualmente están implementados en la biblioteca NetworkX [30] del lenguaje de programación Python.

Por último, dadas las potenciales aplicaciones que la detección de comunidades presenta para la neurobiología, específicamente para estudiar la relación entre la estructura del cerebro humano con su comportamiento cognitivo [8, 58], pero tomando en cuenta que este trabajo se desarrolla en torno a la evaluación de nuestro método, desarrollamos un análisis sobre las comunidades de la red neuronal del nematodo *C. Elegans*. De este modo, se pudieron comparar los agrupamientos detectados por nuestro método contra los que ya han sido ampliamente estudiados en la literatura, tanto en el contexto de detección de comunidades [19], como en referencia a las características biológicas de dicha red [19, 20, 21].

En este capítulo presentamos primero una introducción a los conceptos básicos de la detección de comunidades en grafos. Posteriormente se desarrolla el marco teórico con los conceptos fundamentales para este trabajo. Por último, se describen las definiciones desarrolladas en torno a la modularidad y nuestro método, seguidas por la metodología y resultados de la evaluación de este procedimiento.

4.2 Introducción a la detección de comunidades en grafos por optimización de la modularidad

Para desarrollar nuestra introducción seguiremos lo planteado en el trabajo de Fortunato [6], específicamente en la sección sobre los elementos y nociones básicas de la detección de comunidades en grafos. En dicho trabajo, se remarca que el reconocimiento de comunidades es un problema teórico que a la fecha no se encuentra estrictamente definido. Debido a esto, se señala también que existen múltiples procedimientos para detectar estos agrupamientos, sin que estos se basen en definiciones comunes respecto a lo que se entiende por una comunidad.

No obstante, la mayoría de los métodos para el estudio de estos agrupamientos [14, 16, 57, 59, 60] basan sus hipótesis en torno a una noción de comunidad. En esta, se considera que dada una partición de los vértices de un grafo en comunidades, deberán existir más aristas uniendo vértices que se encuentran en una misma comunidad que conectando vértices en distintas comunidades.

En este trabajo buscaremos estudiar algunas de las hipótesis que se derivan de tal noción, unas originalmente discutidas en los trabajos de Girvan y Newman [16, 14], y otras que hacen referencia al problema, planteado por Fortunato y Barthélemy [15], del límite de resolución de la modularidad. A continuación se resumen algunos de los aspectos básicos relacionados a la noción de comunidad, describiendo primero el problema computacional que la detección de estos agrupamientos conlleva. Posteriormente se presenta la definición original de modularidad, seguida por algunos conceptos que serán de ayuda en el estudio del límite de resolución de esta medida.

Problema computacional de la detección de comunidades

Dos aspectos importantes en el estudio y comparación de algoritmos son, la cantidad de memoria que estos deben utilizar, y la cantidad de operaciones que deben realizar al resolver los problemas para los que fueron diseñados [2]. En este apartado se discute la cantidad de operaciones que conlleva estudiar todas y cada una de las posibles particiones de un conjunto.

Es necesario mencionar que la cantidad de operaciones que debe desarrollar un algoritmo, conocida también como **complejidad en tiempo** del algoritmo o tiempo de ejecución de este, se estudia por medio de la llamada **notación de la gran O** [56]. Por medio de esta se busca dar una función, dependiente de la cantidad de datos a analizar, que acote superiormente a la cantidad de operaciones en cada paso del algoritmo en cuestión. De esta forma, cuando se dice que un algoritmo presenta una complejidad en tiempo de $O(f(n))$, se hace referencia a que la máxima cantidad de operaciones que tal algoritmo debe desarrollar sobre n datos para cumplir su objetivo, será precisamente de **orden** $f(n)$.

En particular, es común que el tiempo de ejecución de los algoritmos en grafos se exprese en términos de la cantidad de vértices y aristas que estos tienen. Así, dado un grafo de orden n y tamaño m, comúnmente se encontrarán algoritmos con complejidades de la forma $O(n^a m^b)$, para algún par de números enteros no negativos a y b. Los tiempos de ejecución escritos de esta forma son llamados **polinomiales** [6].

Existen también tiempos de ejecución conocidos como **no polinomiales**, de los que son ejemplo, sobre n datos, cantidades de operaciones que se pueden expresar en términos de $n!$ o 2^n. El inconveniente con tales expresiones, radica en que el total de operaciones que desarrollarían algoritmos con estas complejidades, crece desmedidamente respecto a pequeños incrementos en la cantidad de datos. Esto se traduce finalmente en tiempos físicos que vuelven prácticamente imposible determinar la solución del problema para el que fueron planteados [56].

Este es el caso del problema de estudiar todas las particiones de un conjunto, por ejemplo de n vértices en un grafo, ya que la cantidad de todas las posibles particiones es dada por el número de Bell [2], para el que entre sus múltiples propiedades, se conocen expresiones que dependen de $n!$ [61], con lo que el estudio exhaustivo de estas particiones resulta computacionalmente impráctico.

De esta forma, para desarrollar el reconocimiento de comunidades en grafos, se requieren métodos que den soluciones óptimas o aproximadas, respecto a lo que se entiende por una comunidad [6]. Esta es la razón por la que los métodos planteados para el análisis de este problema comúnmente parten de hipótesis en torno a la noción de comunidad.

Evaluación de particiones por medio de la modularidad

Para poder determinar si una partición de los vértices de un grafo representa comunidades en este, es necesario contar con una medida que indique objetivamente la coherencia de las clases en dicha partición respecto a la noción de comunidad. Buscando evaluar esto, Newman y Girvan propusieron originalmente la definición de modularidad de un grafo no dirigido respecto a una partición [14].

Esta es una medida que toma valores en el intervalo $[-1, 1]$, y sirve para estudiar qué tan parecidos a comunidades son los grupos dados por una partición. De forma muy general, si para una partición la modularidad adquiere un valor cercano a 1, los grupos en ella serán coherentes con la noción de comunidad. Por otro lado, si esta es cercana a -1, entonces la partición representa grupos que permiten la existencia de más aristas entre vértices de distinta comunidad que dentro de las mismas comunidades.

Además, para todo grafo no dirigido, si se considera una partición en la que todos los vértices están dentro del mismo grupo, entonces la modularidad será igual a 0. Así, es común que en la práctica una modularidad entre 0.3 y 0.7 resulte significativa respecto a la noción de comunidad [14].

La modularidad es una medida basada en un **modelo nulo** o grafo aleatorio [6, 14]. Lo anterior quiere decir que para la modularidad, la noción de comunidad también comprende la existencia de más aristas dentro de las comunidades en comparación con las que existirían en esos mismos grupos si el grafo analizado hubiera sido generado aleatoriamente.

Para explicar esto, recurriremos a la forma general para la modularidad planteada en [6]. Aquí, se denota como P_{uv} al valor esperado de la cantidad de aristas entre los vértices u y v, respecto a un modelo nulo cualquiera, permitiendo en general que exista más de una arista entre cada par de vértices. Con esto, dada una partición $\tilde{P}$, se tiene que la **modularidad** Q respecto a $\tilde{P}$, de un grafo no dirigido $G = (V, E)$ de tamaño m, se puede calcular como la suma sobre toda pareja ordenada de vértices

$$Q = \frac{1}{2m} \sum_{V \times V} (A_{uv} - P_{uv}) \delta(C_u, C_v) \tag{4.1}$$

donde $V \times V$ es el producto cartesiano del conjunto de vértices de G con sigo mismo, A_{uv} es 1 si existe en G una arista entre u y v, y es 0 si no, mientras que $\delta(C_u, C_v)$ es una función que vale 1 cuando $C_u = C_v$ y 0 si son diferentes, en la cual, C_u y C_v son los grupos en $\tilde{P}$ a los que pertenecen u y v respectivamente.

Para formular su modularidad, Newman y Girvan plantearon el modelo nulo de esta medida para considerar preservar en este la secuencia de grado del grafo. Esto deriva en el modelo para generar grafos aleatorios llamado **modelo de configuraciones** [6].

En este último, para conservar la secuencia de grado de un grafo al aleatorizar las aristas en él, se deben considerar todas las parejas ordenadas de vértices. Esto da razón de desarrollar la suma en la ecuación (4.1) sobre $V \times V$. Además, para dicho modelo se cumple que $P_{uv} = \frac{k_u k_v}{2m}$, con lo que la **modularidad** original queda escrita como

$$Q = \frac{1}{2m} \sum_{V \times V} (A_{uv} - \frac{k_u k_v}{2m}) \delta(C_u, C_v) \tag{4.2}$$

donde k_u y k_v son cada uno los grados de los vértices u y v. Esta última expresión tiene una forma equivalente, descrita en [15], que desarrolla la suma sobre todas las M clases en una partición $\tilde{P}$

$$Q = \sum_{c=1}^{M} \Big[\frac{l_c}{m} - \Big(\frac{d_c}{2m} \Big)^2 \Big] \tag{4.3}$$

expresión en donde l_c es la cantidad de aristas uniendo a los vértices de la clase con índice c, y a su vez d_c es la suma de los grados de los vértices en esta misma clase.

Con las expresiones (4.2) y (4.3), el problema de detección de comunidades se enfoca ahora en encontrar métodos eficientes para dar particiones significativas sobre un grafo, para posteriormente evaluar la modularidad de cada una de ellas y escoger a la que la maximice.

Sin embargo, dado que la modularidad se construye a partir de un modelo nulo específico, su formulación parece arbitraria [6], ya que el escoger otro modelo nulo podría proporcionar distintos resultados. Aunado a esto, en su definición se encuentra implícito el problema del límite de resolución.

Problema del límite de resolución de la modularidad

Este problema se plantea en [15], partiendo de la ecuación (4.3), al considerar que para obtener una modularidad positiva, toda clase significativa c debe cumplir necesariamente

$$\frac{l_c}{m} - \left(\frac{d_c}{2m}\right)^2 > 0 \tag{4.4}$$

Así, al desarrollar esta expresión se obtienen restricciones respecto al número máximo de aristas que pueden existir dentro de una misma comunidad. Específicamente, se formula que

$$l_c < \frac{4m}{(a+2)^2} \tag{4.5}$$

para algún entero $a \geq 0$, que representa la cantidad de aristas que conectan a un vértice v con vértices de distinta comunidad, dividida entre la cantidad de conexiones que unen a v con vértices de su misma comunidad. De esta forma, al tomar $a > 0$, se obtiene que el número de aristas dentro de la comunidad c esta limitado por alguna proporción del tamaño del grafo, cuando se puede considerar que en general esta cantidad no debería estar limitada.

Posteriormente, a partir de (4.5) con $a > 0$, se demuestra en [15], como es que dado un grafo compuesto por cliques maximales (fig. 4.2), conectados entre sí por un número escaso de aristas, la modularidad adquiere valores mayores al unir más de un clique en una sola comunidad, que detectando cada clique maximal como una comunidad por separado, lo que es inconsistente con la noción de comunidad.

Debe notarse que el problema del límite de resolución surgió al asumir que (4.4) es una condición necesaria para toda comunidad. Tomando en cuenta este análisis, basaremos nuestro método de detección de comunidades en la premisa de que no existe una única partición significativa, aspecto que se relacionará en nuestra formulación con el concepto de orden jerárquico entre comunidades.

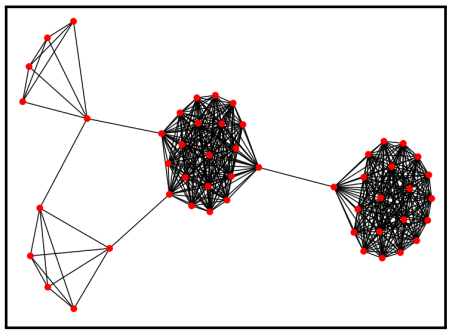

Figura 4.2: Ejemplo de un grafo compuesto por cliques maximales como plateado en [15].

Orden jerárquico de comunidades

De entre los métodos para la detección de comunidades en grafos, destacan aquellos llamados de **agrupamiento jerárquico divisivo** [6]. Estos métodos se caracterizan por dividir de manera recursiva ciertas particiones del conjunto de vértices de un grafo, considerando primero que todos los vértices son parte de una misma comunidad, y terminando cuando cada vértice representa una comunidad por sí solo.

Ejemplo de este tipo de procedimientos es el algoritmo Girvan-Newman [16], que se basa en la definición de centralidad de arista, y del cual se hablará más adelante en el marco teórico. Por el momento, cabe mencionar que dicho procedimiento se basa en la supresión recursiva de las aristas del grafo, de forma que las comunidades que se detectan con él, se encuentran conformadas por componentes conexas respecto a cada paso determinado durante el borrado de aristas.

Así, dada una comunidad representada por una componente conexa, si se llegara a remover una arista puente dentro de ella, entonces dicha comunidad sería partida en otras. Con esto finalmente se obtienen comunidades anidadas, lo que permite explorar múltiples niveles de una estructura en comunidades de un mismo grafo [6].

Dado que el problema del límite de resolución está asociado directamente con la idea de comunidades anidadas, consideramos que los métodos de agrupamiento jerárquico divisivos resultan ideales para contrarrestar o restringir el alcance de dicho problema.

Actualmente, el algoritmo Girvan-Newman representa uno de los mejores métodos para detectar comunidades por medio de la maximización de la modularidad [57], presentando una alternativa al problema de su optimización. Sin embargo, este algoritmo tiene una alta complejidad en tiempo, lo que limita su aplicación a grafos con un reducido número de vértices (≤ 100).

Se debe mencionar que el método de detección de comunidades desarrollado en el presente trabajo también cumple ser de agrupamiento jerárquico y divisivo, y también se encuentra basado en la hipótesis en torno a la centralidad de las aristas [16], por lo que se buscó plantear en este una alternativa al límite de resolución.

No obstante, debido al tiempo físico que demanda el procedimiento Girvan-Newman, únicamente compararemos nuestro método contra dicho algoritmo de forma teórica. A pesar de esto, para desarrollar una comparación práctica recurriremos al algoritmo Clauset-Newman-Moore [36], resumido también en el siguiente marco teórico.

4.3 Marco teórico

En esta sección damos primero una descripción de las propiedades de la red de interacciones neuronales de *C. Elegans*. Posteriormente se resumen tres algoritmos de detección de comunidades en grafos: el algoritmo Girvan-Newman [16], el algoritmo Clauset-Newman-Moore [36] y el método Spinglass [59].

Después se presentan los conceptos de teoría de probabilidad necesarios para nuestra formulación de la modularidad. Por último, se incluye la definición de modularidad para grafos con pesos en las aristas, que será de utilidad para desarrollar nuestro planteamiento de la modularidad.

Red de interacciones neuronales de *C. Elegans* Hermafrodita

Caenorhabditis Elegans (*C. Elegans*) es un nematodo o gusano redondo (fig. 4.3) [21] con piel transparente, y que puede presentar uno de dos sexos, hermafrodita o macho. Habita naturalmente tanto en cuerpos de agua como en algunos entornos terrestres, y se alimenta principalmente de bacterias y otros microbios. Aunado a esto, tiene un ciclo de vida de entre 2 y 3 días, por lo que representa un organismo modelo para diferentes áreas de la ciencia, como la genómica, biología del desarrollo y la neurociencia.

En este trabajo estudiamos y comparamos algunas de las comunidades que se pueden detectar en la red de interacciones neuronales que se desarrolla en el sistema nervioso del *C. Elegans* hermafrodita, organismo que ha sido ampliamente estudiado por White et. al [20].

Figura 4.3: Microscopía DIC de un espécimen de *C. Elegans* hermafrodita adulto.
(Imagen tomada y adaptada de [62])

El sistema nervioso de dicha variedad se encuentra conformado por 302 neuronas, distribuidas entre dos colecciones principales que se encuentra a su vez conectadas por medio de solo dos neuronas: un subsistema nervioso somático, o que se distribuye por el cuerpo de todo el organismo, constituido por 282 neuronas, y otro subsistema de 20 neuronas que se localiza en los músculos de la faringe del organismo. En específico, nuestro análisis se centró en la red de interacciones en el subsistema nervioso somático de *C. Elegans*, haciendo uso de la base de datos *Wormatlas* [21].

En dicha base de datos, dedicada al estudio de este nematodo, se puede hallar una lista de las 302 neuronas y su clasificación [63]. Estas se encuentran nombradas con una codificación de letras mayúsculas y números. Las primeras 2 o 3 letras del nombre se relacionan con la morfología y tipo de conexión sináptica que desarrolla cada neurona, refiriendo a 118 clases originalmente determinadas en [20]. Seguido a estas letras se puede encontrar un número que indica un índice para múltiples neuronas en una misma clase. Algunos nombres contienen también una o más letras referentes a las simetrías anatómicas del organismo. En este último caso, la clase de la neurona está dada por las primeras tres letras, mientras que las demás letras corresponden a las simetrías: izquierda (L), derecha (R), dorsal (D) y ventral (V).

En particular, el subsistema nervioso somático se encuentra constituido por 103 clases (A1) tomadas de las 118. Además, *Wormatlas* proporciona en [64], la información necesaria para particionar el conjunto total de las neuronas por su agrupamiento en 10 ganglios anatómicos distintos (A2), donde de manera breve, un ganglio es un cúmulo formado por los cuerpos de un cierto número de neuronas, que funge como recolector y redistribuidor de las señales sinápticas. Así, para esta tesis comparamos las particiones (A1) y (A2) tanto contra los agrupamientos recuperados por medio de nuestro método, como contra los detectados con el algoritmo Clauset-Newman-Moore y una partición (A3) en comunidades enumeradas en [19], de la que se hablará más en la metodología de este capítulo.

Algoritmo Girvan-Newman

El procedimiento planteado por Girvan y Newman en [16], se encuentra basado en el concepto de **centralidad de una arista**, valor definido originalmente en ese mismo trabajo. Dicha medida representa la proporción de caminos de menor distancia que pasan por una arista dada, de entre los existentes entre todas y cada una de las parejas de vértices.

De esta forma, considerando la noción de comunidad, se puede intuir que una arista que une dos vértices en distintas comunidades, puede llegar a pertenecer a múltiples caminos de menor distancia entre parejas de vértices tomados de las dos comunidades que conecta, con lo que eventualmente esta arista tendría una alta centralidad.

Este algoritmo se puede resumir en cuatro pasos[16]: (1) calcular la centralidad de toda arista en el grafo de interés, (2) remover del grafo alguna de las aristas que presenten la máxima centralidad, con lo que se obtiene un grafo modificado, (3) recalcular la centralidad para todas las aristas que se hallan visto afectadas por la remoción, y (4) repetir este procedimiento desde el paso (2) sobre el grafo modificado, hasta que se hallan removido todas las aristas. De esta manera, se toma como una comunidad a cada componente conexa del grafo modificado, obteniendo una nueva partición en comunidades cada vez que incremente la cantidad de componentes conexas.

El cálculo de la centralidad de todas las aristas en el grafo representa por sí mismo un algoritmo de orden $O(nm)$. Así, este algoritmo tiene una complejidad en tiempo de $O(nm^2)$, sin considerar la evaluación de la modularidad para cada partición.

Por otro lado, se menciona también en [16] que una estrategia evaluada durante el desarrollo del algoritmo, consistió en calcular la centralidad de las aristas una sola vez y removerlas siguiendo un orden de mayor a menor centralidad. Sin embargo, se advierte en dicho trabajo que tal estrategia no otorga resultados significativos, ya que las comunidades se pueden encontrar conectadas por más de una arista, lo que afecta la centralidad que estas pueden llegar a tener.

Algoritmo Clauset-Newman-Moore

En Clauset et.al [36] se trata ampliamente el concepto de **incremento en la modularidad**, definición en la que se basa el algoritmo Clauset-Newman-Moore (CNM). En esta tesis no profundizaremos en tal idea, ya que nuestro interés en el algoritmo CNM se enfoca principalmente en su aplicación práctica, debido a su rapidez computacional y los resultados que puede proporcionar respecto a la detección de comunidades.

El CNM es un algoritmo de **agrupamiento jerárquico aglomerativo** [6]. Esto es, contrario al procedimiento Girvan-Newman, que divide recursivamente un grafo, CNM comienza considerando que cada vértice es una comunidad por si mismo, desarrollando la construcción de agrupamientos al unir paso a paso dos o más vértices en una misma clase, hasta que todos los vértices se encuentran en la misma comunidad.

En esencia, dado una grafo donde inicialmente cada vértice es considerado como una comunidad individual, CNM calcula los incrementos que habría en la modularidad si se uniera cada par de comunidades en una sola. De todas las posibles uniones que puede realizar, este algoritmo escoge a aquellas que proporcionen el mayor incremento en la modularidad, devolviendo una nueva partición en comunidades al conformar tales uniones. Esto se repite recursivamente hasta que todos los vértices del grafo se encuentran en el mismo grupo.

La eficiencia de este método se basa principalmente en las estructuras de datos abstractas a las que recurre para evaluar y escoger todos los incrementos en la modularidad. Con esto, se señala en [36], que para grafos con orden n y tamaño m, donde $m \sim n$, y para el que se deben realizar aproximadamente $\log n$ repeticiones de unión entre comunidades, este algoritmo tiene una complejidad de $O(n \log^2 n)$, donde $\log^2 n$ es el cuadrado del logaritmo base 2 de n. Además, si n es suficientemente pequeño (≤ 20), se pueden obtener tiempos físicos de ejecución prácticamente lineales, denotados por $O(n)$, con lo que se aprecia que este algoritmo resulta ser más rápido que el procedimiento Girvan-Newman.

Algoritmo Spinglass

Dado que en este capítulo se desarrolla una formulación de la modularidad por medio de teoría de probabilidad, se describe aquí el método Spinglass [59, 60], representativo del estado del arte, considerando estudiar como trabajo a futuro las posibles relaciones entre los fundamentos de este y nuestra formulación.

El procedimiento Spinglass [59] es un método basado en un modelo físico llamado **Modelo de Potts**, utilizado en mecánica estadística para el estudio de partículas elementales en una lattice. Con este, se establece para un grafo una función llamada **hamiltoniano de Potts**. La motivación para este método reside en la idea de que la detección de comunidades en grafos se puede estudiar, por analogía, como un fenómeno de minimización de la energía de un sistema físico, de modo que al contrario de la modularidad, aquí se busca minimizar dicho hamiltoniano.

Se han dado formas más generales [60] (eq. 4.6) para el hamiltoniano de Potts originalmente planteado en [59], de modo que se ha encontrado que estas generalizan a su vez a la modularidad (eq. 4.2), pero con la diferencia de que estas se encuentran libres del problema del límite de resolución. Así, para un grafo $G = (V, E)$ no dirigido y conexo, se tiene que el **hamiltoniano generalizado de Potts** está dado por la suma sobre toda pareja ordenada de vértices

$$H = -\sum_{V\times V} (a_{uv}A_{uv} - b_{uv}(1 - A_{uv}))\delta(C_u, C_v) \tag{4.6}$$

para una colección de números reales a_{uv}, $b_{uv} \geq 0$, asignados a cada par de vértices $u, v \in V$, y donde A_{uv} y $\delta(C_u, C_v)$ están definidos como en (4.1). Se debe mencionar que aunque este método resulta tener una alta capacidad para reconocer acertadamente comunidades en grafos, también se ha mostrado que este demanda un elevado tiempo físico para detectarlos [57], por lo que tampoco ha sido completamente aceptado como método único para detectar comunidades en grafos.

Teoría de probabilidad

Basados en [17], se resumen a continuación las definiciones de teoría de probabilidad necesarias para formular nuestro método de detección de comunidades. Específicamente se presentan aquellas relacionadas con las propiedades de un espacio de probabilidad, seguidas por la de probabilidad condicional y la de variable aleatoria. Se deberá considerar para el presente trabajo, que estas se desarrollan sobre un **espacio muestral** Ω discreto, o numerable.

Definición 4.3.1. *Cualquier colección F de subconjuntos de un espacio muestral Ω, es llamada σ-**álgebra** de Ω si para F se cumplen las siguientes tres condiciones:*
i) $\Omega \in F$.
ii) Si $A \in F$, *entonces* $A^c \in F$.
iii) Si $A_1, A_2, ... \in F$, *entonces* $\bigcup_{k=1}^{\infty} A_k \in F$.

De la anterior definición se tiene que todo elemento en Ω es comúnmente llamado **evento simple**, mientras que todos los miembros de una σ-álgebra son conocidos como **eventos compuestos**.

Definición 4.3.2. *Dado un espacio muestral Ω sobre el que se ha definido una σ-álgebra F, y dada una función $P : F \to \mathbb{R}$, se dice que la terna (Ω, F, P) es un **espacio de probabilidad**, si para P se cumplen a la vez:*
i) $P(A) \geq 0, \forall A \in F$.
ii) $P(\Omega) = 1$.
iii) $P(\bigcup_{k=1}^{\infty} A_k) = \sum_{k=1}^{\infty} P(A_k)$, *para toda sucesión* $A_1, A_2, ...$, *donde* $A_i \cap A_j = \varnothing$, *siempre que* $i \neq j$.

Al cumplirse tales condiciones, P es llamada **medida de probabilidad**, y la imagen de un evento compuesto A bajo P, es decir $P(A)$, es llamada **probabilidad** de A, y en particular se tendrá que $P(\{\varnothing\}) = 0$.

Definición 4.3.3. *Sean A y B dos eventos compuestos en un espacio de probabilidad. Si $P(B) \neq 0$, se define como **probabilidad condicional** de A, dado B, al valor*

$$P(A|B) := \frac{P(A\cap B)}{P(B)}$$

Más aún, si se cumple $P(A \cap B) = P(A)P(B)$, o equivalentemente $P(A|B) = P(A)$ o $P(B|A) = P(B)$, se dice que A y B son **eventos independientes entre sí**.

Definición 4.3.4. *Se dice que una función $X : \Omega \to \mathbb{R}$ en un espacio de probabilidad (Ω, F, P), es una* ***variable aleatoria discreta*** *si para cualquier $x \in \mathbb{R}$ se cumple que*

$$\{\omega \in \Omega | X(\omega) \leq x\} \in F$$

Modularidad en grafos con pesos no negativos

Para dar las definiciones en las que se basa nuestro método de detección de comunidades, debemos primero introducir el concepto de grafo con pesos. Esto nos permitirá reproducir inicialmente una formulación general de la modularidad [27], descrita más adelante, que posteriormente restringiremos al tomar en cuenta una hipótesis basada en la definición de centralidad de arista.

Dado un grafo no dirigido $G = (V, E)$, se dice que este tiene **pesos no negativos en las aristas** cuando para G se define una función $W : E \to \mathbb{R}_{\geq 0}$. Es posible encontrar grafos con pesos negativos, sin embargo, las definiciones relativas a la modularidad en torno a ellos salen del alcance de esta tesis, por lo que nos restringiremos a analizar grafos cuyas aristas tienen pesos positivos o iguales a cero. En general, se denotará por w_{uv} al peso de la arista $\{u, v\} \in E$, donde abusando de la notación, es común denotar $w_{uv} = 0$ si $\{u, v\} \notin E$.

Buscando estudiar las implicaciones de utilizar pesos empíricos en la detección de comunidades sobre grafos que modelan sistemas reales, Newman presenta un análisis en [27], donde amplía su definición original de modularidad (eq. 4.2) [14] a grafos con pesos no negativos, experimentales o derivados de las propiedades matemáticas del grafo. En dicho trabajo, se determina que la **modularidad** para un grafo de este tipo se puede escribir como

$$Q = \frac{1}{2W} \sum_{V \times V} (w_{uv} - \frac{W_u W_v}{2W}) \delta(C_u, C_v) \tag{4.7}$$

expresión en la que W representa la suma de los pesos de todas las aristas del grafo, W_u y W_v son las sumas de los pesos de todas las aristas incidentes a los vértices u y v respectivamente, w_{uv} es el peso de la arista que une a los vértices u y v, en donde, de no existir tal arista entonces se toma $w_{uv} = 0$, y $\delta(C_u, C_v)$ esta definido de la forma usual.

4.4 Metodología

Primeramente se habla sobre las hipótesis en las que se basa nuestro método y nuestra definición de modularidad por medio de probabilidad condicional. Se debe mencionar que en la revisión bibliográfica no se encontró referencia a una formulación semejante. Por último se describe nuestro algoritmo y la manera en que este fue evaluado.

Resumen y justificación de las hipótesis en las que se basa nuestro método y nuestra formulación

Nuestra primera hipótesis, a la que haremos referencia como (H1), consiste en considerar que dada una partición de vértices en comunidades, debe existir por lo menos un camino entre toda pareja de vértices en la misma comunidad, es decir que dos vértices no podrán pertenecer a la misma comunidad si no pertenecen a la misma componente conexa. Basados en esto, nuestro método quedará definido únicamente para grafos no dirigidos conexos.

Para formular la segunda hipótesis (H2), asumiremos en consideración al límite de resolución de la modularidad [15], que para todo grafo no dirigido y conexo, debe de existir más de una partición de los vértices en comunidades, aspecto que nuestra formulación debe permitir explorar.

Basados en la hipótesis que sustenta al método Girvan-Newman, también consideramos la centralidad de arista dentro de nuestra formulación, suponiendo que (H3) aquellas aristas con mayor centralidad *tienden* a conectar vértices en distintas comunidades. A diferencia de dicho método, exploraremos esta hipótesis por medio de la teoría de probabilidad.

Para concluir con el resumen de nuestras hipótesis, recurriremos a aquella que dió origen a la medida de modularidad (4.2), asumiendo que (H4) el número de aristas que conectan vértices dentro de la misma comunidad *tiende* a ser mayor, tanto al número de aristas que conectan vértices en distinta comunidad, como al número de aristas que conectarían vértices en la misma comunidad si el grafo en cuestión hubiera sido generado aleatoriamente preservando su secuencia de grado. Esto nos permitirá, debido a su relación con el mencionado modelo de configuraciones [6], explorar valores que se definen como sumas sobre el producto cartesiano de los vértices de un grafo.

Debe remarcarse que en las hipótesis (H3) y (H4) se hace referencia a la *tendencia* de las características que se deben cumplir en torno a una comunidad. Esto se resume así en consideración a la hipótesis (H2), de modo que se respete la posibilidad de obtener más de una partición en comuindades.

Formulación de la modularidad por medio de probabilidad condicional

Comenzaremos por establecer una definición que nos permitirá hablar de los pesos de un grafo no dirigido, pero asociando estos a parejas ordenadas de vértices (H4). Seguido a esto damos definiciones relacionadas con teoría de la probabilidad que nos permitirán recuperar la forma de la modularidad para grafos con pesos no negativos. Se debe considerar en todo momento que estas definiciones aplican a grafos no dirigidos conexos (H1), a menos que se especifique lo contrario explícitamente.

Definición 4.4.1. *Sea $G = (V, E)$ un grafo no dirigido, y sea $W : E \to \mathbb{R}_{\geq 0}$ una función de pesos no negativos en las aristas de G, para la que se denota con w_{uv} al peso de la arista $\{u, v\} \in E$. Definimos como* ***función de exclusividades en*** *G* ***basadas en*** *W, al mapeo $\chi : V \times V \to \mathbb{R}_{\geq 0}$ dado por*

$$\chi(u, v) = \begin{cases} w_{uv} & si\ \{u, v\} \in E \\ 0 & si\ \{u, v\} \notin E \end{cases}$$

De esta forma, se tiene que $\chi(u, v) = \chi(v, u)$, y dado que implícito en nuestra definición de grafo no dirigido se tiene que estos no pueden tener lazos, entonces se tendrá $\chi(v, v) = 0$, para todo vértice v.

Por otro lado, se debe señalar que de aquí en adelante haremos uso de la notación χ_A para denotar a la suma de las exclusividades sobre cualquier colección A de parejas ordenadas de vértices.

Definición 4.4.2. *Sea $G = (V, E)$ un grafo no dirigido y conexo, con conjunto de vértices y aristas no vacíos, y para el que se ha determinado alguna función de exclusividades χ, donde existe por lo menos una pareja de vértices $(a, b) \in V \times V$ tal que $\chi(a, b) \neq 0$, se define como* ***espacio de exclusividades en*** *G* ***respecto a*** *χ, al espacio de probabilidad*

$$\mathbb{P}_{(G;\chi)} \ := \ (V \times V, \ \ \wp[V \times V], \ \ P)$$

para el que $\wp[V \times V]$ denota al conjunto potencia de todas las parejas ordenadas de vértices de G, y P es una función de probabilidad tal que

$$P(\{(u, v)\}) = \frac{\chi(u, v)}{\chi_{V \times V}} \qquad \forall (u, v) \in V \times V, \qquad \textit{donde} \qquad \chi_{V \times V} = \sum_{(x,y) \in V \times V} \chi(x, y)$$

Dado que $\wp[V \times V]$ contiene a todos los posibles subconjuntos de $V \times V$, se puede verificar que $\wp[V \times V]$ es una σ-álgebra para $V \times V$, además, ya que se considera por lo menos una pareja ordenada con exclusividad distinta de 0, se garantiza que la función P cumple con los axiomas de una medida de probabilidad sobre $V \times V$. Más aún, a partir de que $\chi(u, v) = \chi(v, u)$, se tiene que $P(\{(u, v)\}) = P(\{(v, u)\})$.

Con el espacio de exclusividades $\mathbb{P}_{(G;\chi)}$ de un grafo G, buscamos representar la probabilidad que tiene una arista de unir a dos vértices en una misma comunidad, dependiendo de la exclusividad χ que se asocie a ella, dado que suponemos que no existe una única partición en comunidades (H2).

Más abajo relacionaremos lo anterior con la hipótesis (H3), pero por el momento se estudiarán algunas otras implicaciones de (H4) por medio de la definición de relaciones que tiene un vértice dado.

Definición 4.4.3. *Sea $G = (V, E)$ un grafo no dirigido, para todo vértice $v \in V$ definimos como **colección de relaciones** de v en G, al conjunto*

$$R_v = \{(x, y) \in V \times V \quad | \quad \{x, y\} \quad \textit{es incidente a} \quad v \quad \textit{en } G\}$$

Observación 4.4.1. *Dado un grafo $G = (V, E)$ y dos vértices $u, v \in V$, se tiene que $\{u, v\} \in E$ si, y solo si*

$$R_v \cap R_u = \{(u, v), (v, u)\}$$

Demostración. Dada la definición del conjunto de relaciones de un vértice, se aprecia que la intersección de dos de estos está formada por las permutaciones de las aristas que son incidentes a ambos vértices, de modo que la única arista que puede ser incidente a dos vértices distintos al mismo tiempo, será la arista que los une, asumiendo que esta existe (ver fig. 4.4).

■

De forma similar, se puede mostrar también que si dos vértices no se encuentran unidos por una arista, entonces la intersección de sus conjuntos de relaciones es vacío. Además, se debe observar que los conjuntos de relaciones y todas las operaciones que entre estos se pueden realizar, pueden ser considerados como eventos compuestos dentro de un espacio de exclusividades de un grafo.

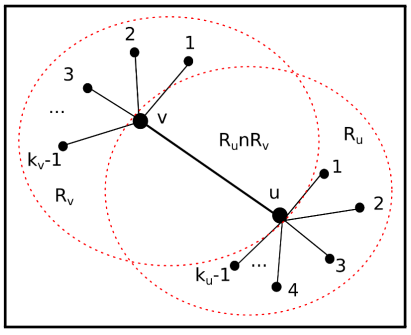

Figura 4.4: Intersección de los conjuntos de relaciones de dos vértices.

Considérese ahora que dos eventos compuestos A y B son independientes cuando $P(A|B) = P(A)$, o de otra forma $P(A \cap B) = P(A)P(B)$, de donde $P(A \cap B) - P(A)P(B) = 0$. Motivados por esto, y tomando en cuenta que dados dos vértices u, v conectados por una arista en un grafo, la probabilidad asociada a $R_u \cap R_v$ es el doble de probabilidad que tendrían cada una de las parejas (u, v) o (v, u), damos la siguiente definición buscando evaluar la influencia de un conjunto de relaciones sobre otro.

Definición 4.4.4. *Sea $G = (V, E)$ un grafo para el que sa ha definido un espacio de exclusividades. Definimos como **dependencia** entre las relaciones de dos vértices en G, a la variable aleatoria $D : V \times V \to \mathbb{R}$, dada por*

$$D(u, v) = \frac{P(R_u \cap R_v)}{2} - \frac{P(R_u)P(R_v)}{4}$$

Es importante señalar que para cualquier número real x, toda colección de parejas que presente una dependencia menor o igual a x estará en la σ-álgebra de un espacio de exclusividades, ya que esta es el conjunto potencia de las parejas ordenadas de vértices. Aunado a esto se tiene nuevamente que $D(u, v) = D(v, u)$, con lo que se puede intuir que cada arista en un grafo tiene asociado un valor de dependencia, lo que nos permitirá, considerando la siguiente observación, ordenarlas y borrarlas iterativamente del grafo, comenzando con aquellas de menor dependencia hasta las de mayor dependencia.

Observación 4.4.2. *Sea $G = (V, E)$ un grafo para el que se ha definido un espacio de exclusividades, y dados dos vértices u y v en G, para los que $P(R_u) > 0$ y $P(R_v) > 0$, sin pérdida de generalidad se tiene que*

$$D(u, v) > 0 \iff P(R_u|R_v) > \frac{P(R_u)}{2}$$

Demostración. Estableciendo $P(R_u) > 0$ y $P(R_v) > 0$, es posible obtener la probabilidad condicional en la pasada proposición. Con esto, la prueba en ambos sentidos se sigue inmediatamente de la definición de la dependencia.

■

De la anterior proposición se puede verificar que valores negativos de dependencia darán probabilidades condicionales menores a las de cada evento por separado, mientras que dependencias iguales a 0 hablarán de conjuntos independientes de relaciones entre vértices. Así, dada una partición de los vértices de un grafo, buscaremos evaluar la suma de las dependencias sobre todas las parejas ordenadas de vértices en un mismo grupo en dicha partición, de modo que al maximizar esta medida, se busque optimizar la probabilidad condicional asociada a los pares de conjuntos de relaciones en cada grupo.

Definición 4.4.5. *Dado un grafo $G = (V, E)$, no dirigido y conexo, para el que se ha definido un espacio de exclusividades, y dada una partición $\tilde{P}$ de los vértices de G. Se define como **modularidad** Q de G respecto a $\tilde{P}$, a la suma de las dependencias asociadas a cada pareja ordenada de vértices, tomados de los grupos de $\tilde{P}$, en símbolos*

$$Q = \sum_{(u,v)\in V\times V} [D(u, v)]\delta(C_u, C_v)$$

donde C_u y C_v son los grupos a los que pertencen dos vértices u y v respectivamente, y $\delta(C_u, C_v)$ es una función que vale 1 si $C_u = C_v$, y 0 en otro caso.

Para mostrar cómo nuestra definición de modularidad recupera la modularidad para grafos no dirigidos y conexos (eq.4.7), con pesos no negativos en las aristas, se debe considerar que al existir una arista entre dos vértice $u, v \in V$, la probabilidad $P(R_u \cap R_v)$ será el doble de la probabilidad de cualquiera de las parejas ordenadas (u, v) o (v, u) (obs. 4.4.1). Además, si u y v no están conectados por una arista, esta probabilidad será 0 ya que $R_u \cap R_v = \varnothing$, a la vez que $P(\{(u, v)\} = P(\{(v, u)\}) = 0$. Con esto, y abusando de la notación, podemos escribir en general

$$P(R_u \cap R_v) = 2P(\{(u, v)\}) = \frac{2\chi(u, v)}{\chi_{V\times V}} \tag{4.8}$$

$$P(R_u) = \frac{\chi_{R_u}}{\chi_{V\times V}} \tag{4.9}$$

Retomando $w_{uv} = 0$ siempre que $\{u, v\} \notin E$, se puede escribir para cualquier pareja de vértices $\chi(u, v) = w_{uv}$. Por otro lado, $\chi_{V\times V}$ es el doble de la suma de los pesos en un grafo, es decir $\chi_{V\times V} = 2W$, mientras que χ_{R_u} es el doble de la suma de los pesos de todas las aristas incidentes al vértice u, o sea $\chi_{R_u} = 2W_u$. Con esto, las ecuaciones (4.8) y (4.9), se pueden reescribir cada una como

$$P(R_u \cap R_v) = \frac{w_{uv}}{W} \tag{4.10}$$

$$P(R_u) = \frac{W_u}{W} \tag{4.11}$$

Así, la dependencia (def. 4.4.4) de una pareja ordenada de vértices cualesquiera, se puede escribir como sigue

$$D(u, v) = \frac{P(R_u \cap R_v)}{2} - \frac{P(R_u)P(R_v)}{4} = \frac{w_{uv}}{2W} - \frac{W_u W_v}{4W^2} \tag{4.12}$$

De esta forma, la modularidad, como definida en 4.4.5, tiene su forma equivalente en

$$Q = \frac{1}{2W} \sum_{(u,v)\in V\times V} [w_{uv} - \frac{W_u W_v}{2W}]\delta(C_u, C_v) \tag{4.13}$$

que es la misma expresión para la modularidad en grafos pesados (4.7). Más aún, cuando se define el espacio de exclusividades respecto a una función de pesos no negativos sobre las aristas, tal que todas las aristas tengan el mismo peso o la misma exclusividad, por simplicidad supóngase $w_{uv} = A_{uv}$ que es 1 si $\{u, v\} \in E$ y 0 en otro caso, se tiene (ver fig. 4.4)

$$P(R_u \cap R_v) = \frac{A_{uv}}{m} \tag{4.14}$$

$$P(R_u) = \frac{k_u}{m} \tag{4.15}$$

Con lo que finalmente se recupera la modularidad originalmente planteada en [14],

$$Q = \frac{1}{2m} \sum_{(u,v)\in V\times V} [A_{uv} - \frac{k_u k_v}{2m}]\delta(C_u, C_v) \tag{4.16}$$

Con esto se muestra que la modularidad se encuentra ligada con la definición de probabilidad condicional, inclusive en su forma general para grafos con pesos no negativos en las aristas. Sin embargo, aunado al hecho de haber asumido (H4), equivalente a asumir el modelo de configuraciones en (4.1), ahora se tiene el problema de definir la exclusividad que tiene asociada cada pareja de vértices. Para esto, retomaremos (H3), que bajo nuestra formulación, queda interpretada como que aquellas aristas con mayor centralidad, tienen menor probabilidad de ser aristas uniendo vértices en una misma comunidad.

Definición 4.4.6. *Sea $G = (V, E)$ un grafo no dirigido y conexo, para el que se han calculado todos los valores de centralidad en sus aristas, denotados por c_{uv} para toda arista $\{u, v\} \in E$, y denotando por z al máximo de estos valores. Se define como* ***exclusividad basada en la centralidad****, a la función*

$$\chi_c(u, v) = \begin{cases} w_{uv} = z - c_{uv} & \textit{si } \{u, v\} \in E, \textit{y almenos una arista tiene centralidad distinta a la de las demás} \\ w_{uv} = 1 & \textit{si } \{u, v\} \in E, \textit{y todas las aristas tienen la misma centralidad} \\ w_{uv} = 0 & \textit{si } \{u, v\} \notin E \end{cases}$$

La anterior definición, da sustento a la hipótesis (H3) en nuestra formulación, pero además, da razón también de la hipótesis (H1), ya que de ser G un grafo disconexo, la centralidad máxima de una componente conexa podría dar resultados no significativos sobre las comunidades de otra componente conexa.

Además, en consideración del límite de resolución de la modularidad, donde se muestra que la máxima cantidad de aristas que puede haber en una comunidad se encuentra restringido por el tamaño del grafo, se obtiene, a partir de la definición de exclusividad basada en la centralidad, una propiedad que favorece que la dependencia entre una pareja de vértices se vea afectada según el vecindario de estos.

Observación 4.4.3. *Sea $G = (V, E)$ un grafo para el que se ha definido un espacio de exclusividades basadas en la centralidad, y donde existe al menos una arista con centralidad diferente de la de las demás aristas, dados dos vértices u y v para los que $P(R_u) > 0$ y $P(R_v) > 0$, se tiene*

$$D(u, v) > 0 \iff z - \frac{W_u W_v}{2W} > c_{uv}$$

donde z y c_{uv} están definidos como en (4.4.6), mientras que W_u, W_v y W son como en las ecuaciones (4.10) y (4.11).

Demostración. Nuevamente, estableciendo $P(R_u) > 0$ y $P(R_v) > 0$ y de la definición de la dependencia junto con la de exclusividad basada en la centralidad, es posible dar directamente esta demostración en ambos sentidos.

■

Lo anterior indica que la centralidad de una arista debe ser menor a cierto margen para representar un incremento en la probabilidad condicional entre las relaciones de los vértices a los que une. Más importante aún es que dicho margen depende del vecindario de estos vértices, siendo que mientras más centrales sean las aristas que comparten a un mismo vértice, más difícil será para algunas de ellas ser conexiones entre vértices de una misma comunidad, lo que esperamos sea una restricción sobre el límite de resolución de la modularidad.

Por último, se presentan todas las definiciones que nos permitirán formular directamente, en el siguiente apartado, nuestro método de detección de comunidades en grafos y su relación con (H2).

Definición 4.4.7. *Dado un grafo $G = (V, E)$, para el que se ha definido un espacio de exclusividades, y sea d un valor en el rango de la dependencia de las parejas ordenadas en $V \times V$, definimos como d-**aristas** al conjunto*

$$\eta_d = \{\{u, v\} \in E \mid D(u, v) = D(v, u) \geq d\}$$

Definición 4.4.8. *Sea $G = (V, E)$ un grafo, para el que se ha definido un espacio de exclusividades, y sea d un valor en el rango de la dependencia de las parejas de $V \times V$, definimos como **imagen** d-**modular** de G al grafo $G_d = (V, \eta_d)$, y a toda componente conexa de G_d la llamamos d-comunidad de G.*

Definición 4.4.9. *Para un grafo $G = (V, E)$ que tiene definido un espacio de exclusividades, y para un valor d en el rango de la dependencia de las parejas de $V \times V$, definimos como d-**modularidad**, a la modularidad Q_d que tiene el grafo G respecto a la partición de V en sus d-comunidades, como definida en (4.4.5).*

Definición 4.4.10. *Sea $G = (V, E)$ un grafo para el que se ha definido un espacio de exclusividades, y sea d_0 un valor en el rango de la dependencia de las parejas de $V \times V$, se define como **comunidades de modularidad máxima** a todas las d_0-comunidades de G, si se cumple que i) la modularidad Q_{d_0} es igual a la máxima de las d-modularidades evaluadas respecto a todo posible valor d de la dependencia, al mismo tiempo que ii) el número de d_0-comunidades es el mayor de entre todos los valores d para los que $Q_d = Q_{d_0}$, y iii) si d_0 es el mayor valor de entre todos los posibles d que cumplen i) y ii).*

Método propio: Optimización de la Modularidad por Comparación de la Dependencia (MODC)

De la observación (4.4.2), se intuye que aquellas parejas de vértices con mayor dependencia asociada, representan aristas con mayor probabilidad condicional para unir dos vértices dentro de una misma comunidad. De forma opuesta, aquellas que presentan una menor dependencia, e incluso dependencia negativa o nula, tendrán menor probabilidad de ser parejas de vértices en una misma comunidad, por lo que, de existir inicialmente en el grafo, la remoción gradual de las aristas que estas últimas representan puede mostrar diferentes agrupamientos de los vértices en comunidades.

Con lo anterior, proponemos ordenar las aristas según la dependencia que sus permutaciones tengan asociadas, para posteriormente removerlas conforme tal valor, comenzando por las de menor dependencia hasta haber eliminado todas las aristas del grafo. Con esta operación gradual de borrado, buscaremos detectar aquella partición en componentes conexas de los vértices que maximice la modularidad, definida como la suma de las dependencias dentro de cada componente.

A continuación presentamos los pasos a seguir en el procedimiento de detección de comunidades en grafos, que llamamos **Optimización de la Modularidad por Comparación de la Dependencia** (o **MODC**, por sus siglas en inglés, Modularity Optimization by Dependence Comparison). Este queda definido para grafos no dirigidos y conexos, y con una evaluación de la dependencia hecha por medio de exclusividades basadas en la centralidad de arista. En la figura (4.5) se presenta también un diagrama que lo resume.

Algoritmo 4.4.1 Modularity Optimization by Dependence Comparison (MODC)

Input: un grafo conexo y no dirigido $G = (V, E)$.
Output: la partición de V en comunidades de modularidad máxima.
(1) Dado $G = (V, E)$, no dirigido y conexo, evaluar la centralidad para toda arista.
(2) Para toda pareja de vértices $u, v \in V$, determinar la exclusividad basada en la centralidad $\chi_c(u, v)$.
(3) Determinar la dependencia de toda pareja de vértices por medio de (4.12).
(4) Crear un conjunto $\mathbb{D}$ de los valores de dependencia $D(u, v) = D(v, u)$ tomados de cada arista correspondiente $\{u, v\} \in E$.
(5) Seleccionar el menor valor $d \in \mathbb{D}$, y determinar a todas las d-aristas.
(6) Con las d-aristas crear G_d, y evaluar Q_d de G respecto a la partición de V en d-comunidades.
(7) Guardar el valor d y su correspondiente Q_d, borrando luego d del conjunto $\mathbb{D}$.
(8) Repetir este procedimiento desde el paso (5), hasta que G_d tenga conjunto de aristas vacío.
(9) De entre todos los valores Q_d resultantes, tomar al máximo, y reportar de G_d, las correspondientes d-comunidades como comunidades de modularidad máxima en G.

Discusión del tiempo de ejecución de MODC

Para estudiar el tiempo de ejecución del procedimiento MODC, consideramos un grafo conexo y no dirigido de orden n y tamaño m. Por medio del algoritmo Girvan-Newman, sabemos que el cálculo de la centralidad de aristas, que es también el paso (1) de MODC, es una subrutina de orden $O(nm)$, mientras que los pasos (2) y (3) de MODC se pueden efectuar en $O(n^2)$ operaciones, incluso si se recurre a evaluar los valores de exclusividad y dependencia por medio de las combinaciones con repetición de los vértices, en vez de su producto cartesiano.

Por otro lado, determinar los valores de dependencia asociados a las aristas (4) y escoger el mínimo entre estos (5), son pasos que se puede efectuar al recorrer el conjunto de aristas en su totalidad, es decir $O(m)$, que se ve ya superado por la obtención de las centralidades. Por último, para el paso (6), consideramos que un peor caso para MODC, o caso que conlleva el mayor número de operaciones, sería recibir un grafo tal que cada arista tenga un valor de dependencia distinto, es decir, máximo m valores de dependencia. Asumiendo que tal grafo existe, cada iteración de los pasos (5) a (8) conllevaría remover una única arista del grafo, pero calculando la modularidad por cada una de estas remociones.

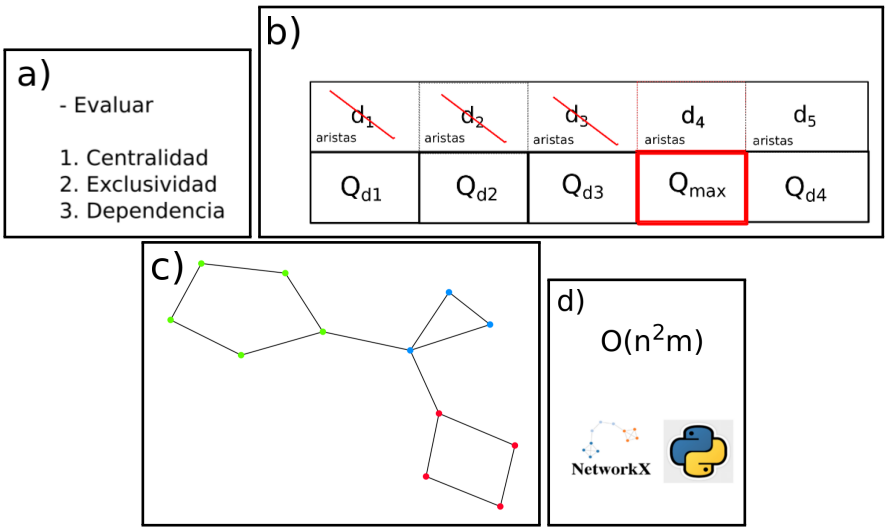

Figura 4.5: Resumen de los pasos de MODC: a) determinar centralidad, exclusividad y dependencia asociada a cada arista, b) iterar sobre los valores de dependencia de las aristas, obteniendo las d-comunidades, y c) detectar la partición de modularidad máxima. Estimamos que este algoritmo tiene una complejidad en tiempo de $O(n^2m)$, y llevamos a cabo su implementación computacional en lenguaje Python, haciendo particular uso de la biblioteca NetworkX [30].

Dado que la evaluación de la modularidad se puede considerar como un proceso de $O(n^2)$, entonces el paso (6) para el peor escenario comprende $O(n^2m)$ acciones. De esta forma, estimamos que el algoritmo MODC tiene una complejidad en tiempo de $O(n^2m)$, derivada del paso (6), ya que por inspección esta resulta ser mayor al tiempo de ejecución de las demás subrutinas.

Para desarrollar una comparación con el algoritmo Girvan-Newman, que tiene un tiempo de ejecución $O(nm^2)$ sin calcular la modularidad para cada posible partición, podemos pensar en un grafo para el que se cumpla $m > n$. Inmediatamente se aprecia que $nm^2 > n^2m$, lo que muestra que el algoritmo MODC posee un menor tiempo de ejecución que el algoritmo Girvan-Newman. Más aún, si se supone que $n \sim m$, ambos algoritmos conllevarían $O(n^3)$ operaciones. Sin embargo, el procedimiento completo para detectar las comunidades de mayor modularidad con el método Girvan-Newman alcanza una complejidad de $O(n^5)$, con lo que se verifica que MODC presenta una ventaja en tiempo de ejecución respecto a dicho procedimiento.

Planteamiento de alternativa al límite de resolución de la modualridad

En la definición de d-comunidades, se aprecia que estas nos permiten explorar diversas particiones de un grafo. Más aún, estas particiones se encuentran fundamentadas en la relación de la modularidad con la probabilidad condicional, dada por la dependencia de las parejas de vértices. De esta forma, aquellas particiones con una modularidad positiva o cercana a la mayor, pueden también resultar significativas respecto a la noción de comunidad, fundamentando su relevancia por medio de la probabilidad asociada a ellas. Así, en el caso de que las comunidades de modularidad máxima se vieran afectadas por el límite de resolución, se puede recurrir al estudio de los valores de dependencia que producen modularidades semejantes al máximo.

En este trabajo no estudiamos a detalle las implicaciones de esta alternativa. No obstante, se aprecia que nuestra formulación de la modularidad respeta (H2), ofreciendo razones probabilísticas para el estudio de particiones en comunidades con modularidad menor a la máxima. Entre el análisis práctico desarrollado en esta tesis, planteado a continuación, se describen algunos experimentos computacionales que dan seguimiento al estudio del límite de resolución y sus efectos sobre nuestro método.

Evaluación de MODC

Para evaluar la funcionalidad del procedimiento MODC, primero se utilizó este con grafos pequeños (orden y tamaño ∼20). Posteriormente, se analizaron las comunidades de modularidad máxima detectadas con este método sobre la red conocida como el Club de Karate de Zachary [31, 16], ampliamente utilizada en esta área. Luego se estudió su aplicación sobre grafos aleatorios de prueba y sobre la red neuronal de *C. Elegans*, contrastando los resultados contra los obtenidos con el algoritmo CNM por medio de la medida llamada *información mutua normalizada* [29]. Finalmente, se analizaron dos grafos propuestos en [15] para evaluar los efectos del límite de resolución de la modularidad.

Implementación computacional de MODC y su aplicación a grafos pequeños

Se hizo una implementación de MODC en lenguaje Python usando las bibliotecas: NetworkX [30], Matplotlib [38], Numpy [39] y MPmath [40]. De entre estas, se debe remarcar el uso de la biblioteca MPmath, cuyo propósito es el manejo computacional de alta precisión de números decimales grandes. Esto es importante ya que en este método se opera esencialmente con probabilidades, y es sabido que en general, las operaciones computacionales sobre números decimales puede llagar a proporcionar resultados erróneos o desviados del correcto. Por otro lado, para revisar inicialmente la funcionalidad de MODC, se aplicó este a algunos grafos pequeños (orden y tamaño ∼20). En la figura 4.6, se muestran cuatro de estos grafos, que llamamos: *Escoba-Doble*, *5,7-Paleta*, *Tres-Ciclos* y *Cuadrados-Completos*.

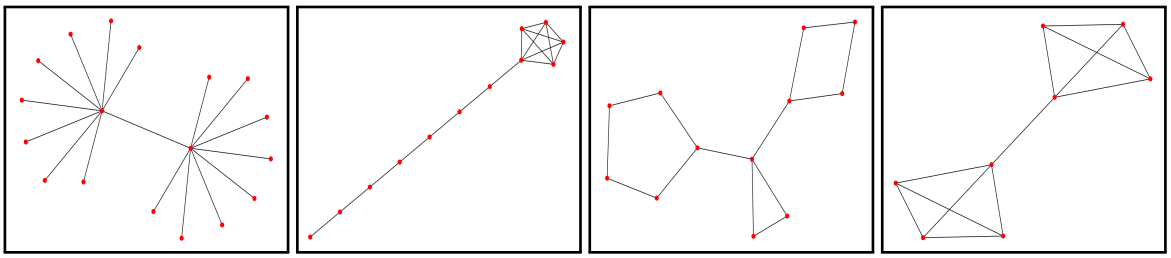

Figura 4.6: Grafos pequeños, de izquierda a derecha: Escoba-Doble, 5,7-Paleta, Tres-Ciclos y Cuadrados-Completos.

Análisis de la red del Club de Karate de Zachary

La red de amistades conocida como el Club de Karate de Zachary (fig. 4.7), es un grafo no dirigido y conexo que representa las relaciones de amistad entre participantes de un club de karate, que formó parte de un estudio desarrollado por Zachary y Wayne [31] en 1977. Posteriormente esta fue analizada como parte del planteamiento del algoritmo Girvan-Newman [16], y desde entonces ha sido ampliamente utilizada para evaluar algoritmos de detección de comunidades en grafos [6]. Así, buscamos contrastar las comunidades de moduralidad máxima detectadas con nuestro método, contra la información que se posee sobre esta red [31, 16].

Análisis de grafos de prueba I: Planted-Models

Para incrementar la certeza en la utilidad de un método de detección de comunidades en grafos, es necesario estudiar su aplicación en múltiples ejemplares generados aleatoriamente. Como se mencionó en el capítulo 3 de esta tesis, la producción de grafos aleatorios es un área de estudio por sí sola, compuesta por múltiples métodos y modelos que proporcionan grafos con propiedades distintas. En nuestro caso, hemos recurrido al análisis de dos modelos, los llamados l-Planted-model [6] y los modelos-LFR [29].

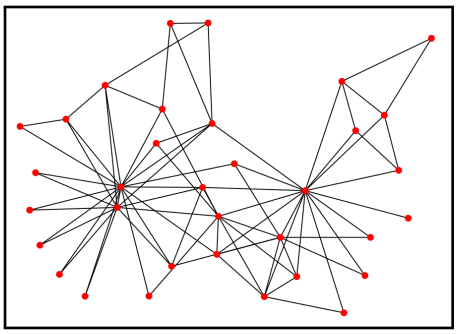

Figura 4.7: Red del Club de Karate de Zachary.

El primero de ellos hace referencia a los grafos de prueba utilizados también en el trabajo de Girvan y Newman [16]. Estos son grafos compuestos por l comunidades cada una conformada por g vértices, donde los vértices en un mismo grupo son conectados de manera aleatoria con una probabilidad p_{in}, mientras que los vértices en distintas comunidades son unidos con probabilidad p_{out}. Ambas probabilidades se encuentran relacionadas directamente con el grado promedio $< k >= z_{in} + z_{out}$ de los vértices, donde se tiene la cantidad de conexiones promedio que unen a cada vértice con otros de su misma comunidad $z_{in} = p_{in}(g-1)$ y de distinta comunidad $z_{out} = p_{out}g(l-1)$. De este modo, uno puede hacer variar dichas probabilidades de tal forma que se obtengan valores deseados para tales grados promedio.

En este trabajo seguimos lo planteado en [16, 6]. Así, definimos las probabilidades de tal manera que el grado promedio de los vértices sea $< k >= 16$, donde $z_{in} = 16 - z_{out}$ para $z_{out} = 1, 2, 3, ..., 15$. Todo esto para $l = 4$, es decir cuatro comunidades conformadas cada una por $g = 32$ vértices, con lo que se tiene un orden de $n = 128$ y específicamente $p_{in} = z_{in}/31$ mientras que $p_{out} = z_{out}/96$. Generamos 100 grafos aleatorios, por medio de la biblioteca NetworkX, por cada pareja de z_{in} y z_{out}, obteniendo valores promedio para las propiedades que estos presentan. Con estos valores, comúnmente se espera poder detectar cerca de 4 comunidades para los casos con z_{out} en el intervalo $[1, 8]$, dado que con $z_{out} > z_{in}$ se tendría un mayor número de conexiones por vértice con vértices de distinta comunidad que con los de su misma comunidad.

Análisis de grafos de prueba II: Modelos-LFR

Implícito en la formulación de los planted-models, se asume que toda comunidad se encuentra conformada por la misma cantidad de vértices. Sin embargo, en redes que modelan sistemas reales, se ha visto que esta suposición no es del todo correcta [6], ya que pueden existir comunidades con cantidades de vértices anómalas respecto a su promedio, tanto inferior como superiormente.

Por esta razón, se propone en [6, 29], otro modelo de detección de comunidades en grafos, llamado modelo-LFR, donde tanto el grado promedio de los vértices, así como la cantidad de aristas en cada comunidad, siguen leyes de potencias, para las que se proporcionan exponentes ek y ec respectivamente. Este modelo funciona por medio de un parámetro de mezcla μ, que indica la proporción de aristas que existen entre vértices de distintas comunidades, conectando vértices al considerar también su grado promedio $< k >$ y el mínimo número de vértices r que puede existir en una misma comunidad.

En nuestro análisis generamos 100 modelos-LFR por cada $\mu = 0.1, 0.15, 0.2, 0.25, ...0.55, 0.6$, promediando también las propiedades asociadas a estos. Para crear estos grafos se recurrió nuevamente a la biblioteca NetworkX, donde en [65] se menciona que este modelo, a pesar de ser descrito como robusto por sus autores [29], en la práctica genera grafos correctamente para un conjunto reducido de parámetros, aspecto que confirmamos al utilizarlo. Por esta razón, basados en un ejemplo funcional propuesto en el mismo sitio de NetworkX, generamos nuestros grafos con un orden de $n = 500$, $ek = 3$, $ec = 2$, grado promedio $< k >= 10$, y con un mínimo número de vértices por comunidad de $r = 5$.

Para la producción y análisis de todos estos grafos aleatorios, se recurrió a la implementación de la programación en paralelo, por medio de la biblioteca Multiprocessing [66] ya incluida también en el lenguaje Python, lo que nos permitió reducir el tiempo físico que este trabajo conlleva. La pronta ejecución y terminación de estos programas se hizo posible gracias a un servidor provisto por el Laboratorio de Visualización Científica (LAVIS) [41], del Instituto de Neurobiología de la UNAM, campus Juriquilla.

Análisis sobre *C. Elegans*

La red de interacciones neuronales de *C. Elegans* es comúnmente modelada como un grafo dirigido con etiquetas, lazos y múltiples flechas [20]. No obstante, basándonos en el trabajo presentado en [19], desarrollamos un análisis de detección de comunidades sobre un grafo no dirigido simple y conexo, que es una representación del sistema nervioso somático del nematodo. Este está conformado por 2287 aristas, y 279 neuronas de las 282 originales de este sistema, donde se excluyen aquellas llamadas VC06, CANL y CANR, ya que estas últimas no presentan conexiones evidentes con las demás. Tal grafo puede ser reconstruido con la información proporcionada por la base de datos Wormatlas [64], al unir con una arista dos neuronas para las que se conozca una interacción, sin importar la naturaleza de esta última. Además, en [19] se desarrolla la detección de comunidades sobre dicho grafo no dirigido, por medio del método conocido como *Modelo Estocástico de Bloques*, y se presenta una partición en 9 comunidades (A3), que fue de utilidad para nuestro análisis comparativo.

Más aún, las neuronas en el sistema nervioso somático de *C. Elegans*, se pueden agrupar también por su morfología y el tipo de conexiones sinápticas que cada una desarrolla, en un total de 103 clases. Sin embargo, habiendo retirado tres neuronas, se obtiene una partición de los vértices en esta red en 102 clases (A1). Por otro lado, también es posible agrupar las neuronas en este grafo por medio de su pertenencia a 10 ganglios anatómicos distintos (A2), igualmente proporcionados por Wormatlas [64].

Así, para esta tesis llevamos a cabo un análisis de comunidades, comparando las particiones (A1) y (A2) contra: los agrupamientos recuperados por medio de MODC, las comunidades detectadas con el algoritmo Clauset-Newman-Moore, y la partición (A3) dada en [19] por el Modelo Estocástico de Bloques. Para comparar todas estas particiones, se recurrió a la medida conocida como información mutua normalizada, utilizándola como se plantea en [29].

Comparación por medio de la información mutua normalizada

La información mutua normalizada es una medida basada en la teoría matemática de la información, y permite evaluar la similitud entre dos particiones de un conjunto [6]. Aunque el estudio de esta medida se encuentra fuera del alcance de esta tesis, a continuación se da una breve descripción de sus propiedades.

Representando dos particiones $P_1 = \{P_{11}, P_{12}, ..., P_{1n}\}$ y $P_2 = \{P_{21}, P_{22}, ..., P_{2m}\}$ de un conjunto dado, una con n clases y la otra con m clases, la **información mutua** entre ellas $I(P_1, P_2)$, indica cuanto se puede saber sobre P_1 si se conoce a P_2 y viceversa. Por otro lado, la información mutua normalizada, toma en cuenta la incertidumbre asociada a cada partición, representada por la medida llamada entropía $H(P_1)$ de cada una de las particiones. Con esto, la **información mutua normalizada** se define como el cociente

$$I_{norm}(P_1, P_2) = \frac{2I(P_1, P_2)}{H(P_1) + H(P_2)} \tag{4.17}$$

Así, la información mutua normalizada es igual a 1 cuando las particiones P_1 y P_2 son idénticas, mientras que tiene un valor esperado de 0 si las particiones son independientes. En este trabajo utilizamos la función llamada *normalized mutual information score* de la biblioteca Sklearn [32], para programar en lenguaje Python la comparación de las comunidades detectadas en distintas ocasiones.

Grafos para evaluar los efectos del límite de resolución

Por último, Fortunato y Barthélemy presentan en su estudio sobre el límite de resolución de la modularidad [15], dos grafos (fig. 4.8) útiles para la evaluación de los efectos que tiene este problema sobre los métodos de detección de comunidades basados en la optimización de esta medida.

El primero de estos grafos, representa un ejemplo conformado por 30 cliques de orden 5, de modo que cada clique se conecta con otros dos hasta formar un arreglo circular. En este, el objetivo reside en detectar cada clique como una única comunidad. A su vez, el segundo ejemplo consiste en un grafo conformado por dos cliques compuestos por 20 vértices y dos cliques con 5 vértices cada uno, donde se busca distinguir a los 4 cliques por separado. Para nuestro estudio, aplicamos MODC a ambos grafos, obteniendo resultados que se muestran dentro de la siguiente sección.

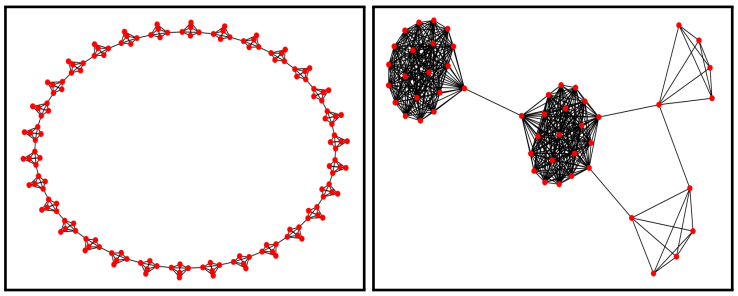

Figura 4.8: Arreglo circular de 30 cliques y grafo compuesto por 4 cliques.

4.5 Resultados

Aplicación a grafos pequeños

Para los grafos pequeños se obtuvieron entre 2 y 3 comunidades (fig. 4.9). En el caso del grafo *Escoba-Doble*, se obtuvieron 2 comunidades, conectadas entre sí por una única pareja de vértices. Esta partición proporcionó una modularidad máxima, como definida en (4.4.5), de 0.5 a una dependencia de 0.023. Por otro lado, para *5,7-Paleta*, que es un grafo conformado por un clique de orden 5 y un camino de longitud 7 sujeto a dicho clique, se identificaron dos comunidades, siendo cada una de estas, una parte del mencionado camino, y el clique junto con una arista que lo une al camino. En este último caso la modularidad máxima se detectó para una dependencia de 0.001, y fue de 0.25.

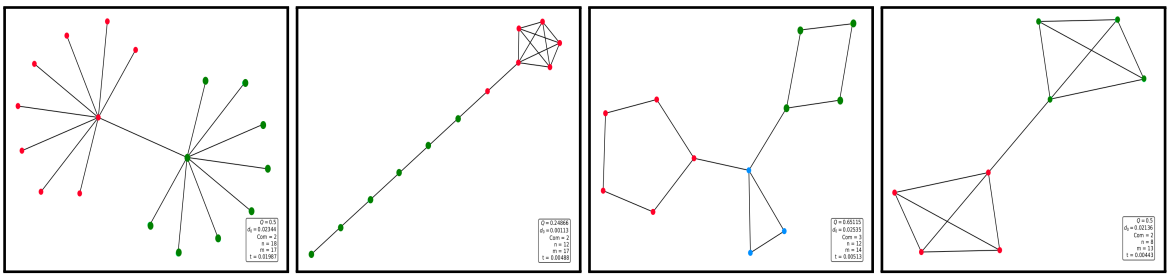

Figura 4.9: Comunidades en grafos pequeños: 2 en Escoba-Doble, 2 en 5,7-Paleta, 3 en Tres-Ciclos y 2 en Cuadrados-Completos.

En cuanto al ejemplo de *Tres-Ciclos*, que como su nombre lo indica esta formado por ciclos, en específico un triángulo, un cuadrado y un pentágono, unidos por escasas aristas, fue posible detectar cada uno de estos ciclos por separado, obteniendo una modularidad de 0.65 con una dependencia de 0.025. Por último, para el grafo *Cuadrados-Completos*, constituido por dos grafos completos de cuatro vértices unidos por una sola arista, se obtuvo una modularidad máxima de 0.5, con una dependencia de 0.021, para una partición en donde se encuentran separados tales subgrafos completos.

Análisis de la red del club de karate de Zachary

La red conocida como el club de karate de Zachary, se encuentra conformada por las relaciones de amistad entre 34 participantes de una asociación deportiva, y fue descrita primeramente en [31] por Zachary y Wayne. Dicha investigación, se enfocó en el análisis de los agrupamientos que surgieron cuando este club se separó en otros dos, debido tanto a conflictos entre los organizadores de cada uno de los nuevos clubes, como a la afinidad que tenían los participantes por las ideas de estos organizadores. En la figura (4.10. a) se muestran enumerados de 1 a 34 los vértices que representan a cada uno de los participantes del club original, y por medio de círculos blancos y cuadrados grises se señalan las comunidades conformadas por las dos asociaciones resultantes de la fisión de dicho grupo.

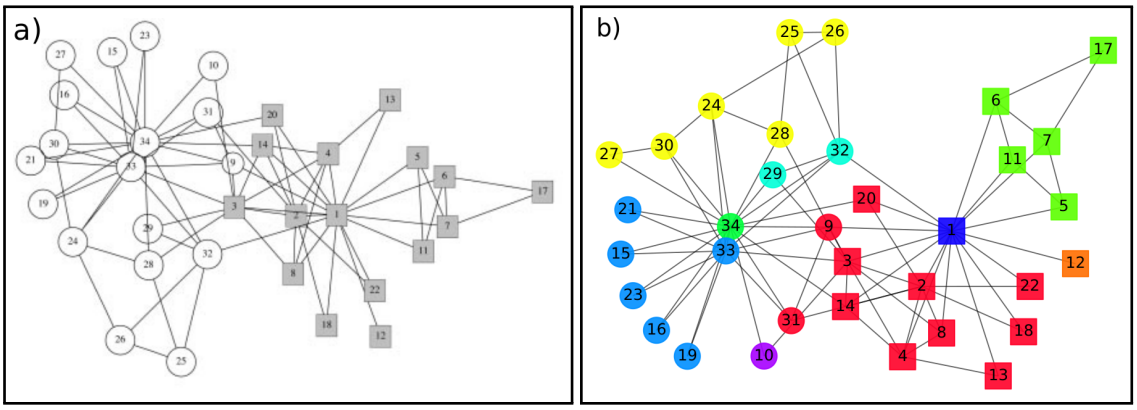

Figura 4.10: Partición esperada y partición obtenida por MODC en la red del club de karate de Zachary.

Zachary y Wayne plantearon un método de análisis de agrupamientos en grafos, con el que detectaron correctamente a las asociaciones de karate que estudiaron por un periodo de 3 años, fallando solo en asignar al vértice 9 su verdadera asociación. Posteriormente, por medio del algoritmo Girvan-Newman, se logró detectar acertadamente en [16] a las dos asociaciones descritas por Zachary, con la excepción de que el vértice indicado por el número 3 fue clasificado erróneamente. Debido a esto, esta red se ha convertido en un ejemplo básico en el estudio de las comunidades en grafos, representando un ejercicio concreto donde el objetivo radica en lograr detectar a los grupos de karate. Por medio de MODC se detectaron 9 comunidades con una modularidad máxima de 0.31 y dependencia de 0.006, contenidas o anidadas en los grupos descritos por Zachary. Así, en la figura (4.10. b), se muestra cómo todos los vértices (a excepción de los enumerados con 9 y 31), que tienen un mismo color (comunidades detectadas con MODC), también están etiquetados con el mismo distintivo (círculo o cuadrado) de la asociación a la que pertenecen.

Aunado a esto, en [31] se menciona que el participante con número 10, independientemente del club al que pertenecía, no presentaba afinidad con la ideología de ninguno de los coordinadores, situación que puede explicar el reconocimiento por medio del procedimiento MODC, de este vértice como una comunidad por sí solo. Se debe mencionar que este también era el caso de los vértices 17 y 19, sin embargo, estos presentan relaciones de amistad solo con vértices de su mismo club, mientras que 10 muestra una relación con el vértice 3, que se encuentra en una asociación distinta a la de 10, lo que sitúa a este último como un contacto entre los dos clubes.

Análisis de grafos de prueba I: Planted-Models

De todos los planted-models generados, se obtuvieron grafos con aproximadamente 1000 aristas, todos construidos con 128 vértices. Se generaron 100 grafos por cada pareja (z_{in}, z_{out}) planteada en la metodología de esta tesis, promediando por cada par: la cantidad de comunidades, la modularidad máxima, la dependencia que ofrece esta modularidad máxima y la información mutua normalizada entre MODC y el algoritmo CNM. En este apartado se muestran los resultados de las primeras tres propiedades, y en la sección de discusión se analizan los valores para la información mutua normalizada.

Junto con los procedimiento MODC y CNM, se desarrolló también el análisis con MODC considerando una exclusividad igual para toda pareja de vértices, versión que se nombró *MODC_simple*, con propósito de evaluar la relevancia de la hipótesis de centralidad de arista (H3).

En la figura 4.11 se incluyen algunos resultados de las comunidades proporcionadas por CNM y MODC para 3 pares de grados promedio, de forma que se aprecia cómo estas se van volviendo más difíciles de detectar conforme aumenta z_{out}. Por otro lado en la gráfica de la figura 4.12 se observa cómo con MODC se detectan inicialmente los 4 agrupamientos debidos. Sin embargo, no fue posible detectar 4 comunidades en torno a $z_{in} = z_{out} = 8$, pareja para la que se determinaron cerca de 50 comunidades.

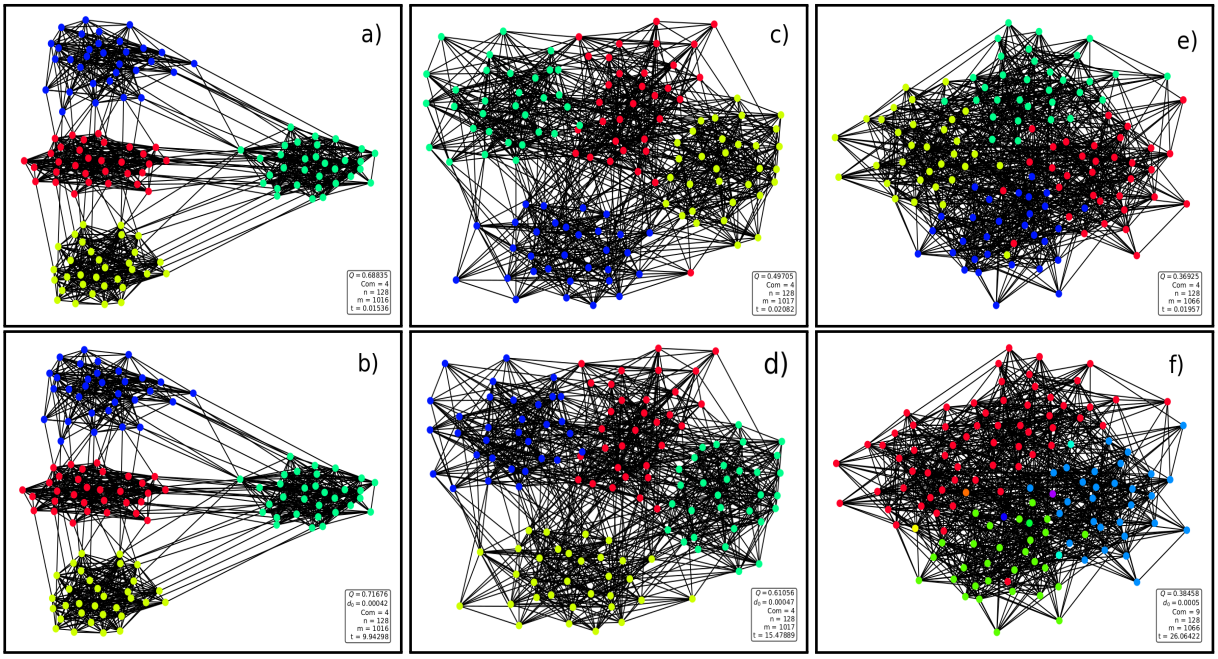

Figura 4.11: Planted-Models: a) y b) $z_{out} = 1$, c) y d) $z_{out} = 4$, y e) y f) $z_{out} = 6$.
Además: a), c) y e) fueron detectados con CNM, mientras que b), d) y f) con MODC.

No obstante, se debe recalcar que con MODC_simple, este valor es todavía mayor, siendo que incluso para la pareja $(z_{in} = 15, z_{out} = 1)$ se obtiene ya más de 50 agrupamientos, lo que refuerza nuestra confianza en (H3). Se obtuvo un efecto similar para la modularidad promedio de todas estas particiones (fig. 4.13), donde MODC y CNM aportan inicialmente una modularidad aproximada a 0.7, que decae hasta valores entre 0.1 para MODC y 0.2 para CNM, mientras que MODC_simple se mantiene en todo caso con una modularidad máxima entre 0.5 y 1.5. Estas modularidades se obtuvieron con MODC para una dependencia entre 0.0004 y 0.0006.

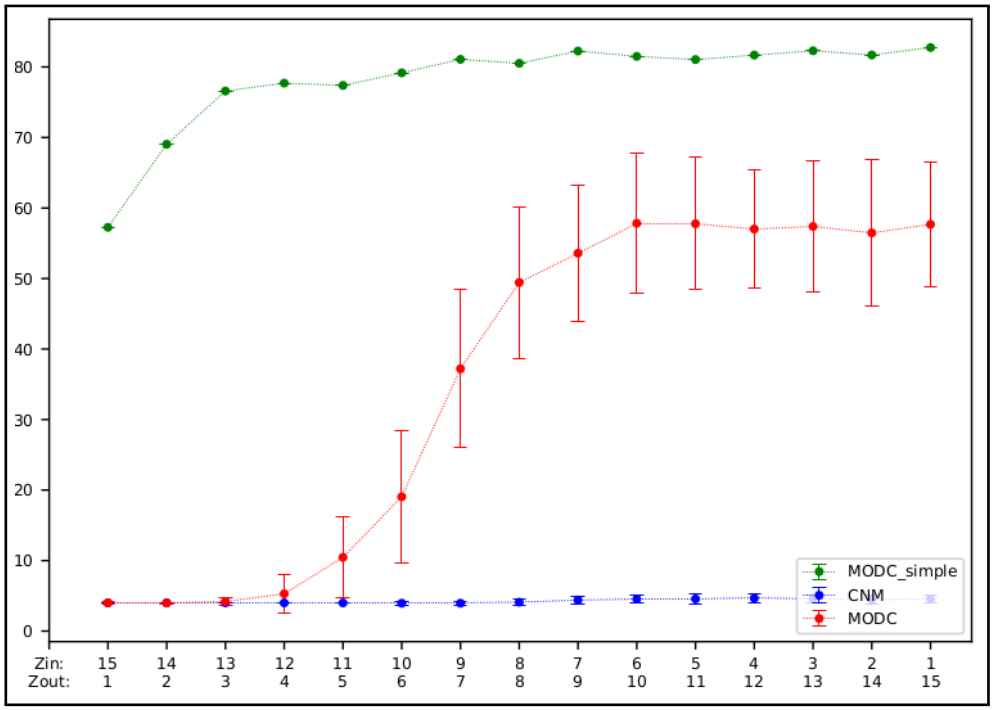

Figura 4.12: Cantidad de comunidades promedio y su desviación estándar (lineas verticales).

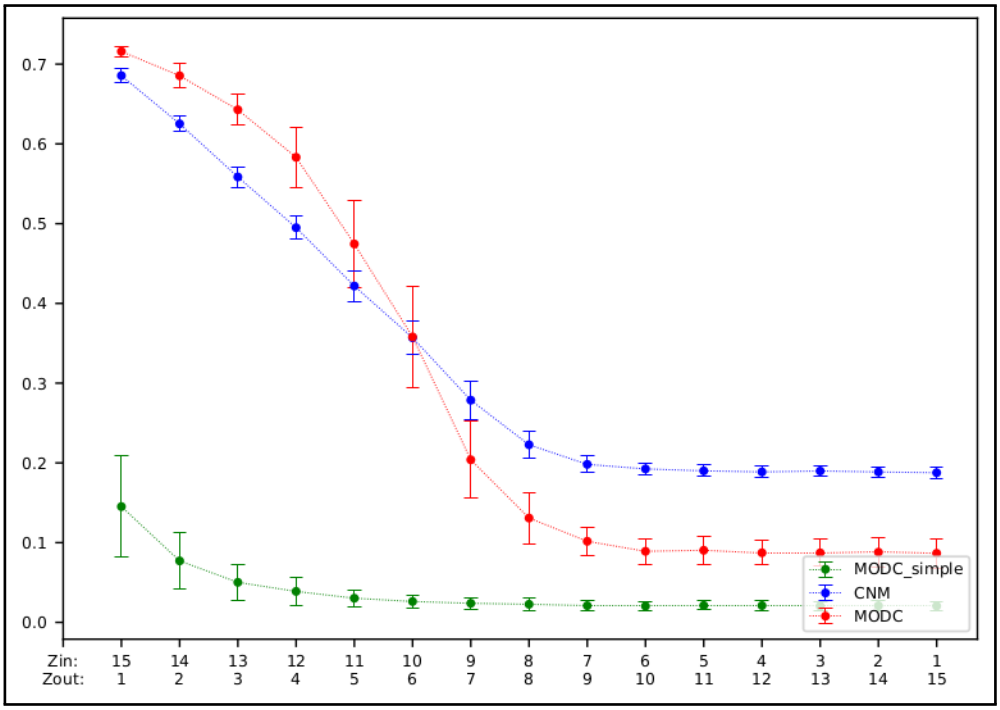

Figura 4.13: Modularidad promedio para planted-models y su desviación estándar.

Análisis de grafos de prueba II: Modelos-LFR

En el caso de los modelos-LFR, se generaron grafos con una cantidad de aristas que varía entre 2600 y 3000, siempre con 500 vértices. En este caso se produjeron 100 grafos por cada valor del parámetro de mezcla $\mu = 0.1, 0.15, 0.2, ..., 0.6$, promediando igualmente: la cantidad de comunidades, la modularidad máxima y la dependencia que ofrece esta modularidad máxima.

En este caso se contó con la partición ideal de cada grafo, etiquetada como partición *REAL*, proporcionada por la función de la biblioteca NetworkX [65] utilizada para generar estos modelos, de modo que las propiedades mencionadas también se promediaron para este agrupamiento objetivo. Además, en la sección de discusión se incluye la información mutua normalizada entre la partición real y las comunidades detectadas con MODC, CNM y nuevamente con MODC_simple.

De esta forma, en la figura 4.14 se incluyen resultados representativos de las comunidades reales o esperadas y MODC para 3 valores de mezcla, donde nuevamente se observa que estas son más difíciles de detectar conforme las comunidades se vuelven más *difusas*, entendiendo esto como el aumento del parámetro de mezcla. A diferencia de los planted-models, en esta ocasión se aprecia que MODC presenta algunas dificultades incluso para valores bajos de mezcla. Esto se verifica en la figura 4.15, donde se observa como con MODC se detecta siempre un elevado número de comunidades, alejado de las 20-50 comunidades reales.

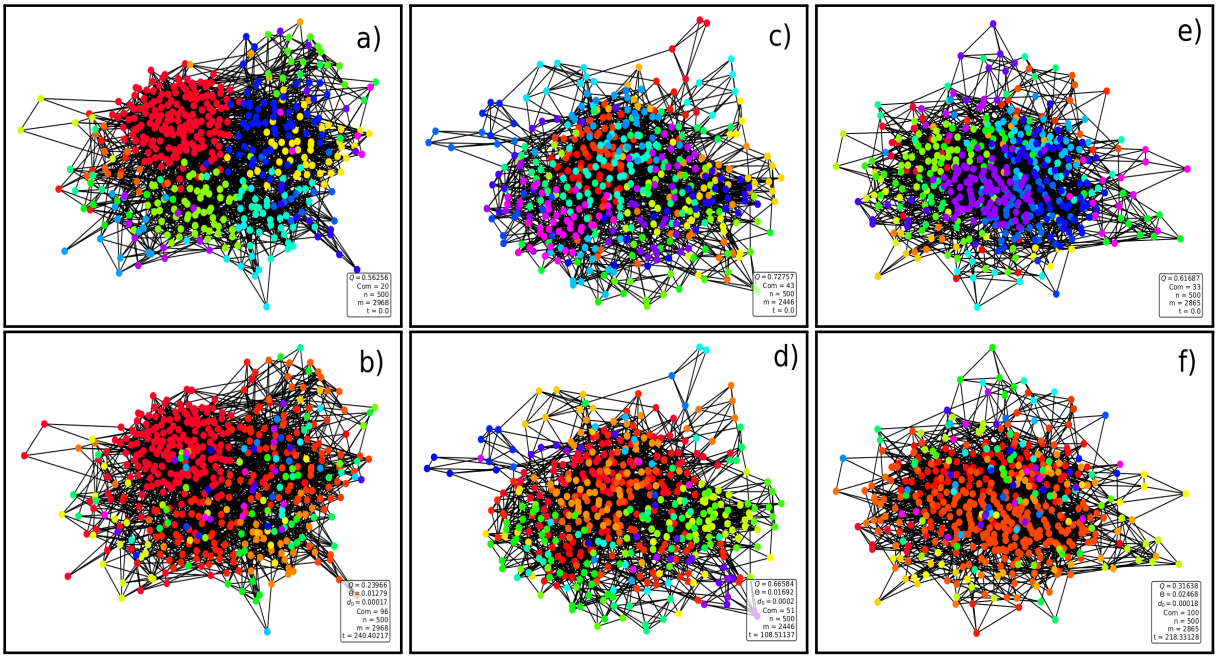

Figura 4.14: Modelos-LFR: a) y b) $\mu = 0.1$, c) y d) $\mu = 0.15$, y e) y f) $\mu = 0.2$.
Donde: a), c) y e) son las particiones Reales, mientras que b), d) y f) se detectaron con MODC.

Sin embargo, nuevamente se obtiene con MODC_simple una cantidad de agrupamientos mayor a la proporcionada por MODC. Aunado a esto, se nota también que el método CNM siempre proporciona un número limitado de comunidades, menor incluso que el contenido en la partición real u objetivo, fenómeno que se ha atribuido con anterioridad al límite de resolución de la modularidad [29]. No obstante, CNM proporciona (fig. 4.16) siempre una modularidad semejante a la de la partición real, tomando valores entre 0.2 y 0.7, mientras que con MODC se obtienen modularidades de 0.1 a 0.5, dadas por una dependencia entre 0.00017 y 0.00021.

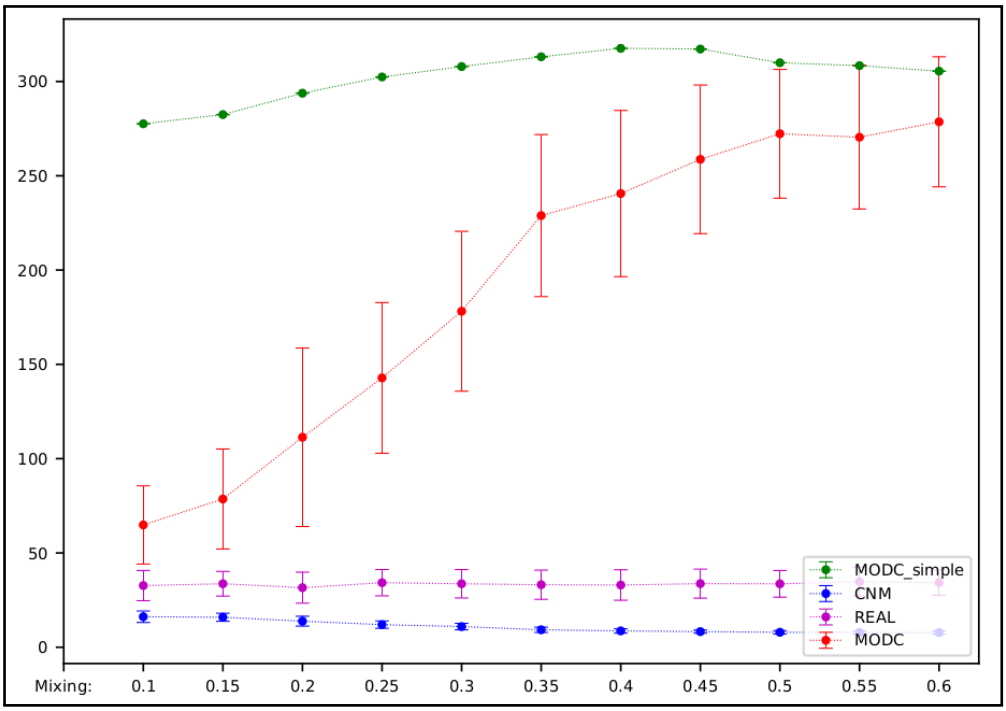

Figura 4.15: Cantidad de comunidades promedio y su desviación estándar en modelos-LFR.

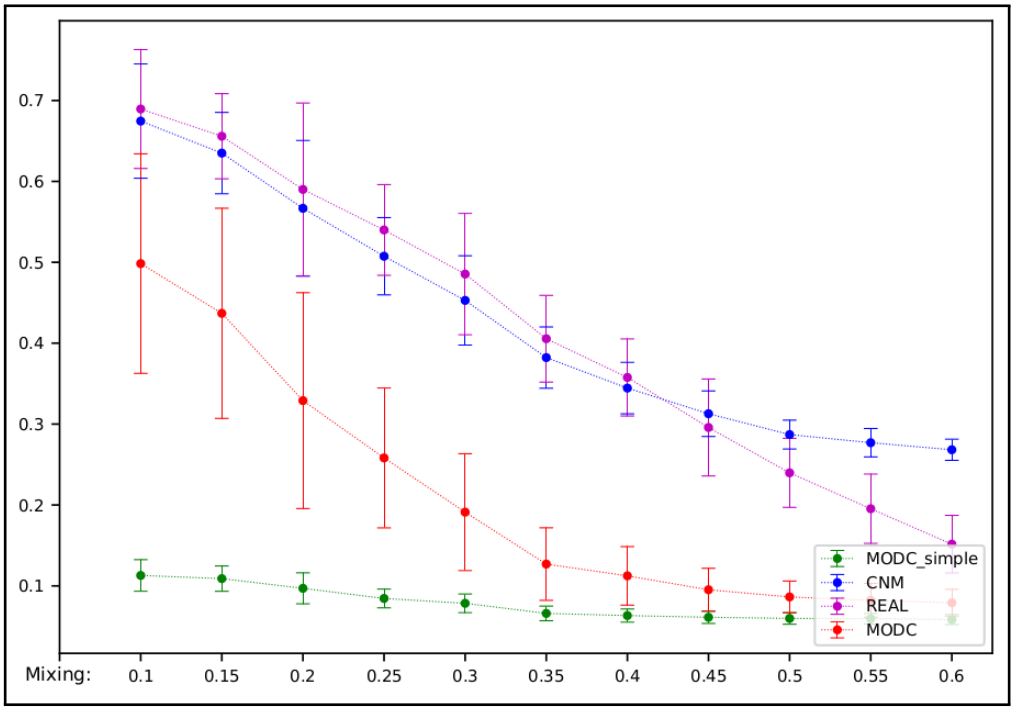

Figura 4.16: Modularidad promedio para modelos-LFR y su desviación estándar.

Análisis sobre *C. Elegans*

En la figura 4.17 se presentan imágenes de la red conexa de orden 279 y tamaño 2287, que modela el sistema nervioso de *C. Elegans*. Junto con esta, se incluyen dibujos de las comunidades detectadas por medio del algoritmo CNM y MODC. Para el caso de CNM se obtuvieron únicamente 4 comunidades, que aportan una modularidad de 0.39, mientras que la partición determinada por MODC se encuentra conformada por 93 grupos, con una modularidad máxima de 0.2 para una dependencia de 0.00023.

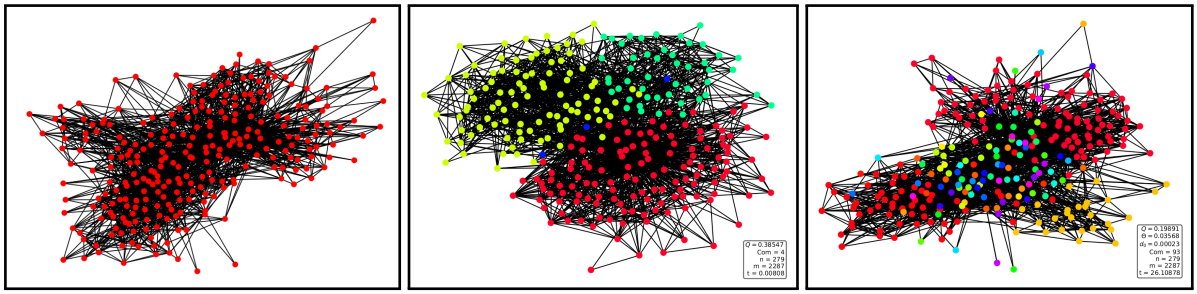

Figura 4.17: De izquierda a derecha: red de *C. Elegans* analizada, partición de esta red con CNM y partición con MODC.

Habiendo determinado tales particiones, y retomando la obtenida con el Modelo Estocástico de Bloques (A3), proporcionada en [19], se procedió a comparar estas contra las particiones biológicas dadas por la distribución de las neuronas en clases relacionadas con su morfología y conexiones (A1), y contra el agrupamiento de las células en ganglios (A2). En al figura 4.18 se muestran los resultados de la información mutua normalizada, obtenidos para cada una de las 6 parejas comparadas. De entre estos 6 valores, se encontró que dicha medida es mayor para la pareja de particiones dadas por MODC y por (A1), que del algoritmo CNM o (A3) contra (A1), situación que se repite al contrastar las 3 particiones teóricas con la que describe el agrupamiento en ganglios (A2).

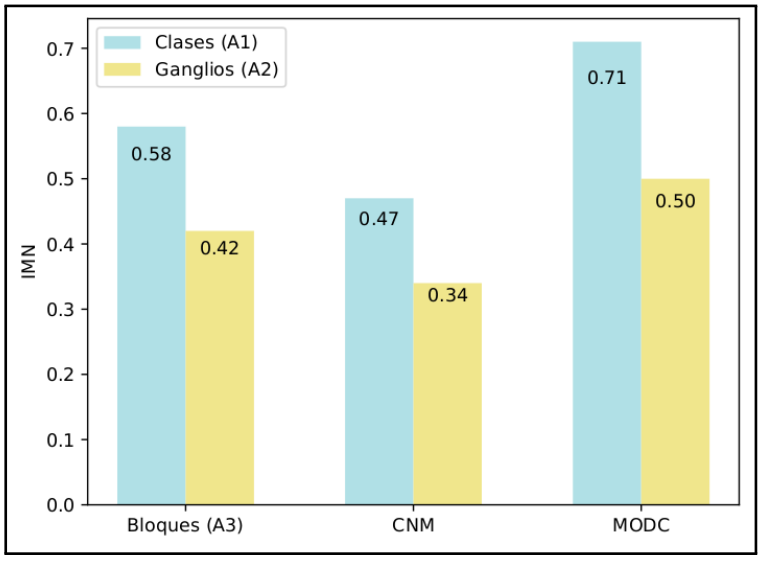

Figura 4.18: Gráfica de barras que compara la información mutua normalizada de las particiones biológicas (A1) y (A2), contra las obtenidas con CNM, MODC, y la (A3) proporcionada en [19].

Análisis de grafos conformados por cliques

Para el arreglo circular de cliques, considerado por Fortunato [15] para evaluar el límite de resolución de la modularidad, se logró determinar una partición en comunidades donde cada clique describe una sola comunidad. Esta partición se detectó con una dependencia de 0.0017, favoreciendo una modularidad de 0.96, muy cercana a 1 y por lo tanto altamente significativa.

Es importante señalar para este caso, que todas las aristas uniendo dos cliques distintos, tuvieron una dependencia de -0.000003, igual a la menor dependencia tomada de entre las aristas del grafo, mientras que las aristas asociando vértices en un mismo clique presentaron dependencias positivas de 0.0001, 0.0017 o 0.0018.

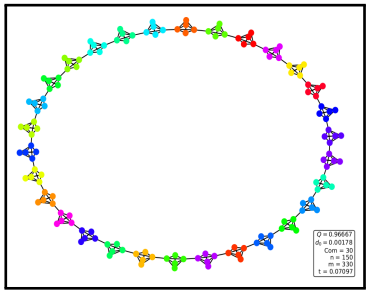

Figura 4.19: Comunidades (30) en arreglo circular de cliques.

Por otro lado, el grafo con 4 cliques no pudo ser particionado como se esperaba. En este caso, se encontraron 2 comunidades, con una modularidad de 0.5 a una dependencia de 0.00065, donde una se corresponde con uno de los cliques de 20 vértices, y la otra está conformada por los cliques restantes. Este resultado sería comúnmente asociado al límite de resolución de la modularidad, sin embargo, nuestro caso se explica por los valores de dependencia en las aristas correspondientes.

Para las aristas dentro del clique de orden 20 unido a los dos pequeños, se encontraron valores de dependencia de ~ 0.00068, no obstante, las 3 aristas que unen a los 2 cliques pequeños entre sí y con el más grande, presentan cada una dependencias de 0.00063, 0.00074 y 0.00111, de forma que las últimas dos no podrían ser borradas por MODC, sin antes haber borrado las aristas dentro del clique grande. Con esto, sería imposible por medio de MODC detectar a los 4 cliques como comunidades por separado, ya que para reconocer a los más pequeños, se tendría que dividir primero al más grande en otras comunidades.

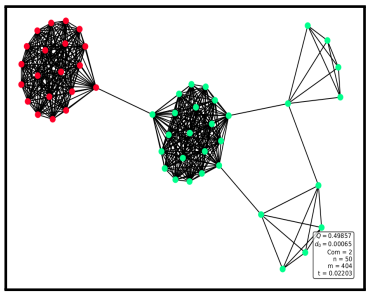

Figura 4.20: Comunidades (2) en arreglo de 4 cliques.

4.6 Discusión

En la gráfica de la figura 4.21 se muestra la información mutua normalizada entre las tres particiones detectadas con CNM, MODC, y MODC_simple sobre los planted-models. Se verifica que inicialmente CNM y MODC otorgan particiones semejantes, facilitando la detección de los cuatro grupos ideales. Mientras, MODC_simple produce nuevamente resultados inconsistentes con la noción de comunidad desde un inicio, aspecto que fortalece nuestra confianza en la hipótesis sobre la centralidad de las aristas.

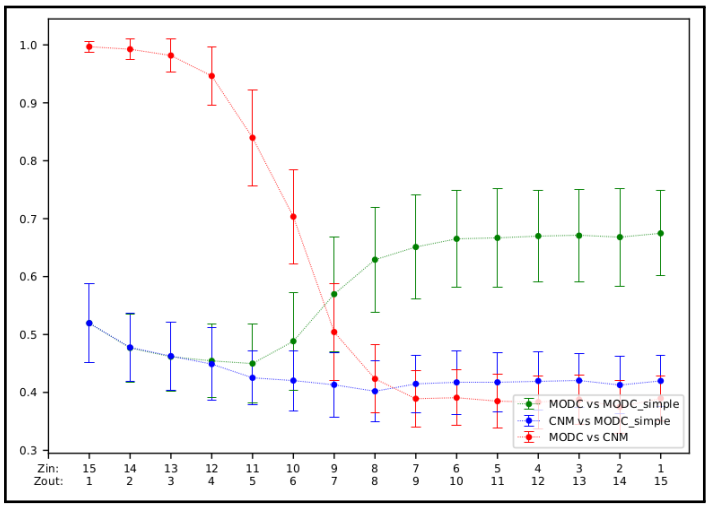

Figura 4.21: Información mutua normalizada promedio en Planted-models.

Esto es semejante a lo obtenido para los modelos-LFR, salvo que en este caso (fig. 4.22) se evaluó la información mutua normalizada de las particiones de CNM, MODC, y MODC_simple contra la REAL o esperada. Aquí MODC parece nuevamente reportar en un inicio particiones semejantes a las obtenidas con el algoritmo CNM, siendo estas a su vez similares (~ 0.9) a la partición real. No obstante, la figura 4.14 de las comunidades detectadas en los modelos-LFR, muestra que MODC no desarrolla una detección clara de las comunidades, ya que estas no presentan parentesco con las particiones reales o esperadas. De esto se intuye que CNM tienen mayor capacidad que MODC para devolver comunidades difusas.

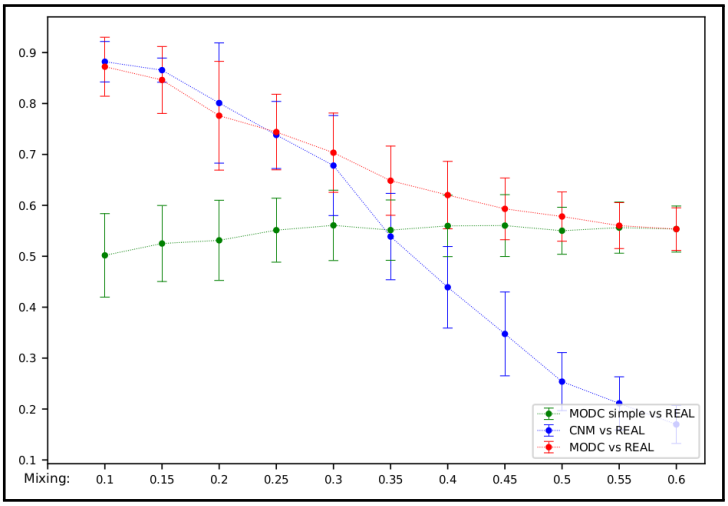

Figura 4.22: Información mutua normalizada promedio en modelos-LFR.

Sin embargo, es importante remarcar que no se encontró referencia alguna en la literatura para la formulación de la modularidad desarrollada en esta tesis, ni tampoco para algoritmos iguales a MODC. Esto implica que tanto nuestra formulación, así como MODC, aportan una nueva perspectiva a la detección de comunidades. En especial, se debe hacer notar que nuestra contribución permite dar argumentos probabilísticos a cada una de las posibles particiones de los vértices de un grafo.

Además, se debe recordar que en el planteamiento del algoritmo Girvan-Newman [16], se advierte no remover las aristas sucesivamente basándose únicamente en sus valores de centralidad. De este modo, los resultados obtenidos sugieren que en nuestro caso, la remoción por medio de los valores de dependencia y no solo de centralidad, permite recuperar favorablemente comunidades siempre y cuando estas no sean suficientemente difusas.

Estos resultados no nos permiten refutar nuestras hipótesis. Sin embargo, es importante reconocer que para comunidades más difusas y en grafos más grandes, MODC aporta un número de comunidades mayor al de las particiones deseadas. Basados en los resultados obtenidos para los modelos-LFR y para los grafos de prueba conformados por cliques, atribuimos este efecto a la definición de exclusividad basada en la centralidad, considerando que la centralidad de arista es útil pero no suficiente para contrarrestar los efectos del aumento de la cantidad de aristas que unen vértices en distintas comunidades.

Por último, se debe notar que al mostrar resultados favorables para comunidades no difusas, y al mismo tiempo determinar un alto número de comunidades, es posible que MODC comprenda restricciones sobre el límite de resolución de la modularidad, además de permitir la exploración de un orden jerárquico de comunidades y tener un tiempo de ejecución mejor que el algoritmo Girvan-Newman. Así, consideramos que nuestra formulación de la modularidad y el procedimiento MODC deben ser estudiados más a detalle.

4.7 Recomendaciones

Se debe remarcar que debido a que MODC es un procedimiento definido para grafos no dirigidos y conexos, el análisis de comunidades sobre la red neuronal de *C. Elegans* se realizó sobre un subgrafo de este tipo, basado en el trabajo presentado en [19]. No obstante, es importante señalar que para modelar fielmente esta red, es necesario recurrir a un grafo dirigido, con etiquetas y múltiples aristas. De esta manera, se debe reconocer que las comunidades determinadas por MODC pueden representar principalmente conexiones anatómicas, más que grupos funcionales de neuronas. Por esta razón, en caso de buscar estudiar los agrupamientos en esta red según la función de cada célula, se recomienda recurrir a trabajos que tomen en cuenta la dirección y naturaleza de cada una de las interacciones [20].

Por otro lado, bajo la suposición de que existe alguna medida que junto con la centralidad de arista permita detectar comunidades incluso cuando estas sean difusas, se recomienda que en futuros trabajos de detección de comunidades por medio de MODC, se amplíe la definición de exclusividad basada en la centralidad para considerar dicha medida.

Se considera que el valor esperado o *esperanza* [45] de la dependencia de cada pareja ordenada de vértices, así como los demás *momentos* asociados a la distribución de esta variable aleatoria en un grafo, pueden llegar a ser importantes para describir un sistema modelado con redes, por lo que se recomienda estudiar a detalle las características e interpretación de dichas medidas.

Además, se debe reconocer que la exclusividad es una definición que parece sobrar, ya que únicamente refleja a los pesos no negativos inicialmente asociados a cada arista. Sin embargo, se debe considerar que si en un futuro se busca ampliar y evaluar este método sobre grafos dirigidos, dicha definición puede resultar de utilidad para facilitar la descripción del peso no negativo asociado a cada flecha.

Se debe considerar también que recientemente Amelio y Pizzuti [67] señalaron que la medida de información mutua normalizada puede presentar un sesgo en ciertos casos, favoreciendo a las particiones que tienen un mayor número de comunidades. Debido a esto, en dicho trabajo se propone una modificación de esta medida, obteniendo así la llamada *información mutua normalizada escalada* (SNMI por sus siglas en inglés). Dado que para muchos de los resultados obtenidos con MODC se determinó un alto número de agrupamientos, se recomienda recurrir a la SNMI para evaluar la utilidad de este y otros algoritmos de detección de comunidades.

Es importante hacer notar que el tiempo de ejecución de MODC se puede reducir en la práctica al analizar, por medio de la programación en paralelo, las particiones de los distintos valores de dependencia, y más aún, esto se puede desarrollar utilizando las facilidades que ofrece el lenguaje Python por medio de la biblioteca Multiprocessing [66].

Finalmente, en el marco teórico de este capítulo se presentó el hamiltoniano de Potts, y se mencionó que este representa una generalización de la modularidad. Debido a esto, se recomienda que en caso de buscar desarrollar mejoras al algoritmo MODC, se estudien primero las posibles relaciones entre el hamiltoniano de Potts y la formulación de la modularidad aquí desarrollada, en caso de que se puedan ampliar primeramente las definiciones de dependencia y modularidad.

Capítulo 5

Aprendizaje automático con propiedades de redes de regulación genética en cáncer de mama

Existen enfermedades propiciadas por múltiples factores tanto genéticos como ambientales. Estas son comúnmente llamadas enfermedades complejas [22]. Ejemplo conciso de estas patologías es el cáncer de mama [9]. Dado que esta afección es difícil de tratar, su detección temprana cobra importancia cuando se busca prevenir el incremento de la mortalidad debida a ella [23].

5.1 Problema tecnológico: clasificación de redes de regulación genética en cáncer de mama por medio de aprendizaje automático

A la fecha existen múltiples ejemplos de aplicaciones del aprendizaje automático como herramienta de apoyo en el diagnóstico e investigación del cáncer de mama y otras enfermedades [68, 69, 70]. Estas se han diseñado principalmente para el diagnóstico asistido por computador en el área de imagen médica, usados por ejemplo, para determinar la probabilidad de que un tumor resulte ser maligno.

No obstante, existen otros tipos de datos que no están relacionados directamente con la imagenología y la radiología. Ejemplo de esto son las redes de regulación genética (fig. 5.1) y las redes de interacciones entre proteínas, de las cuales se puede obtener y comparar información distintiva del cáncer [25, 26, 43, 9, 71].

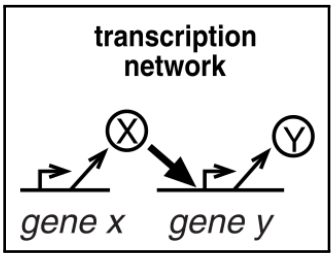

Figura 5.1: Caricatura de una interacción entre dos genes debido a la transcripción de uno de ellos. En conjunto, este tipo de interacciones forman una red de regulación genética transcripcional. *(Imagen tomada y adaptada de [33]).*

Por ejemplo, en el trabajo de Correira et. al [9], se muestra un análisis orientado a la inferencia y clasificación de redes de interacciones entre proteínas, por métodos de aprendizaje automático sobre datos relacionados al cáncer de mama. Específicamente se usaron métodos de clasificación como K-Nearest-Neightbors, Support-Vector-Machine y Random Forest [7], todos de aprendizaje supervisado. Con estos, se buscó distinguir redes inferidas a partir de tejido infestado con células cancerígenas de entre redes inferidas sobre tejido sano.

Bajo la hipótesis de que las interacciones genéticas se ven afectadas en sistemas sujetos al cáncer, se planteó en dicho trabajo clasificar estas redes por medio de 51 propiedades que caracterizan al grafo en su totalidad. Con estas propiedades de redes, y los ya mencionados métodos de aprendizaje automático, Correira y su equipo fueron capaces de clasificar las redes de interacciones entre proteínas con proporciones de aciertos (o exactitudes [7]) cercanas a 0.95.

Motivados por dicho trabajo, en este capítulo evaluamos la capacidad que tiene un sistema de redes neuronales artificiales, para distinguir redes de regulación genética relacionadas con cáncer de mama de entre redes generadas aleatoriamente, utilizando únicamente 14 propiedades de estas redes.

Esto se desarrolla principalmente como una prueba de concepto para posibles trabajos a futuro, buscando explorar las bases para el estudio de la viabilidad que tiene la aplicación de esta metodología, basada en redes neuronales artificiales, en la detección temprana del cáncer de mama.

Es importante recalcar que en general, al implementar un sistema de aprendizaje automático, no únicamente se pone a prueba la capacidad de diferenciar estadísticamente las redes relacionadas al cáncer de aquellas aleatorias, más aún, de lograr resultados favorables también se obtiene un sistema computacional con potencial para predecir esta enfermedad sobre redes inferidas de datos de pacientes que se sospechen sujetos a ella.

Para desarrollar nuestro análisis, se presenta primero una introducción a los conceptos fundamentales sobre aprendizaje automático. Posteriormente se incluyen en el marco teórico las definiciones correspondientes a redes neuronales artificiales, seguidas por la descripción de la metodología empleada para el entrenamiento y evaluación de estas redes. Para concluir se presentan los resultados obtenidos por la clasificación de las redes de regulación genética que modelan cáncer de mama.

5.2 Introducción al aprendizaje autómático o machine learning

El aprendizaje automático, o machine learning (ML), es el estudio de algoritmos diseñados para reconocer (o aproximar) patrones en un primer conjunto de datos empíricos, llamados de entrenamiento, y posteriormente extrapolar estos patrones a datos que no les hayan sido presentados previamente, o de prueba. De esta forma, se logra simular el proceso de aprendizaje por medio de un método computacional [72].

Para aprender, un algoritmo de ML se sustenta en distintas áreas de las matemáticas, como son la teoría de probabilidad, estadística, cálculo, álgebra lineal y optimización. Junto con estas, figuran en el análisis y desarrollo de estos algoritmos múltiples métodos de minería de datos y diseño de experimentos [7]. De esta manera, el ML es una área de las ciencias de la computación enfocada al análisis de datos, pero por medio de programas capaces de adquirir experiencia de sus propias *observaciones*.

Los distintos algoritmos de ML tienen, de forma muy general, dos maneras para *estudiar* los datos que se les presentan. Estas formas dependen de si existe o no un mecanismo por el que un algoritmo obtenga *retroalimentación*, es decir, el medio por el que un programa reconoce cuándo ha aprendido algo correctamente. Así, se tienen los llamados métodos de *aprendizaje no supervisado* y *aprendizaje supervisado*.

Generalidades del aprendizaje no supervisado

Los métodos del aprendizaje automático no supervisado se caracterizan por no tener una retroalimentación. En estos, el reconocimiento de patrones se da por medio de medidas de similitud establecidas sobre los datos a analizar, sin que en estos se indiquen los patrones específicos que se deben aprender.

Un ejemplo concreto son los algoritmos que detectan agrupamientos en datos sobre un espacio Euclideano [73], donde el parentesco entre toda pareja de datos puede estar dado por umbrales sobre la distancia entre ellos. Así, un algoritmo de este tipo puede reasignar los agrupamientos entre los datos cada vez que más de estos le son proporcionados, sin saber de antemano a que grupos pertenece cada vector de datos.

Específicamente, el aprendizaje no supervisado está definido como la tarea que desarrollan ciertos algoritmos, al aprender aspectos asociados a datos que no presentan etiquetas, usando principalmente técnicas geométricas o criterios estadísticos [74]. Debido a que los datos no presentan patrones *a priori*, las inferencias realizadas con estos métodos se pueden considerar como no sesgadas.

No obstante, existen problemas para los que es preferible contar con las características que se desean estudiar en los datos, de forma que se puedan poner a prueba hipótesis específicas, como es el caso de nuestra clasificación de redes de regulación genética y redes aleatorias. Los algoritmos que toman en cuenta esto son llamados métodos de aprendizaje supervisado.

Generalidades del aprendizaje supervisado

En el aprendizaje supervisado los datos siempre deberán estar etiquetados según los patrones que se busca reconocer, factor de donde surge la retroalimentación para estos algoritmos. Tal etiqueta es llamada variable de respuesta, y es la que permite evaluar y redirigir el aprendizaje sobre los datos de entrenamiento, pero también es la variable que se busca predecir sobre los datos de prueba.

La variable de respuesta puede ser numérica (discreta o continua) o categórica. Por ejemplo, una variable numérica podría ser la esperanza de vida de un paciente, mientras que una variable categórica puede ser un indicador de la presencia o etapa de un tumor maligno. Además, es importante hacer notar que una variable categórica puede representarse por números enteros para efectos prácticos.

En general, todos los algoritmos de aprendizaje supervisado cuya variable de respuesta es numérica son conocidos como regresores, mientras que aquellos que cuentan con variables categóricas son llamados clasificadores. En este trabajo nos enfocaremos en estudiar un tipo particular de método de aprendizaje supervisado, llamado *red neuronal artificial*, que en primera instancia puede ser utilizada tanto como regresor o como clasificador.

Redes neuronales artificiales

Las redes neuronales artificiales se encuentran inspiradas en el funcionamiento del sistema nervioso de los organismos vivos [24]. De esta forma, estos sistemas computacionales están conformados por unidades de procesamiento llamadas neuronas, que desarrollan intercambios de información con el objetivo de detectar patrones sobre datos.

Una red neuronal artificial modificará sus parámetros internos de intercambio de información, buscando poder aproximar poco a poco los datos que le son presentados. Para facilitar el estudio de estas redes, analizaremos su *estructura* por separado de su *comportamiento* (fig. 5.2).

Por estructura de una red neuronal artificial, nos referimos a las propiedades que tienen las neuronas en ella, y la manera en que estas están conectadas. En el marco teórico se abordarán las definiciones correspondientes a estas características. Sin embargo, por el momento cabe mencionar que esta estructura se puede formular respecto a la definición de **grafo acíclico dirigido** o **DAG** (directed aciclic graph) [7]. Es decir, una red neuronal artificial será, para este trabajo, un grafo dirigido sin ciclos ni lazos cuyos vértices son llamados neuronas, de manera que la información en esta red pueda propagarse entre las neuronas desde un punto de acceso hasta un punto de respuesta.

Mientras la información se propaga entre las neuronas, esta se verá modificada por los parámetros internos de la red, de modo que al verse alterados dichos parámetros, la red podrá simular un aprendizaje. Para lograr este comportamiento, se desarrollará el entrenamiento de estos sistemas por medio del algoritmo de retropropagación, del cual también se habla en la siguiente sección.

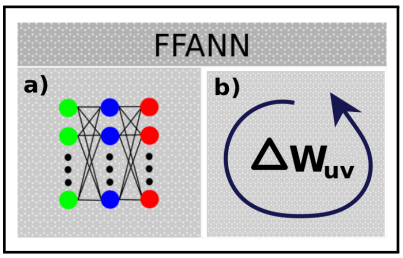

Figura 5.2: Caricatura de una red neuronal como una caja gris, donde se muestran las ideas de a) su estructura, y b) su comportamiento.

5.3 Marco teórico

Primero se presentan los conceptos relacionados con el cáncer y las redes de regulación genética. Posteriormente se habla sobre la estructura de una red neuronal artificial y sobre el algoritmo de retropropagación. Por último se presentan dos métodos que comúnmente se utilizan para desarrollar el entrenamiento de una red neuronal artificial.

Origen del cáncer

A lo largo de su vida, una célula pasa por una serie de fases que en conjunto son conocidas como ciclo celular [5]. A través de este, la célula crece y se duplica en otras que la sustituyen en el tejido donde esta habitaba. Durante cada duplicación, y en todo el ciclo, la información genética de una célula se puede ver afectada de distintas formas, sufriendo así mutaciones químicas que en ocasiones modifican drásticamente el comportamiento de la célula. Si este tipo de mutaciones se llegan a acumular, el ciclo celular es súbitamente terminado, y comienza el proceso de la apoptosis, o muerte programada de la célula, iniciado por esta misma o por el sistema inmunológico del organismo al que pertenece.

Sin embargo, en ocasiones, esta misma acumulación de mutaciones provoca que la célula evada la apoptosis, de modo que puede continuar con el ciclo celular sin cumplir las funciones que debería desarrollar normalmente. El peligro de que esto suceda, radica en que la célula pasará eventualmente por la duplicación, y las nuevas dos células que de esta se originarán, tampoco cumplirán las funciones que de ellas se requieren y nuevamente evadirán la apoptosis. Un agrupamiento formado por este tipo de células, y que haya adquirido un tamaño considerable, es comúnmente conocido como tumor [28]. Este último puede ser a su vez benigno, si no afecta al tejido que lo rodea, o maligno, si este perjudica a los órganos adyacentes y consume la energía que otras partes del cuerpo requieren. Este último tipo de tumores son lo que se conoce como cáncer, y la migración de las células en ellos al resto del cuerpo, es llamada metástasis.

Ya que el comportamiento de una célula se deriva de la regulación de los genes en ella, y dado que en el cáncer, las interacciones genéticas se ven modificadas debido a las mutaciones que los mismos genes sufren [28], es de interés analizar esta enfermedad por medio de grafos que muestren las interacciones entre estos genes, modelos conocidos como redes de regulación genética.

Redes de regulación genética

Las funciones de una célula se ven principalmente determinadas por la información genética en ella. Esto se debe a que la información en algunos de los genes del ADN de una célula es transcrita a ARN, para después ser traducida a proteínas.

A este proceso se le conoce como expresión genética [25], y posterior a él, las proteínas pasarán a ser parte de cada una de las reacciones y estructuras que dan forma a una célula. Con esto, la cantidad de ARN o proteínas que se obtienen en cierto tiempo y bajo ciertas condiciones a partir de un gen específico, es llamada nivel de expresión de ese gen.

Dentro de una célula existen múltiples tipos de interacciones moleculares. Entre estas destacan aquellas que ocurren entre genes y proteínas. Cuando entre estos se da una interacción que determina o modifica el nivel de expresión de un gen, entonces se dice que tal interacción es de regulación genética.

Debido a que el efecto de un gen o proteína sobre el nivel de expresión de otro no es recíproco, estas son modeladas por medio de grafos dirigidos. Así, una colección de interacciones de regulación genética presentes en un organismo es llamada red de regulación genética.

Por otro lado, es posible que un gen afecte su propia expresión por medio de la o las proteínas que de él se derivan. A este proceso se le conoce como autorregulación, y también se puede modelar por medio de lazos en las redes de regulación genética. De la misma manera, es posible que las interacciones genéticas se den por más de una reacción bioquímica [25], aspecto que se puede estudiar con multiaristas en una red. Sin embargo, en este trabajo trataremos únicamente con redes de regulación genética dirigidas y sin lazos ni múltiples aristas.

Estructura de una red neuronal artificial

En este apartado se presentan las definiciones de las propiedades que tienen los vértices de una red neuronal artificial y la manera en la que estos están conectados. Se debe señalar que para establecer nuestras definiciones nos apoyaremos completamente en el trabajo de Raúl Rojas [24], donde se presentan definiciones con un carácter más general, así como un tratamiento amplio de la historia y características de las redes neuronales artificiales.

Dado un digrafo G, denotaremos por V al conjunto de vértices de G y por A al conjunto de flechas en G. Además, para un vértice $v \in V$, nos referiremos como **vecindario de salida** de v al conjunto $N_v^+ = \{u \in V | (v,u) \in A\}$ y como su **vecindario de entrada** al conjunto $N_v^- = \{u \in V | (u,v) \in A\}$.

Ya que un DAG no presenta ciclos, entonces siempre existirá en él por lo menos un vértice con vecindario de salida vacío, o lo que es lo mismo, grado de salida cero, misma razón por la que también existirá al menos un vértice con grado de entrada cero [4]. Por medio de un DAG, buscaremos que en una red neuronal artificial se pueda reproducir la propagación de información desde un punto de acceso hasta un punto de respuesta.

Por esta razón, una red neuronal artificial se definirá por medio de un DAG, de modo que el punto de acceso de la información a la red estará dado por el vértice con grado de entrada igual a cero, mientras que el punto de respuesta será el vértice con grado de salida cero. Sin embargo, para garantizar la correcta propagación de la información en la red, es necesario especificar que este DAG sea débilmente conexo. Tomando en cuenta estos requisitos proponemos la siguiente definición.

Definición 5.3.1. *Sea $G = (V, A)$ un grafo dirigido acíclico y débilmente conexo. Dado algún entero n, diremos que G es un **DAG de propagación hacia el frente**, si para V se puede encontrar una partición $\tilde{P} = \{Q_1, Q_2, ..., Q_n\}$, tal que:*
i) $|Q_n| = 1$ y $v \in Q_n$ cumple $N_v^+ = \varnothing$.
ii) si $(u, v) \in A$, entonces $u \in Q_i$ y $v \in Q_{i+1}$, con $i = 1, 2, ..., n-1$.
iii) para todo $u \in Q_i$, existe por lo menos un $v \in Q_{i+1}$, tal que $(u, v) \in A$, para $i = 1, 2, ..., n-1$.
iv) para todo $u \in Q_i$, existe por lo menos un $v \in Q_{i-1}$, tal que $(v, u) \in A$, para $i = 2, ..., n$.

Si tales propiedades se cumplen, entonces cada clase Q_i de la partición $\tilde{P}$ será llamada i**-ésima** capa del DAG de propagación hacia el frente (fig. 5.3). Ya que todo vértice en las capas con $i = 2, 3, ..., n-1$ debe apuntar y ser apuntado por vértices en capas adyacentes, estos deberán pertenecer por lo menos a un camino dirigido desde los vértices en Q_1 hacia el vértice en Q_n, donde solo el vértice con grado de salida cero estará en Q_n, y de forma similar, los vértices con grado de entrada cero pertenecerán a Q_1.

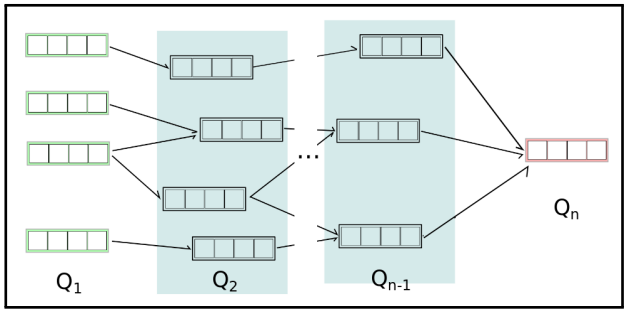

Figura 5.3: Ejemplo de un DAG de propagación hacia el frente.

En la siguiente definición se condensan las propiedades que deben tener las flechas y los vértices de un DAG de propagación hacia el frente para poder realizar la transferencia de información. Esta definición contempla los mismos valores utilizados por Raúl Rojas [24] para manipular la información sobre los vértices de una red neuronal artificial. No obstante, a diferencia de Rojas trataremos estos valores como asociados directamente a cada vértice, con el único propósito de simplificar nuestra explicación sobre el mecanismo por el que se da la transferencia de información en una red neuronal.

Definición 5.3.2. *Sea $G = (V, A)$ un DAG de propagación hacia el frente, diremos que G es una **red neuronal artificial de propagación hacia el frente**, o **Feed Forward Artificial Neural Network (FFANN)**, siempre y cuando i) para toda flecha $(u, v) \in A$ se tenga definido un peso $w_{uv} \in \mathbb{R}$ y ii) todo vértice $v \in V$ tenga asociada una tupla $(x_v, o_v, o'_v, \delta_v)$ de cuatro números reales que quedan definidos como sigue:*

1. *Si $N_v^- = \varnothing \rightarrow x_v$ puede tomar un valor real arbitrario. Pero si $N_v^- \neq \varnothing \rightarrow x_v = \sum_{u \in N_v^-} o_u w_{uv}$*
2. *$o_v = F_v(x_v)$, para cualquier función F_v continua y diferenciable, llamada **función de activación** de v*
3. *$o'_v = \frac{dF_v}{dx_v}$*
4. *Si $N_v^+ = \varnothing \rightarrow \delta_v = o'_v$. Pero si $N_v^+ \neq \varnothing \rightarrow \delta_v = (o'_v)(\sum_{u \in N_v^+} w_{vu}\delta_u)$*

*En particular, la función de activación F_v del único vértice $v \in Q_n$ es conocida como **función de red**.*

Si un DAG cumple estas definiciones, entonces los vértices en él son llamados neuronas (fig. 5.4). Con estas propiedades, las neuronas en la capa Q_1 podrán recibir datos arbitrarios, sobre los que se harán las operaciones necesarias para obtener retroalimentación por medio de la capa Q_n.

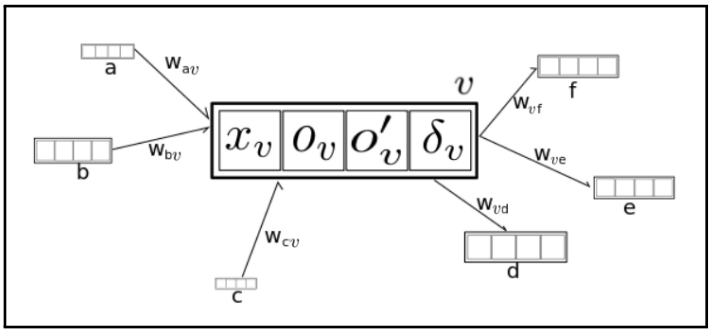

Figura 5.4: Una neurona y sus atributos

Posteriormente, con dicha retroalimentación se realizarán iterativamente modificaciones a los pesos de todas las flechas, que se deben considerar inicialmente como arbitrarios. Estas modificaciones se harán de tal forma que con cada una de ellas la red neuronal podrá aproximar los patrones que se le otorguen. Todos los detalles de este procedimiento se tratarán después de presentar el algoritmo de retropropagación.

Por el momento centramos nuestra atención a la llamada función de activación. Como establecido por Rojas, la función de activación puede ser arbitraria, mientras sea continua y diferenciable. No obstante, para este trabajo nos restringiremos a usar la **función logística** como función de activación, ampliamente usada para este propósito [7, 24], y dada por

$$\phi(x) = \frac{e^x}{1 + e^x} \tag{5.1}$$

para la que se cumple además

$$\frac{d(\phi(x))}{dx} = \phi(x)(1 - \phi(x)) \tag{5.2}$$

característica que facilita la implementación computacional de una red neuronal artificial. Por ahora se puede considerar que todas las neuronas poseen a la función logística como función de activación. No obstante, para lograr un aprendizaje por medio del algoritmo de retropropagación se deberá primero modificar la definición de la función de red.

Aprendizaje por el algoritmo de retropropagación

Antes de hablar respecto al desarrollo del proceso de aprendizaje de una red neuronal, es importante comprender el mecanismo sobre el que este se sustenta. Esta característica no es obtenida únicamente por medio de la estructura de una FFANN, sino que se debe recurrir a un algoritmo concreto que indique como manipular los datos en ella.

Para esto, se presenta a continuación el algoritmo de retropropagación, que consta esencialmente de dos fases: una donde los cálculos sobre las neuronas se hacen siguiendo la dirección de las aristas, o *hacia el frente*, y otra donde estos se hacen en sentido contrario (*backpropagation* o *retropropagación*, lo que le da el nombre a este procedimiento).

Algoritmo 5.3.1 Retropropagación

Input: $G = (V, E)$ una FFANN, con n capas $Q_1, Q_2, ...Q_n$.
Output: el conjunto $\{\delta_v\}$ tomadas de todo $v \in Q_1$

// Paso 1: Dar valores iniciales a la red
dar algún valor real x_v^0 a x_v para todo $v \in Q_1$
evaluar o_v y o'_v para todo $v \in Q_1$ como se establece en la definición 5.3.2
// Paso 2: Propagación hacia el frente
for Q_i con i desde 2 hasta n **do**
  for $v \in Q_i$ **do**
    evaluar x_v, o_v y o'_v como se establece en la definición 5.3.2
  end for
end for
// Paso 3: retropropagación
for Q_i con i desde n hasta 1 **do**
  for $v \in Q_i$ **do**
    evaluar δ_v como se establece en la definición 5.3.2
  end for
end for
return($\{\delta_v\}$ tomadas de todo $v \in Q_1$)

Raúl Rojas demuestra en [24] por inducción sobre la cantidad de neuronas, que a partir del algoritmo de retropropagación, para cualquier neurona $v \in Q_1$ y para la única neurona $u \in Q_n$, se tiene

$$\delta_v = \frac{\partial F_u}{\partial x_v}\Big|_{x_v = x_v^0} \tag{5.3}$$

donde F_u es la función de red, y x_v^0 es el valor real arbitrario asignado a x_v. Es decir, una FFANN, como hasta ahora definida, constituye junto con el algoritmo de retropropagación, un método numérico del cálculo de una derivada, o de forma más general, de un gradiente.

Más aún, el resultado anterior no solo es cierto para las neuronas en Q_1, sino que en genral, δ_v para cualquier neurona v de la FFANN, será la derivada parcial de la función de red F_u respecto a x_v, evaluada según la definición 5.3.2. Para poder hacer uso de este hecho en el aprendizaje de una FFANN, primero se debe plantear una función de red de la que se pueda obtener una retroalimentación en el proceso de entrenamiento.

Existen múltiples formas para desarrollar el entrenamiento de una FFANN. No obstante, todas estas formas se basan en el concepto de *par de entrenamiento*. Este simplemente es una pareja de datos sobre los que se busca que la red detecte un patrón. Para lograr esto, la variable de respuesta de este sistema de ML debe existir en este par de entrenamiento.

Específicamente, al proporcionarle un vector $\vec{x} =< x_1^0, x_2^0, ..., x_a^0 >$ de a números reales a una FFANN con a neuronas en la capa Q_1, se buscará aprender a reconocer al vector $\vec{y} =< y_1^0, y_2^0, ..., y_b^0 >$ de b números reales, que representa a la variable de respuesta. De esta manera, un **par de entrenamiento** para una FFANN será de la forma $(\vec{x}, \vec{y}) \in \mathbb{R}^a \times \mathbb{R}^b$.

Así, por medio de una colección de pares de entrenamiento, se espera que una FFANN aprenda a reconocer una función $\Psi : \mathbb{R}^a \to S \subseteq \mathbb{R}^b$ de la que se supone su existencia, más no se conoce su forma analítica. En particular, si S es una colección de b-adas de números enteros, y estos codifican a un problema categórico, entonces una FFANN que aproxime una función así, será llamada clasificador. Por otro lado, si se tiene interés en recuperar valores continuos, entonces esta FFANN será llamada regresor.

Para poder aproximar esta función, una FFANN deberá obtener retroalimentación por medio de una función de red escrita en términos de los valores de la variable de respuesta, proporcionados por cada par de entrenamiento. Para construir una función de este tipo, se analizará una FFANN conocida como **red neuronal por capas**, que tiene una estructura específica [24]. En este trabajo estudiaremos aquella compuesta solo por 5 capas. En la figura siguiente se muestra esta red, y para facilitar la formulación de sus propiedades, se considera un único par de entrenamiento $(< x_1^0, x_2^0, ..., x_a^0 >, < y_1^0, y_2^0, ..., y_b^0 >)$.

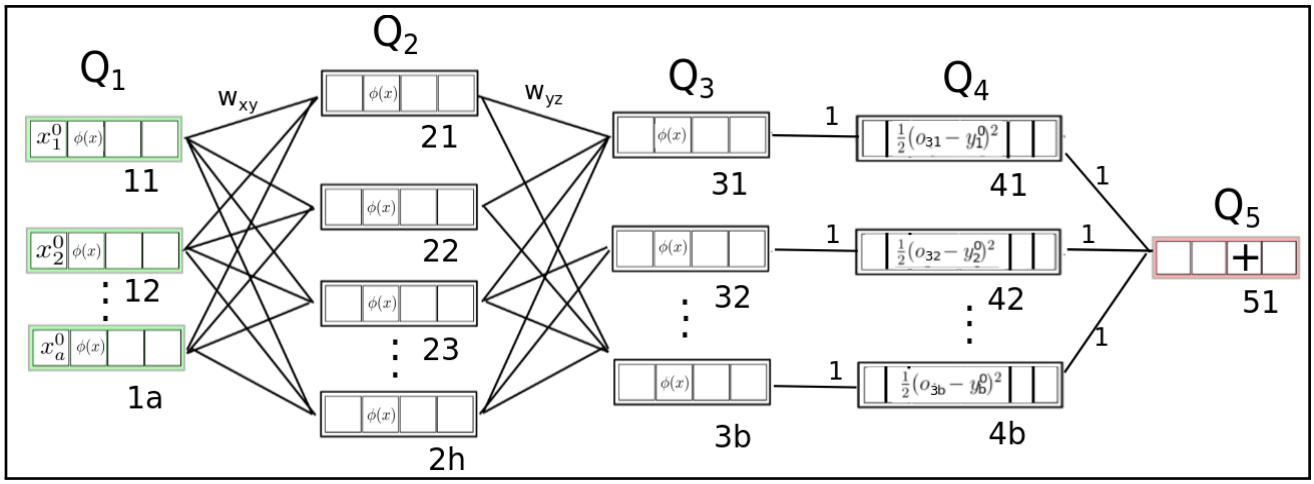

Figura 5.5: Red neuronal por capas. No se muestra el peso de toda flecha, pero se considera que todas estas tienen un peso real arbitrario, a excepción de las que involucran neuronas en las capas Q_4 y Q_5.

En esta, la capa Q_1 recibe comúnmente el nombre de **capa de entrada**, ya que por esta es por donde se proporciona la información a la red. Por otro lado Q_3 es la **capa de salida**, que es de donde se espera obtener la reproducción de la variable de respuesta, al buscar hacer $o_{3i} \simeq y_i^0$ para toda neurona en ella.

Además, la capa Q_2 es conocida como **capa oculta**, y sirve para favorecer la existencia de múltiples arcos entre la capa de entrada y la de salida. Por último, a las capas Q_4 y Q_5 las llamaremos en conjunto **módulo para la evaluación del error** (**MPEE**).

Debe señalarse que en nuestra versión de la red neuronal por capas, todas las neuronas de la capa de entrada se encuentran conectadas por una flecha con todas las de la capa oculta, y estas a su vez con las de la capa de salida. De esta forma, denotando por h a la cantidad de neuronas en la capa oculta, la red tendrá $ah+hb$ flechas, sin contar las que involucran neuronas en el MPEE, que tienen asociados pesos iguales a 1.

Todas las neuronas tienen a la función logística como función de activación, a excepción de las neuronas en el MPEE, ya que por medio estas dos últimas capas es posible formular la función de red. Así, considerando que en cada una de las neuronas en la capa Q_4 se desarrolla la función de activación $\frac{1}{2}(o_{3i} - y_i^0)^2$, y que este es el valor que se proporciona a la neurona $u \in Q_5$, cuya función de activación es una suma, se tiene que la función de red se puede escribir como

$$E := F_u = \frac{1}{2}\sum_{i=1}^{b}(o_{3i} - y_i^0)^2 \tag{5.4}$$

Esta función de activación representa entonces una medida del error con el que una red por capas reproduce o predice una variable de respuesta. De esta manera, al minimizar esta función por medio de su gradiente, la red producirá predicciones $o_{3i} \simeq y_i^0$ con mayor exactitud, simulando así un aprendizaje.

Para poder utilizar este gradiente como medio de aprendizaje se aprovecharán los pesos en las flechas de una red, que hasta ahora han sido definidos como arbitrarios. Sin embargo, se debe notar que cuando estos pesos se ven modificados, las respuestas o_v y δ_v de todas las neuronas se ven alteradas. Recurriremos al método de optimización numérica conocido como **descenso por gradiente** para modificar estos pesos. Con este, los pesos pueden ser modificados iterativamente, añadiéndoles un incremento proporcional a la derivada del error respecto a cada peso, pero con signo opuesto a esta.

Ya que el gradiente de una función representa al vector que apunta en la dirección de máximo incremento de la función, al utilizar este con signo opuesto, se obtienen proporciones que en conjunto apuntan en la dirección de máxima reducción del error. Así, se buscará sumar a cada peso w_{uv} un incremento

$$\Delta w_{uv} = -\lambda \frac{\partial E}{\partial w_{uv}} \tag{5.5}$$

donde λ es una constante para todos los pesos, conocida como **ritmo de aprendizaje** o **learning rate**. Así, conforme más alejada de cero se encuentre la taza de error respecto al crecimiento de un peso particular, mayor será la modificación que este deberá sufrir.

No obstante, no todos los pesos de una red por capas se deben modificar, ya que es necesario que los que involucran neuronas en el MPEE tengan siempre valor de 1 para no alterar la expresión del error. Sin embargo, se debe determinar la derivada del error respecto a los demás $ah + bh$ pesos. Para esto, con apoyo de la figura 5.6, se aprecia que para cualquier neurona v en la capa oculta o la capa de salida, la derivada parcial respecto al peso w_{uv} de una arista (u, v) estará dada por la regla de la cadena

$$\frac{\partial E}{\partial w_{uv}} = \frac{\partial E}{\partial x_v} \frac{\partial x_v}{\partial w_{uv}} \tag{5.6}$$

ya que por definición $x_v = \sum_{z \in N_v^-} o_z w_{zv}$, con lo que la parcial del error se puede reescribir como

$$\frac{\partial E}{\partial w_{uv}} = \frac{\partial E}{\partial x_v} o_u \tag{5.7}$$

y como se sabe que δ_v representa a la derivada parcial de la función de red respecto a x_v, se tiene

$$\frac{\partial E}{\partial w_{uv}} = o_u \delta_v \qquad \text{y} \qquad \Delta w_{uv} = -\lambda o_u \delta_v \tag{5.8}$$

Con estas expresiones será posible aplicar iterativamente el algoritmo de retropropagación, efectuando por cada iteración una modificación a los pesos por medio del descenso por gradiente, con lo que eventualmente se podrá obtener un aprendizaje acertado por parte de la red. Sin embargo, comúnmente se cuenta con más de un par de entrenamiento, por lo que en la siguiente sección se presentan algunas estrategias a desarrollar en este caso.

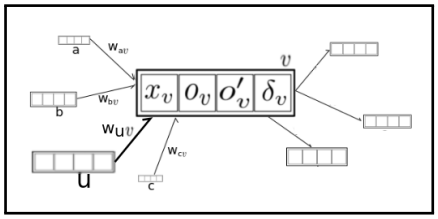

Figura 5.6: Modificacion sobre los pesos.

Métodos para el entrenamiento de una red neuronal por capas

Ya se ha visto como dado un solo par de entrenamiento, es posible modificar los pesos de una red por capas usando el descenso por gradiente y el algoritmo de retropropagación para que esta aprenda a reproducir a dicha pareja. Sin embargo, el problema ahora consiste en entrenar a la red para que aprenda a reconocer a un conjunto de pares de entrenamiento.

Para lograr esto, se debe considerar la manera en la que se hacen las actualizaciones a los pesos, respecto a cada iteración del algoritmo de retropropagación y respecto a como se desarrolla la lectura computacional del conjunto de datos de entrenamiento. Dependiendo de en qué momento y cómo se lleven a cabo estas modificaciones y lecturas, se tendrán en primera instancia dos posibles métodos para entrenar a una red neuronal por capas: descenso clásico por gradiente y descenso estocástico por gradiete.

- Descenso clásico por gradiente

El descenso clásico por gradiente, también llamado Off-line training o Batch traininig, consiste en obtener el incremento para los pesos sin aplicarlo a estos, por medio de una reproducción del algoritmo de retropropagación para todas y cada una de las pareja de entrenamiento. Una vez que se poseen todos los incrementos se podrá llevar a cabo las modificaciones de cada peso. De esta forma, para p parejas de entrenamiento y una iteración del algoritmo de retropropagación por cada pareja, se obtendrá para cada peso w_{uv} los incrementos

$$\{\Delta_1 w_{uv}^{(1)}, \Delta_2 w_{uv}^{(1)}, ..., \Delta_p w_{uv}^{(1)}\} \tag{5.9}$$

con los que el peso w_{uv} podrá ser actualizado por medio de

$$\Delta w_{uv}^{(1)} = \Delta_1 w_{uv}^{(1)} + \Delta_2 w_{uv}^{(1)} + ... + \Delta_p w_{uv}^{(1)} \tag{5.10}$$

Cada iteración del algoritmo de retropropagación junto con la actualización de los pesos es llamada una época. Así, los pesos podrán ser modificados en la i-ésima época por medio de $\Delta w_{uv}^{(i)}$.

- Descenso estocástico por gradiente

Es posible realizar la actualización de los pesos inmediatamente después de haber presentado un solo par de entrenamiento a la red, y repetir este proceso hasta concluir con todos los pares de entrenamiento disponibles. Esta manera de realizar las actualizaciones a los pesos es llamada On-line training o entrenamiento secuencial.

De esta forma, una época de entrenamiento estará compuesta por una iteración de la ejecución del algoritmo de retropropagación para un solo par y la actualización inmediata de los pesos, hasta acabar con todas las parejas de entrenamiento. Nuevamente, se podrá realizar más de una época, optimizando el aprendizaje de la red por cada una de estas.

Como descrito por Rojas [24], con este método las actualizaciones individuales de los pesos no siguen del todo a la dirección de máxima reducción del error. Sin embargo, al tomar aleatoriamente las parejas de entrenamiento por cada época de aprendizaje, se puede obtener en promedio una oscilación en torno a la dirección del negativo del gradiente del error, razón por la que este método recibe también el nombre de descenso estocástico por gradiente. Este aspecto representa una ventaja para el aprendizaje, ya que debido a la inicialización aleatoria de los pesos, la reducción del error no garantiza encontrar siempre un mínimo global para este, por lo que oscilar durante la reducción del error reduce el riesgo de caer en mínimos locales.

5.4 Metodología

En este trabajo se desarrolló una implementación de una red por capas en Python, usando solo la paquetería básica de este lenguaje. Posteriormente, dado que nuestra implementación requiere de mayor tiempo físico para ejecutarse en comparación con las redes ya implementadas en Python, específicamente las de la biblioteca SKLearn [32], se desarrolló un análisis de clasificación por medio de dicha biblioteca, sobre redes de regulación genética que modelan cáncer de mama. A continuación se describen las generalidades de nuestro programa, así como el proceso de obtención de dichas redes de regulación, y los experimentos definidos para su estudio.

Construcción de una red por capas con la paquetería básica de Python

Por medio de las definiciones presentadas en el marco teórico se desarrolló un programa en Python, que utiliza solo las estructuras de datos básicas de este lenguaje para construir y entrenar a una red neuronal por capas, que de manera general funciona como un regresor. El código se encuentra distribuido en cuatro scripts, cada uno de ellos se encarga de: revisar y preparar los datos, construir la red neuronal por capas, entrenar a este modelo, e imprimir los resultados del aprendizaje sobre un archivo de prueba.

En general, el programa recibe dos archivos de texto plano, uno que contiene a los pares de entrenamiento de la función que se busca aprender, y otro formado por datos de prueba que se reimprimen al finalizar el programa, adyacentes ahora a los datos predichos por la red para su comparación. Junto con este último se devuelve también un archivo PDF que contiene una imagen de la curva de aprendizaje de la red, donde esta muestra al error promedio obtenido sobre todo par de entrenamiento por cada época del aprendizaje.

Las neuronas de esta red tienen asociada una función de activación logística, a excepción del MPEE, y desarrolla su aprendizaje por medio del algoritmo de retropropagación junto con un descenso estocástico por gradiente. Para controlar el funcionamiento de este programa se pueden definir entre sus parámetros la cantidad de épocas que debe desarrollar sobre los datos de entrenamiento. Además, es posible modificar en él la cantidad de neuronas en la capa oculta de la red, mientras que el número de neuronas en las capas de entrada y salida quedarán determinadas por los pares de entrenamiento.

Para evaluar la funcionalidad de este código se implementó un ejercicio que es común en el estudio de las redes neuronales. Este consistió en entrenar a una red para poder reproducir la función booleana XOR [24], que recibe el valor de dos variables booleanas, y devuelve 1 cuando una sola de estas es 1, y 0 de cualquier otra forma. No obstante, el programa es capaz de recibir una función arbitraria, aunque la eficiencia de su entrenamiento dependerá de la complejidad de tal función, así como de la cantidad de datos que se proporcionen y de las iteraciones que se hagan sobre estos.

Tanto los cuatro scripts, así como el ejemplo de la función XOR, se pueden encontrar en el repositorio público: `https://github.com/MarcosLaffitte/FFANN_XOR`, creado para esta tesis, que contiene también instrucciones de como correr el programa. Para funcionar, este requiere solo de la paquetería básica de Python3.5, y muestra resultados en pantalla conforme su ejecución. Se advierte que únicamente se evaluó su correcto funcionamiento en sistemas Linux.

Dicha implementación fue de utilidad para corroborar y reforzar el conocimiento que se poseía sobre las redes neuronales artificiales. Sin embargo, este programa requiere de un alto tiempo físico de ejecución, en comparación con los modelos de redes neuronales ya existentes en la biblioteca SKLearn [32], también del lenguaje Python. Estos modelos son capaces de completar su entrenamiento en un menor tiempo físico, lo que permite entrenar a múltiples redes neuronales en el mismo tiempo en que nuestro programa entrenaría a una sola, facilitando así la evaluación del aprendizaje promedio que estas presentaron en nuestro análisis de redes de regulación genética.

Clasificación de redes de regulación genética

Motivados por el trabajo de Correira et. al [9], en este capítulo evaluamos la capacidad que tiene un sistema de redes neuronales artificiales, para distinguir redes de regulación genética relacionadas con cáncer de mama de entre redes generadas aleatoriamente, utilizando solo 14 propiedades de estas redes.

Para esto, se obtuvieron 48 redes de regulación genética, débilmente conexas y sin lazos ni múltiples aristas, a partir de una base de datos pública [37]. Posteriormente, para cada una de estas redes se formularon otras dos redes aleatorias, una de ellas preservando solo el orden y tamaño de su contraparte real, mientras que la otra preservó la secuencia de grado de la red que modela cáncer. La primera fue generada por medio del modelo *gnm_random_graph* de la biblioteca NetworkX, y la segunda a partir del modelo *directed_Havel_Hakimi_graph*, también existente en dicha biblioteca. Ambos modelos proporcionan digrafos sin lazos ni múltiples aristas, y además se utilizaron revisando que los grafos devueltos por ellos fueran débilmente conexos. Así, se obtuvieron en total 144 redes a clasificar.

Para desarrollar la clasificación se determinó un par de entrenamiento por cada red. Este consistió en una colección de propiedades que caracterizan a una red en su totalidad, y una variable de respuesta booleana con valor de 1 para redes de regulación genética, y 0 para aleatorias, independientemente del modelo utilizado para generarlas.

Estos pares de entrenamiento se construyeron utilizando 14 propiedades de redes, algunas de ellas utilizadas por Correira y colaboradores para su análisis, y evaluadas con apoyo de la biblioteca NetworkX [30]. Además del orden y el tamaño del grafo, se evaluaron algunas características asociadas al grafo subyacente de cada red, como: el diámetro, radio, tamaño del clique más grande y número de cliques maximales. Sin embargo, también se hizo uso de propiedades que toman en cuenta la dirección de las flechas, y fueron: densidad, grado promedio, coeficiente de agrupamiento promedio, la centralidad de alcance global y la transitividad. Estas últimas dos hablan respectivamente sobre la cantidad promedio de vértices asociados por caminos dirigidos, y la proporción de triángulos existentes en la red. Además, se emplearon algunas propiedades de detección de comunidades, también evaluadas sobre el grafo subyacente de cada red, estas fueron: modularidad, dependencia, y el total de comunidades de modularidad máxima, todas obtenidas por medio del algoritmo MODC como fue definido en el capítulo 4 de esta tesis.

El cálculo computacional de dichas propiedades se realizó por medio de NetworkX sobre un análisis en paralelo desarrollado con la biblioteca Multiprocessing. Esto se ejecutó en un tiempo favorable por medio de un servidor provisto por el Laboratorio de Visualización Científica (LAVIS) [41], del Instituto de Neurobiología de la UNAM, campus Juriquilla.

Por otro lado, para construir las redes neuronales artificiales se recurrió a la biblioteca SKLearn [75], utilizando específicamente la función diseñada para generar redes neuronales por capas para clasificación [76]. Todas estas redes se crearon con neuronas cuya función de activación es la función logística, además de entrenar con un descenso estocástico por gradiente.

Junto con estos, se utilizaron los valores predefinidos para todos los demás parámetros de estas redes, a excepción del parámetro llamado momento, que puede tomar valores entre 0 y 1, y para el que se determinó un valor de 0.1. Aunque el estudio de este último queda fuera del alcance de esta tesis, cabe señalar que este se modificó ya que permite reducir la probabilidad de que la red quede atrapada en mínimos locales, a cambio de prolongar el tiempo físico que consume cada época de aprendizaje [24].

Se desarrollaron dos experimentos para evaluar el aprendizaje de las redes bajo dos distintas condiciones. En un primer caso, se evaluaron redes con distintas cantidades de neuronas en la capa oculta, con valores $h = 5, 10, 15, 20, 25$, que aprendieron usando un 90% de las 144 parejas de entrenamiento.

Para el segundo experimento, se hizo variar el porcentaje de datos utilizados para el entrenamiento, incrementando este de 10 en 10 desde 50% hasta 90%, y dejando 5 neuronas en la capa oculta. En todos los casos, se generaron 500 redes neuronales artificiales, desarrollando para cada una de ellas sus correspondientes ensayos de entrenamiento.

Para evaluar el aprendizaje, se obtuvo el promedio de la medida conocida como **exactitud** de la predicción para cada grupo de 500 redes, calculada por medio de la función *accuracy_score* de la biblioteca SKLearn. Esta representa simplemente la proporción de redes correctamente clasificadas, tanto aleatorias como de regulación genética, sobre un cierto número de pares de prueba. En nuestro caso, esta se calculó sobre 15 parejas, tomadas de manera aleatoria de entre los 144 pares de datos disponibles.

Obtención de redes de regulación genética

Para nuestro análisis, se utilizaron 48 redes dirigidas, débilmente conexas, no pesadas y sin lazos ni multiaristas, de regulación genética en humano y relacionadas al cáncer de mama. Estas fueron obtenidas a partir de la base de datos The Cancer Network Galaxy [37], creada por la Universidad de Tokio.

En esta base de datos se menciona que las redes son inferidas por medio de una red bayesiana, aplicada a datos públicos de expresión genética de cáncer. Para cada flecha en estas redes se proporcionan distintos valores de confianza en la existencia de la interacción [77], de modo que uno puede filtrar aristas de ser necesario. Sin embargo, para nuestro análisis estas redes se tomaron tal y como se presentan en esta base de datos.

Estas 48 redes representan un subconjunto de todas las redes presentes en dicha base de datos obtenidas bajo la búsqueda por la frase *breast cancer*, tales que poseen no más de 500 vértices. Esto último se determinó así previendo que el cálculo de las propiedades de redes consumiría gran parte del tiempo para el trabajo en esta sección de la tesis.

Tanto estas redes y una lista de sus propiedades (fig. 5.7), así como los programas utilizados para analizarlas por medio de redes neuronales artificiales, se pueden encontrar en el repositorio público: `https://github.com/MarcosLaffitte/CancerDeMama`.

No.	Propiedad	Red Aleatoria	Red de Regulación Genética
1	Orden	424	424
2	Tamaño	4210	4210
3	Densidad	0.0235	0.0235
4	Grado promedio	9.9292	9.9292
5	Diámetro	4	4
6	Radio	3	3
7	Tamaño del clique más grande	7	8
8	Cantidad de cliques maximales	3466	2637
9	Alcance global	0.7076	0.1577
10	Coeficiente de agrupamiento promedio	0.0320	0.1236
11	Transitividad	0.0328	0.0947
12	Modularidad (MODC)	0.1425	0.1694
13	Dependencia (MODC)	0.0001	0.0001
14	Cantidad de comunidades (MODC)	204	178

Figura 5.7: Ejemplo de las propiedades evaluadas para una red aleatoria y una red de regulación genética usadas en nuestro estudio, ambas con el mismo orden y tamaño. Todas las propiedades están redondeadas a 4 decimales.

5.5 Resultados

A continuación se muestran los resultados obtenidos para el entrenamiento de nuestra red neuronal sobre la función XOR. Posteriormente se incluyen los correspondientes a la exactitud de la predicción obtenida por las redes neuronales de SKLearn sobre la clasificación de redes de regulación genética, tanto para el caso donde se hace variar la cantidad de neuronas en la capa oculta de estos modelos, como para el entrenamiento con diferentes porcentajes de pares de datos.

Resultados del aprendizaje de la función XOR

Durante la reducción del error, es común que un red neuronal encuentre secciones aproximadamente planas o de entrenamiento lento, cosa que pudimos reproducir en el entrenamiento de nuestra red neuronal sobre la función XOR (fig. 5.8) para 10000 épocas de entrenamiento y ritmo de aprendizaje de 1.

En el capítulo 7 de su trabajo [24], Rojas señala que estas secciones que presentan una pequeña pendiente, serían completamente planas si se intentara usar una función escalón o de Heaviside, como función de activación en vez de la función logística. De ser así, el error nunca convergiría y el entrenamiento continuaría indefinidamente.

Sin embargo, incluso usando la función logística como función de activación, este tipo de pendientes retrasan el entrenamiento, por lo que es recomendable siempre buscar estrategias que permitan contrarrestar este efecto sin favorecer la aparición de mínimos locales, de las que Rojas presenta algunas en su texto, incluido el uso del momento en el aprendizaje.

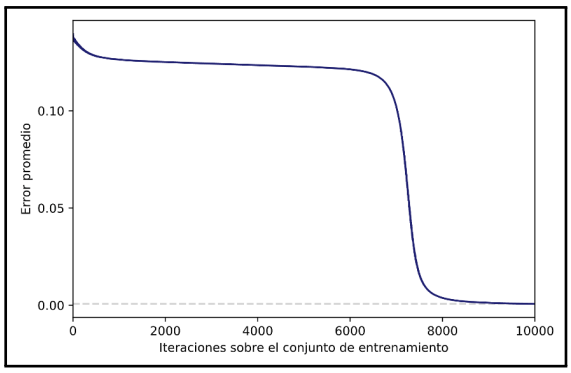

Figura 5.8: Curva de aprendizaje de XOR. Se obtuvo un error mínimo de 0.00069.

Resultados de la clasificación de redes de regulación genética

Por otro lado, en la figura 5.9 se muestra la exactitud promedio del aprendizaje obtenido para la clasificación de las redes de regulación genética, conforme se hace variar la cantidad de neuronas en la capa oculta, realizando siempre el entrenamiento sobre un 90% de los 144 pares disponibles. Para esta exactitud se obtuvieron valores entre 0.73 y 0.77, es decir, se encontró una correcta clasificación de aproximadamente el 75% de los datos de prueba.

Aunado a esto, se observa una correlación positiva entre la exactitud de la predicción y la cantidad de neuronas en la capa oculta, obteniendo un máximo cuando se tienen 25 de estas. Sin embargo, ya que la complejidad del entrenamiento crece conforme crece también la cantidad de pesos que son necesarios modificar, se considera que esta correlación puede llegar a variar de forma no lineal si se continúa incrementando la cantidad de neuronas en la capa oculta, lo que afectaría el aprendizaje en vez de favorecerlo.

Se debe mencionar también que al tomar en cuenta la desviación estándar de estas predicciones, se obtienen para todos los casos exactitudes que varían entre 0.72 y 0.8, efecto atribuido al hecho de que los pesos siempre se inicializan aleatoriamente en una red neuronal. Esta situación da razón de la necesidad de ejecutar una alta cantidad de redes neuronales para incrementar la confianza en las predicciones.

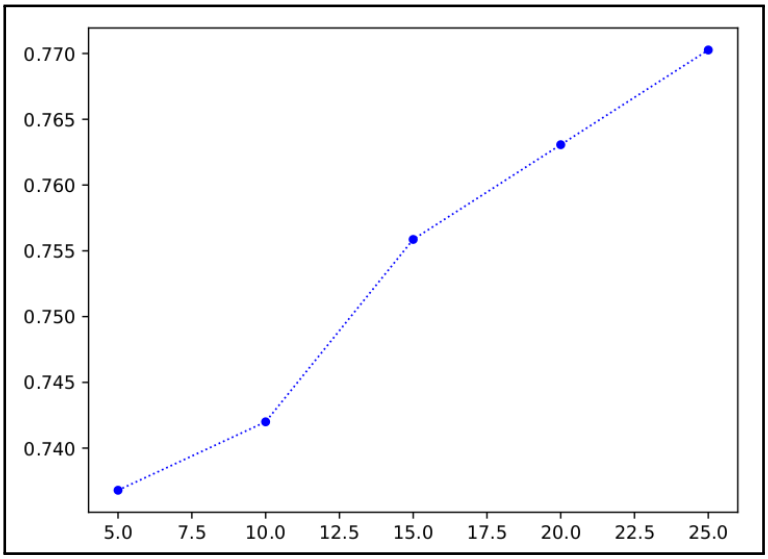

Figura 5.9: Exactitud en el eje y, contra cantidad de neuronas en la capa oculta en el eje x.

En la siguiente imagen se muestran los mismos resultados de la figura 5.9, pero con barras de error correspondientes a una desviación estándar respecto al promedio. Se aprecia que a pesar de existir un ligero incremento en la exactitud, en realidad existe la posibilidad de que una sola predicción aparente dar peores resultados conforme conforme crece la cantidad de neuronas. Debido a la magnitud de este error, se reitera la necesidad de realizar pruebas en un mayor número de redes neuronales. Así, se podría asegurar un correcto aprendizaje y predicción, evitando solo una predicción aleatoria.

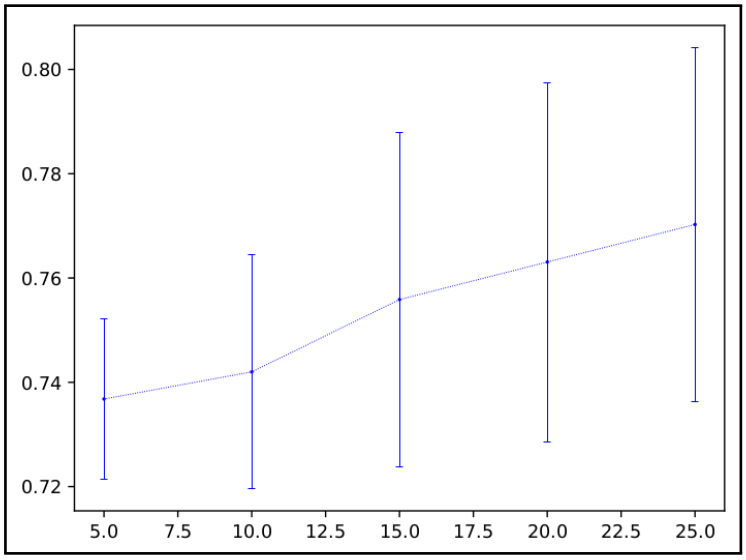

Figura 5.10: Exactitud al variar neuronas en la capa oculta, se incluyen barras de error.

Por otro lado, en la imagen 5.11 se muestran los valores de exactitud promedio obtenidos para la clasificación al hacer variar el porcentaje de datos de entrenamiento, usando 5 neuronas en la capa oculta para todos los casos. Para estas, se encontraron valores que van de 0.732 a 0.737, es decir, una correcta clasificación de cerca del 73% de los datos. Además, se aprecia también una correlación positiva entre el porcentaje de datos de entrenamiento y la exactitud.

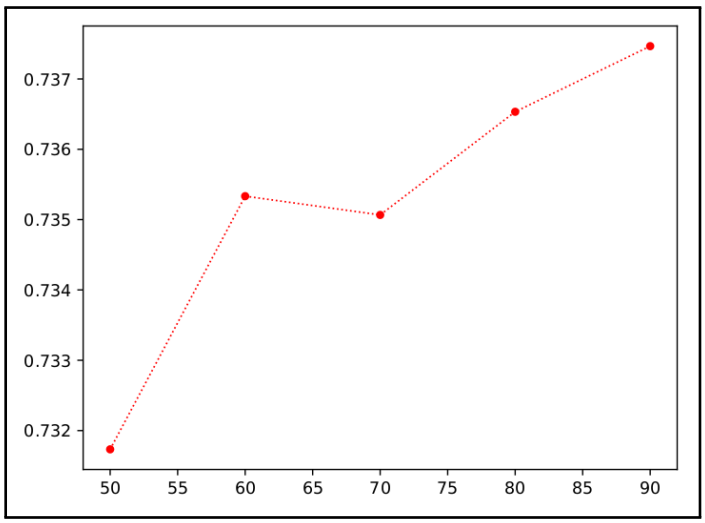

Figura 5.11: Exactitud en el eje y, contra porcentaje de pares de entrenamiento en el eje x.

Para este caso, al considerar la desviación estándar de los resultados (fig. 5.12), se obtienen exactitudes entre 0.68 y 0.78. De esta manera, se aprecia que el pequeño aumento en la exactitud que se tiene al incrementar el porcentaje de datos, se puede llegar a perder nuevamente debido a la inicialización aleatoria de los pesos. Sin embargo, es importante notar que la amplitud del error es mayor cuando se proporciona solo el 50% de los datos para el entrenamiento, reduciéndose para porcentajes mayores.

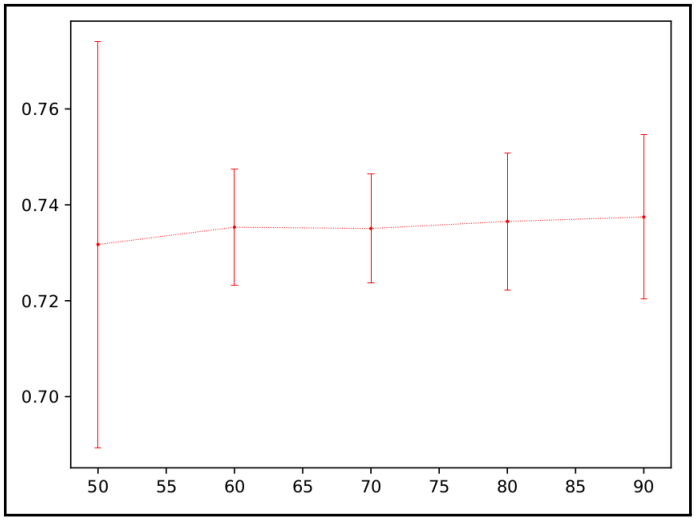

Figura 5.12: Exactitud al variar el porcentaje de datos de entrenamiento, se incluyen barras de error.

5.6 Discusión

Aunque exista un incremento en la capacidad de predicción conforme se aumenta el porcentaje de pares de entrenamiento, es necesario advertir que una mayor cantidad de datos no siempre conlleva un mejor aprendizaje. Esto se debe al efecto conocido como *overfitting*, que es una tendencia de las redes a imitar la información en su aprendizaje, en vez de extrapolar los patrones que detectaron en el entrenamiento, y para el que también se conocen estrategias que reducen sus efectos [7].

Debido a la gran magnitud del error que presentan las predicciones, es necesario hacer énfasis en la necesidad de realizar el entrenamiento de múltiples redes neuronales para cada tratamiento experimental. Además, se debe considerar que estas predicciones se obtuvieron utilizando solo 14 propiedades de redes, mientras que el trabajo de Correira [9] se desarrolló sobre 51 propiedades. De esta manera, es posible intuir que una mayor cantidad de propiedades pueden llegar a reducir el error en las predicciones.

Con clasificaciones cercanas al 75% de aciertos, consideramos como viable el estudio de esta metodología más a detalle. Aunado a esto, se considera favorable la obtención de mayores exactitudes conforme aumentan la cantidad de datos de entrenamiento y la cantidad de neuronas, ya que estas representan respectivamente una mayor flexibilidad en el aprendizaje y un entrenamiento más conciso.

Sin embargo, es necesario advertir que los resultados hasta ahora obtenidos no implican una capacidad para predecir cáncer, sino que simplemente permiten apreciar que existe la posibilidad de diferenciar redes que modelan sistemas reales de redes aleatorias, por lo que a continuación se recomiendan algunos experimentos que se pueden realizar si se desea dar continuación a esta linea de investigación.

5.7 Recomendaciones

Primeramente será necesario evaluar el aprendizaje de estas redes neuronales por medio de otras medidas de valoración, complementando la información que se puede obtener por medio de la exactitud de la predicción. Entre estas figuran las llamadas *sensibilidad* y *especificidad* de una predicción, así como su análisis por medio de la curva de la *carcaterística operativa del receptor*, de las que se puede encontrar más información en [7]. Para incrementar la confianza en los resultados, es necesario realizar pruebas de validación cruzada como descritas en el citado texto.

Además, será necesario explorar las alternativas a los efectos que tienen las ya mencionadas secciones de lento aprendizaje y el overfitting de la red, en particular si se desea desarrollar esta metodología sobre un mayor conjunto de redes de regulación, que tengan a su vez un mayor orden y tamaño que las redes aquí estudiadas.

Más aún, no solo se deben hacer modificaciones sobre las redes neuronales y su evaluación, sino también sobre los pares de entrenamiento. Para estos, se debe considerar que existen parejas de propiedades entre las que pueden existir correlaciones, de modo que, o proporcionan información redundante a la red, o pueden propiciar la confusión de una red de regulación con una aleatoria si es que estas propiedades tienen el mismo valor para ambas, como es el orden, tamaño y grado promedio de nuestras redes. Para evitar esta situación se debe realizar un análisis de selección de variables [7], donde al conocer la distribución de cada una de estas propiedades de red, se escogen las que proporcionen información más relevante de entre todas las propiedades disponibles, quedando así modificados los pares de entrenamiento.

Por otro lado, será necesario saber si estos sistemas de redes neuronales tienen la capacidad para distinguir redes de regulación genética que modelan cáncer de mama de entre otras redes con las mismas propiedades que modelen otros sistemas reales, en particular redes de regulación genética para otros procesos y enfermedades en el organismo del humano. Aunado a esto, se recomienda explorar la posibilidad de contar con más datos o redes de regulación de cáncer de mama, de las que se pueda obtener una mayor cantidad de pares de entrenamiento.

Inclusive sería favorable estudiar métodos de inferencia de redes por medio de datos de expresión genética, con los que se pudiera contar entonces con un protocolo para el diseño y construcción de un sistema de inferencia-clasificación de redes en cáncer.

Por último, se recomienda aprovechar las bibliotecas de aprendizaje automático del lenguaje Python, del cual es ejemplo la paquetería de SKLearn, así como las bibliotecas de análisis de redes, como fue el caso en esta tesis de la ampliamente señalada biblioteca NetworkX. Junto con esto, también se considera necesario implementar todo análisis posible por medio de la programación en paralelo, que se puede hacer por medio de la biblioteca Multiprocessing del mismo lenguaje.

Capítulo 6

Conclusión general

En el capítulo 3 se analizaron los efectos de la disminución de los nutrientes de los que disponen los organismos en redes de interacciones ecológicas, al relacionarlos con la cantidad de interacciones de competencia en ellas. Sin embargo, se notó que para poder extraer conclusiones concretas a partir de las propiedades de una red, siempre es necesario contar con una alta cantidad de ejemplares.

Por otro lado, en el capítulo 4 se desarrolló un análisis sobre el problema de detección de comunidades. Se aportó una formulación de la modularidad por medio de probabilidad condicional, y un método de detección de comunidades basado en dicha formulación. Se considera que los resultados fueron favorables, pero si se busca fortalecer dicho método, será necesario ampliar las hipótesis y las definiciones en las que se basa, completando la noción de comunidad.

Por último, en el capítulo 5 se plantearon las bases para una metodología de clasificación de redes de regulación genética que modelan cáncer de mama, lo que nos permitió explorar las aplicaciones del aprendizaje automático al tomar un enfoque sobre la detección temprana de dicha enfermedad. Se mostró que existe la posibilidad de distinguir redes de regulación en cáncer de entre redes aleatorias. Sin embargo, si se desea continuar con este proyecto, será necesario realizar esta clasificación contra redes reales que modelen otras rutas metabólicas en el organismo humano.

Para concluir, se debe remarcar que la importancia de las redes como modelos matemáticos reside en que permiten estudiar sistemas complejos, o que presentan propiedades emergentes explicadas por las interacciones entre los elementos del sistema [22]. Así, las redes en sistemas biológicos son herramientas que facilitan el estudio de la complejidad en diferentes niveles de organización de la materia viva.

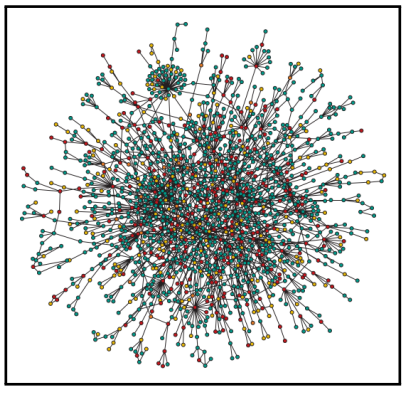

Figura 6.1: Red de interacciones entre proteínas de la bacteria *Saccharomyces cerevisiae*; sistema con una gran cantidad de elementos. *(Imagen tomada y adaptada de [25])*

Lista de figuras

Referencias

[1] M. E. J. Newman, *Networks, An Introduction*, pp. 168–220. Oxford University Press, 2010.

[2] Francesc Comellas et. al, *Matemática Discreta*, p. 335. UPC, 1 ed., 2001.

[3] Narsingh Deo, *Graph Theory with applications to Engineering and Computer Science*, p. 493. Prentice Hall, 1 ed., 1974.

[4] Frank Harary, *Graph Theory*, p. 285. Chapman and Hall, 1 ed., 1969.

[5] Campbell et.al, *Biology*, p. 1484. Pearson, 10 ed., 2014.

[6] Santo Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, pp. 75 – 174, 25 Jan 2010.

[7] K. Ramasubramanian and A. Singh, *Machine Learning using R, A Comprenhensive Guide to Machine Learning*, p. 580. Apress, 2017.

[8] Eric Kandel et. al, *Principles of Neural Science*, pp. 5–336. McGraw Hill, 5 ed., 2013.

[9] Correira et. al, "Prediction of cancer using network topological features," *Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies*, vol. BIOSTEC, pp. 207–215, 2016. https://dl.acm.org/citation.cfm?id=3093393.3093471.

[10] Cristina M. Prieto-Barajas et. al, "Microbial mat ecosystems: Structure types, functional diversity, and biotechnological application," *Electronic Journal of Biotechnology*, p. 9, November 2017. https://doi.org/10.1016/j.ejbt.2017.11.001.

[11] Valerie de Anda et. al, "Understanding the mechanisms behind the response to environmental perturbation in microbial mats: A metagenomic-network based approach," *Frontiers in Microbiology*, vol. 9, p. 24, November 2018. https://www.frontiersin.org/article/10.3389/fmicb.2018.02606.

[12] Shaw et al., "Metamis: a metagenomic microbial interaction simulator based on microbial community profiles," *BMC Bioinformatics*, vol. 17, no. 488, p. 12, November 2016. https://doi.org/10.1186/s12859-016-1359-0.

[13] Valeria Souza et. al, "An endangered oasis of aquatic microbial biodiversity in the chihuahuan desert," *PNAS*, vol. 103, no. 17, p. 6, April 2006. www.pnas.org/cgi/doi/10.1073/pnas.0601434103.

[14] M.E.J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 15, 2004. https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[15] Santo Fortunato and Marc Barthélemy, "Resolution limit in community detection," *PNAS*, vol. 104, no. 1, pp. 36–41, 2007. www.pnas.org/cgi/doi/10.1073/pnas.0605965104.

[16] M. Girvan and M.E.J. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, p. 7821–7826, 2002. www.pnas.org/cgi/doi/10.1073/pnas.122653799.

[17] Luis Rincón, *Introducción a la probabilidad*, p. 330. UNAM, 2013.

[18] Arenas et. al, "A complex network approach to the determination of functional groups in the neural system of c. elegans," *Springer, Berlin, Heidelberg*, vol. 5151, pp. 9–18, 2008. https://doi.org/10.1007/978-3-540-92191-2_2.

[19] Pavlovic et. al, "Stochastic blockmodeling of the modules and core of the caenorhabditis elegans connectome," *PLOS One*, vol. 9, no. 7, p. 16, 2014. https://doi.org/10.1371/journal.pone.0097584.

[20] White et. al, "The structure of the nervous system of the nematode caenorhabditis elegans," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 314, no. 1165, pp. 1–340, 1986. http://www.jstor.org/stable/2990196.

[21] Albert Einstein College of Medicine, Department of Neuroscience, "Wormatlas." `http://www.wormatlas.org/index.html`. Sitio visitado en la fecha: 07/12/2018.

[22] Melanie Mitchell, "Complex systems: Network thinking," *Science Direct - Artificial Intelligence*, vol. 170, pp. 1194 – 1212, 2006. https://doi.org/10.1016/j.artint.2006.10.002.

[23] American Cancer Society, "American Cancer Society Recommendations for the Early Detection of Breast Cancer." `https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection.html`. Sitio visitado en la fecha: 07/12/2018.

[24] Raúl Rojas, *Neural Networks, A Systematic Introduction*, pp. 3–180. Springer, 1 ed., 1996.

[25] T.A. Brown, *Genomes 3*, p. 736. Garland Science, 3 ed., 2002.

[26] Uri Alon, *Systems Biology*, p. 162. Taylor and Francis Group, 1 ed., 2007.

[27] Mark Newman, "Analysis of weighted networks," *Physical Reviews E*, vol. 70, no. 5, p. 9, 2004. https://link.aps.org/doi/10.1103/PhysRevE.70.056131.

[28] Robert A. Weinberg, *The biology of Cancer*, p. 963. Garland Science, 2 ed., 2014.

[29] Lancichinetti et. al, "Benchmark graphs for testing community detection algorithms," *Physical Reviews E*, vol. 78, no. 4, p. 5, 2008. https://link.aps.org/doi/10.1103/PhysRevE.78.046110.

[30] Aric et. al, "Exploring network structure, dynamics, and function using networkx." `https://networkx.github.io/documentation/networkx-1.10/overview.html`. Sitio visitado en la fecha: 07/12/2018.

[31] Zachary and Wayne, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977. www.jstor.org/stable/3629752.

[32] Petregosa et. al, "Scikit learn." `https://scikit-learn.org/stable/about.html`. Sitio visitado en la fecha: 07/12/2018.

[33] R. Milo, "Network motifs: Simple building bolcks of complex networks," *Science*, vol. 298, p. 5, October 2002.

[34] Pavel et. al, "Motif analysis in directed ordered networks and applications to food webs," *Nature Scientific Reports*, p. 9, July 2015.

[35] Uri Alon's Lab, "Mfinder - Network Motif Detection Tool." https://www.weizmann.ac.il/mcb/UriAlon/download/network-motif-software. Sitio visitado en la fecha: 07/12/2018.

[36] Clauset et al., "Finding community structure in very large networks," *Physical Reviews E*, vol. 70, no. 6, p. 6, 2004. https://link.aps.org/doi/10.1103/PhysRevE.70.066111.

[37] University of Tokio, The Institute of Medical Science and The Human Genome Center, "TCNG - The Cancer Network Galaxy 0.14." http://tcng.hgc.jp/index.html. Sitio visitado en la fecha: 07/12/2018.

[38] Hunter et. al, "Matplotlib." https://matplotlib.org/index.html. Sitio visitado en la fecha: 07/12/2018.

[39] NumPy developers, "Numpy." http://www.numpy.org/. Sitio visitado en la fecha: 07/12/2018.

[40] Fredrik Johansson, "Mpmath." http://mpmath.org/. Sitio visitado en la fecha: 07/12/2018.

[41] Instituto de Neurobiología - UNAM, "Lavis." http://lavis.unam.mx/. Sitio visitado en la fecha: 07/12/2018.

[42] Shai S., "Network motifs in the transcriptional regulation network of escherichia coli," *Nature Publishing Group - Advance Online Publication*, p. 5, 22 April 2002.

[43] Abou-Jaoudé et. al, "Logical modeling and dynamical analysis of cellular networks," *Frontiers in genetics*, vol. 7, p. 94, 2016. doi:10.3389/fgene.2016.00094.

[44] Kramer et. al, "Network inference with confidence from multivariate time series," *Phys. Rev. E*, vol. 79, p. 061916, Jun 2009.

[45] Mendenhall, *Introducción a la Probabilidad y Estadística*, p. 780. CENGAGE Learning, 13 ed., 2010.

[46] Sander et. al, "Ecological network inference from long-term presence-absence data," *Scientific Reports*, vol. 7, no. 1, 2017. https://doi.org/10.1038/s41598-017-07009-x.

[47] Cao et. al, "Inferring human microbial dynamics from temporal metagenomics data: Pitfalls and lessons," *BioEssays*, vol. 39.

[48] Angulo et. al, "Fundamental limitations of network reconstruction from temporal data," *Journal of The Royal Society Interface*, vol. 14.

[49] "Integrated Taxonomic Information System." https://www.itis.gov/. Sitio visitado en la fecha: 13/08/2018.

[50] Minot et al, "One Codex." https://onecodex.com/. Sitio visitado en la fecha: 13/06/2018.

[51] Thomas et. al, "Metagenomics - a guide from sampling to data analysis," *Microbial Informatics and Experimentation*, vol. 2, no. 3, p. 12, 2012. doi:10.1186/2042-5783-2-3.

[52] Vincent D. Blondel et. al, "Fast unfolding of communities in large networks," *Physics*, p. 12, July 2008.

[53] U. Alon, "Network motifs: theory and experimental approaches," *Nature Publishing Group*, vol. 8, pp. 450 – 461, June 2007.

[54] Thomas Aynaud , "Louvain community detection." https://github.com/taynaud/python-louvain. Sitio visitado en la fecha: 07/12/2018.

[55] Rivas-Marín et. al, "Evolutionary cell biology of division mode in the bacterial planctomycetes-verrucomicrobia- chlamydiae superphylum," *Frontiers in microbiology*, vol. 7, 1964. doi:10.3389/fmicb.2016.01964.

[56] Michael Sipser, *Introduction to the Theory of Computation*, p. 410. PWS Publishing Company, 1 ed., 1997.

[57] Yang et. al, "A comparative analysis of community detection algorithms on artificial networks," *Scientific Reports*, vol. 6, 2016. https://doi.org/10.1038/srep30750.

[58] Crossley et. al, "Cognitive relevance of the community structure of the human brain functional coactivation network," *Proceedings of the National Academy of Sciences*, vol. 110, no. 28, 2013. https://www.pnas.org/content/110/28/11583.

[59] Ronhovde et. al, "Local resolution-limit-free potts model for community detection," *Phys. Rev. E*, vol. 81, p. 15, 2010. https://link.aps.org/doi/10.1103/PhysRevE.81.046114.

[60] Traag et. al, "Narrow scope for resolution-limit-free community detection," *Phys. Rev. E*, vol. 84, p. 9, 2011. https://link.aps.org/doi/10.1103/PhysRevE.84.016114.

[61] Wolfram MathWorld, "Bell Number." `http://mathworld.wolfram.com/BellNumber.html`. Sitio visitado en la fecha: 13/06/2018.

[62] Albert Einstein College of Medicine, Department of Neuroscience, "Wormatlas - Introduction to C. Elegans Anatomy." `http://www.wormatlas.org/hermaphrodite/introduction/Introframeset.html`. Sitio visitado en la fecha: 07/12/2018.

[63] Albert Einstein College of Medicine, Department of Neuroscience, "Wormatlas - List of individual neurons." `http://www.wormatlas.org/neurons/Individual%20Neurons/Neuronframeset.html`. Sitio visitado en la fecha: 07/12/2018.

[64] Albert Einstein College of Medicine, Department of Neuroscience, "Wormatlas - Neuronal Wiring." `http://www.wormatlas.org/neuronalwiring.html#NeuronalconnectivityII`. Sitio visitado en la fecha: 07/12/2018.

[65] Aric et. al, "Lfr benchmark graph." `https://networkx.github.io/documentation/latest/reference/algorithms/generated/networkx.algorithms.community.community_generators.LFR_benchmark_graph.html`. Sitio visitado en la fecha: 07/12/2018.

[66] Python, "Multiprocessing - process-based parallelism." `https://docs.python.org/3.4/library/multiprocessing.html?highlight=process`. Sitio visitado en la fecha: 07/12/2018.

[67] Amelio and Pizzuti, "Is Normalized Mutual Information a Fair Measure for Comparing Community Detection Methods?," *International Conference on Advances in Social Networks Analysis and Mining*, p. 2, 2015. https://dl.acm.org/citation.cfm?doid=2808797.2809344.

[68] Kunio Doi, "Computer-aided diagnosis in medical imaging: Historical review, current status and future potential," *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society*, vol. 31, 2007. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1955762/.

[69] Kaymak et. al, "Breast cancer image classification using artificial neural networks," *Elsevier - Procedia Computer Science*, vol. 120, p. 6, 2017. https://www.sciencedirect.com/science/article/pii/S1877050917324298?via%3Dihub.

[70] Menéndez et. al, "Artificial neural networks applied to cancer detection in a breast screening programme," *Elsevier - Mathematical and Computer Modeling*, vol. 52, p. 9, 2010. https://www.sciencedirect.com/science/article/pii/S0895717710001378?via

[71] Grechkin et. al, "Identifying network perturbation in cancer," *PLoS - Computational Biology*, vol. 12, 2016. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004888.

[72] B. Lantz, *Machine Learning with R*, p. 396. Packt Publishing, 2013.

[73] Leonard Kaufman and Peter Rousseeuw, *Finding Groups in Data*, p. 355. Wiley-Interscience, 1 ed., 2005.

[74] Laura Igual and Santi Seguí, *Introduction to Data Science*, p. 227. Springer, 1 ed., 2017.

[75] SKLearn developers, "Neural Network Models (supervised)." `https://scikit-learn.org/stable/modules/neural_networks_supervised.html#neural-networks-supervised`. Sitio visitado en la fecha: 13/06/2018.

[76] SKLearn developers, "MLPClassifier." `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier`. Sitio visitado en la fecha: 13/06/2018.

[77] Tamada et.al, "Estimating genome-wide gene networks using nonparametric bayesian network models on massively parallel computers," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 683–697, 2011. 10.1109/TCBB.2010.68.