Capítulo 1

Introducción a las ciencias computacionales

1.1. Conceptos y fundamentos

La investigación científica se ve a menudo limitada por la capacidad del investigador para observar el fenómeno en estudio, o para analizar datos y realizar inferencias que confirmen sus hipótesis de trabajo. Sin embargo, a lo largo de la historia de la ciencia, el ser humano ha ideado formas para vencer esas limitaciones.

Por ejemplo, el genoma humano es conocido gracias al uso de computadores para procesar la enorme cantidad de información derivada de sus elementos constitutivos. El estudio de la estructura y el comportamiento molecular de agentes patógenos, mediante computadoras, ha permitido caracterizar algunas enfermedades y diseñar tratamientos personalizados efectivos. Nuevos materiales para la industria han sido diseñados con base en modelos y simulaciones computacionales de sus elementos a escala molecular. Los efectos del cambio climático son mejor comprendidos ahora gracias a la simulación del intercambio de carbono, que sería imposible estimar sin un computador.

Estos y otros ejemplos muestran cómo la integración de diversas disciplinas y las tecnologías computacionales permite extender los límites de la ciencia. Usamos el término *Ciencias Computacionales* para referirnos en forma conjunta a estas áreas convergentes, extendidas o *mejoradas* con el uso de herramientas computacionales. Pero la noción de las ciencias computacionales no considera solamente el uso de esas herramientas. La investigación de un fenómeno natural comienza con la construcción de modelos matemáticos, que luego son traducidos en programas eficientes para ser ejecutados en computadoras de alto rendimiento.

1.2. Motivación y Justificación

Las *ciencias computacionales* utilizan y necesitan del modelado y la simulación computacional. En términos sencillos podemos definir un *modelo* (científico) como una representación de un fenómeno, sistema o proceso en algún lenguaje formal (matemático), que describe y explica sus elementos característicos así como las relaciones entre ellos. La tarea de *modelado* requiere de la síntesis y representación del conocimiento de expertos y de las observaciones realizadas del fenómeno, para permitir cuantificar, visualizar y simular la dinámica y las propiedades de interés de un problema particular en un dominio de trabajo específico. Los productos del procesamiento de ese modelo pueden ayudar a caracterizar y comprender el fenómeno en estudio.

El modelo constituye, entonces, una explicación científica y permite simular el proceso o sistema que representa. Por su complejidad, usualmente se requiere de un computador para ejecutar estas simulaciones con alta precisión numérica en un tiempo razonable.

La *relevancia* científica de un modelo puede ser estimada a partir de varias métricas. Algunos ejemplos de estas son la *capacidad explicativa* de las relaciones causales en el fenómeno observado; el *nivel de isomorfismo* entre el objeto o fenómeno modelado y el modelo mismo; la *capacidad predictiva* para dar cuenta de instancias

futuras del fenómeno; y el *nivel de granularidad* o de detalle o generalidad en el cual el modelo describe el fenómeno de interés.

En este capítulo nos interesa considerar modelos *computacionales* de fenómenos o procesos naturales, es decir, modelos cuyo diseño considera las características y restricciones propias de una máquina computacional capaz de simularlos. En particular, el espacio (memoria) para representar el fenómeno, y el tiempo de procesamiento (cantidad de instrucciones).

Un modelo computacional debe satisfacer dos propiedades básicas. Primero, debe ser *efectivo*, es decir, proveer en efecto una solución a un problema dado. Segundo, debe ser *eficiente* en el uso de los recursos computacionales ya mencionados. La eficiencia es mayor en el tanto que el tiempo ocioso de los recursos computacionales utilizados es menor.

También interesa responder otras preguntas acerca del problema a modelar y del método a utilizar para resolverlo. Primero, es necesario determinar si el problema a modelar es *computable*, es decir, si es posible construir un modelo computacional efectivo. Segundo, interesa conocer la *complejidad* del problema, para de ahí estimar si será posible producir eficientemente soluciones al mismo. Tercero, es relevante saber si el modelo es *escalable*, esto es, si al incrementar el tamaño o la dimensionalidad del problema, el modelo sigue ayudando a resolver el problema en un tiempo razonable, o en otras palabras, si el método es estable al escalamiento del problema. Finalmente, es relevante determinar si el modelo es *tratable*, esto es, si es posible llevar a cabo la simulación con los recursos computacionales de tiempo y espacio disponibles.

En lo sucesivo nos referiremos a modelos computacionales *computables* y *tratables*.

1.2.1. Antecedente: preguntas fundamentales de la Ciencia de la Computación

La computación utiliza máquinas para resolver problemas. Una máquina computacional es una instancia de un sistema formal, es decir, la implementación de un modelo matemático en un medio computacional[28].

Una pregunta válida en este contexto es si para todo sistema formal existe una máquina computacional *equivalente*, esto es, con la misma capacidad de representación e inferencia. Es decir, que se desea determinar si es posible obtener respuestas de la máquina a cualquier pregunta acerca del *universo de discurso* del modelo.

En el dominio de conocimiento que nos ocupa, el de las ciencias computacionales, un modelo matemático describe un *proceso físico*. Una pregunta de alta relevancia relacionada con esta aseveración, es si se cumple la tesis de Turing, Church y Deutsch[15], que dice que una máquina computacional universal puede simular cualquier sistema físico. Si se cumple, el problema fundamental de las ciencias computacionales consiste en determinar qué máquinas computacionales pueden simular los procesos que interesa comprender.

Un caso ejemplar interesante de investigación científica computacional es el modelo de generación y propagación de potenciales de acción, de Hodgkin y Huxley [24]. Este modelo se origina en observaciones experimentales, y utiliza ecuaciones que describen las relaciones de cambio entre componentes. El nivel de especificidad hace necesario el uso de una computadora, aunque no contar con una en el año 1952 no fue un impedimento para los autores.

Un ejemplo más reciente es el descubrimiento del genoma humano, proyecto de más de una década, que utilizó los mejores supercomputadores de la época (1990-2003) para descubrir la estructura completa del genoma. Otro ejemplo es el enorme aumento en la capacidad de predicción del clima, a pesar de que la efectividad de los métodos actuales está todavía restringida a espacios geográficos muy reducidos y ventanas de predicción muy cortas, o sus predicciones son demasiado generales para tener alguna utilidad en zonas geográficas restringidas.

1.2.2. Hacia un cambio de paradigma de la investigación científica

La ciencia computacional es un área interdisciplinaria que se encuentra en la frontera entre las ciencias básicas, la ciencia de la computación y la matemática. El término *ciencia computacional* es amplio e involucra el desarrollo de sistemas, modelos, algoritmos, simulaciones y soluciones a problemas concretos de las ciencias.

En los últimos años, la investigación científica se ha orientado hacia el trabajo multi- e interdisciplinario; conforme avanza el conocimiento en cada campo, surge cada vez más la necesidad de interactuar con otras disciplinas en busca de teorías, métodos, técnicas y herramientas que complementen las de la disciplina propia. Un efecto colateral de ese proceso es la integración metodológica de las distintas disciplinas involucradas.

Esto hace que el trabajo colaborativo entre investigadores de distintas áreas se convierta en algo imprescindible. La ciencia de la computación provee a esos investigadores recursos y herramientas para la comunicación, el análisis, la representación, la administración y el procesamiento de datos, que hacen posible que los procesos de investigación sean más ágiles y productivos. Además de permitir procesar grandes volúmenes de datos para su análisis, la computación puede también contribuir a la creación y ejecución de experimentos y la elaboración de teorías a partir de los resultados del modelado computacional, la simulación y la visualización científica[45].

Algunos ejemplos de las clases de problemas en los que la computación puede contribuir incluyen problemas cuyo fenómeno de estudio es inaccesible o de difícil acceso, por ejemplo la dinámica de los sistemas planetarios o en general de la materia en el universo[4]; la función cerebral[38]; el modelado de las interacciones físicas y químicas en el nivel molecular[20]; los eventos geológicos y climáticos, y muchos otros procesos naturales.

Un concepto relevante para la tarea de modelado de tales procesos es el de *sistema caótico*. Un sistema es considerado caótico si el problema de predecir su comportamiento es complejo o intratable.[2]

En particular, un modelo computacional pueden volverse *intratable* cuando el sistema que representa es caótico. En particular interesa modelar problemas *complejos*, esto es, aquellos con muchos interesados en su solución pero con objetivos distintos, posiblemente conflictivos, con muchas variables por considerar, para los que parece nunca haber una solución óptima; por el contrario, la búsqueda de una mejor solución es un problema continuo y permanente; y finalmente problemas para los que la experimentación es sumamente costosa o del todo imposible.

Otro factor limitante de un modelo computacional de un sistema o proceso físico es el modelo matemático subyacente. Los sistemas continuos, por ejemplo los fluidos, son comúnmente modelados utilizando sistemas de ecuaciones diferenciales. Pero la solución computacional de un sistema de ecuaciones diferenciales puede ser muy demandante y costosa en tiempo de procesamiento.

El incremento en el uso de las computadoras para resolver esos tipos de problemas ha volcado la atención hacia el desarrollo de métodos numéricos que puedan hacer los cálculos necesarios de manera precisa y eficiente.

1.2.3. Caracterización de la ciencia moderna

La ciencia moderna es altamente inclusiva y transdisciplinaria. Los problemas abiertos en la actualidad pueden clasificarse en las escalas "muy grande" o "muy pequeño", como la astrofísica, la vulcanología, la ciencia de materiales o la neurociencia. Estas y otras áreas que requieren de mucho trabajo colaborativo se encuentran en las listas de prioridad de muchos centros de investigación en todo el mundo. Algunas de las características que describen este conjunto de disciplinas son las siguientes:

- Demandan conocimiento altamente especializado y de áreas relacionadas.
- Las soluciones se construyen en forma colaborativa.
- Utilizan métodos que integran otros de distintas disciplinas.
- La experimentación tradicional, por ejemplo, en un laboratorio en forma aislada, es compleja, ya sea por el alto costo financiero, el riesgo o la imposibilidad de acceso al objeto de estudio.
- Se benefician del modelado matemático y computacional.

El marco de trabajo de la investigación científica moderna reúne muchas disciplinas y enfoques, pero con particular énfasis se resalta la necesidad de crear modelos y realizar simulaciones computacionales en múltiples escalas, es decir, en múltiples niveles de *abstracción* de los procesos en estudio. En la sección se menciona un ejemplo de esto en el dominio de la química computacional.

1.3. Aplicaciones de las Ciencias Computacionales

1.3.1. Ciencias de la Computación

Las ciencias computacionales se caracterizan por incorporar métodos matemáticos y computacionales en la resolución de problemas. Esto hace imprescindible para los investigadores de las disciplinas de aplicación involucrarse en el proceso de modelado computacional.

La Ciencia de la Computación estudia los métodos formales para la resolución de problemas y sus propiedades computacionales. Esto incluye desde el estudio de máquinas, vistas éstas como sistemas formales, hasta su diseño, implementación y validación.

De acuerdo con la IEEE[1] y la ACM[2], la siguiente es la lista de áreas del conocimiento relacionadas con las Ciencias Computacionales, que deben integrar el programa educativo básico de las Ciencias de la Computación [37]:

- *Algoritmos y complejidad*: análisis y diseño de soluciones algorítmicas y sus propiedades computacionales.
- *Arquitectura y organización*: jerarquía de memoria de una computadora y su impacto en el diseño de programas.
- *Estructuras discretas*: estudio de sistemas no continuos esenciales para la teoría y el modelado computacional (grafos y conjuntos).
- *Sistemas inteligentes*: diseño de métodos capaces de realizar tareas en forma autónoma.
- *Redes y comunicaciones*: métodos que permiten establecer vínculos entre máquinas, por ejemplo para envío de datos.
- *Sistemas operativos*: estudio de la asignación óptima de recursos computacionales y el diseño de sistemas que realizan esta labor.
- *Computación paralela y distribuida*: estudio y diseño de sistemas capaces de realizar más de una instrucción por unidad de tiempo.
- *Lenguajes de programación*: involucra el conocimiento de métodos formales para el diseño y creación de programas, desde el nivel gramatical hasta el de compilación y ejecución.

Cabe destacar que la matemática es un área del conocimiento fundamental y transversal a la lista anterior. El currículo universitario de las ciencias de la computación es variado e incluye por lo general otros temas adicionales, como la ética y responsabilidad social, o la seguridad informática. La lista de arriba incluye solamente aquellas áreas del conocimiento que se consideran esenciales para el ejercicio de las ciencias computacionales. El investigador de cualquiera de estas ciencias debe incorporar, con suficiente profundidad, los temas de esa lista.

Para el diseño de modelos o métodos computacionales es importante considerar la complejidad de los algoritmos que los resuelven, ya que un modelo debe ser tratable, es decir, producir resultados en un tiempo y haciendo uso de espacio razonables. Las diferentes arquitecturas computacionales y los lenguajes de programación imponen restricciones físicas y lógicas de representación numérica y de procesamiento, que pueden producir errores como el *redondeo* o el *truncamiento*.

Por otro lado, muchas tareas pueden automatizarse mediante métodos de la *inteligencia artificial*, comúnmente utilizados para la resolución de problemas complejos en tareas como la clasificación, la búsqueda, o la optimización. Algunos buenos ejemplos de textos introductorios al campo de la inteligencia artificial pueden son [5], [10], y más recientemente [35]. En las dos últimas décadas se ha visto un crecimiento considerable en la cantidad de investigaciones de las ciencias computacionales en las que se utilizan métodos de la inteligencia artificial para proveer soluciones cualitativas o aproximadas a sus preguntas de investigación.

[1] Institute of Electrical and Electronic Engineering, http://www.ieee.org
[2] Association for Computing Machinery, http://www.acm.org

Adicionalmente, un conocimiento básico de sistemas operativos y de redes permite al científico computacional no sólo utilizar las máquinas en forma más eficiente, sino entender cómo funciona la asignación de recursos como memoria y tiempo de procesador, los cuales contribuyen al rendimiento de un programa. Finalmente, la mayoría de problemas relevantes en la actualidad son muy complejos y no pueden ser resueltos eficientemente en máquinas aisladas; se hace imprescindible el uso de la computación paralela para diseñar soluciones eficientes que permitan resolver problemas de forma más rápida.

Un científico computacional debe tener al menos un conocimiento de nivel instrumental de estos temas, para que se desempeñe con fluidez y sea capaz de proponer, justificar y trabajar en proyectos multi- y transdisciplinarios con fuertes componentes computacionales. Si bien la ciencia computacional es primordialmente *ciencia mejorada con recursos computacionales*, el impacto de las ciencias de la computación en la gama completa de disciplinas afines es tan alto que no puede ni debe ser ignorado.

1.3.2. Química Computacional

El uso de modelos matemáticos en la Química es importante para producir explicaciones a muchos de los procesos y estructuras complejas, propios del objeto de estudio de esta ciencia. El trabajo en esta área es tan relevante que recientemente se le otorgó el premio Nobel de Química a tres científicos por su trabajo en simulación para predecir procesos [1].

Como se mencionó en la sección anterior, la necesidad de crear modelos y realizar simulaciones computacionales en múltiples escalas es propia de las ciencias computacionales. Un ejemplo de esto es el premio Nobel de Química del año 2013[1], otorgado a los investigadores Arieh Warshel, Michael Levitt y Martin Karplus, "por el desarrollo de modelos multiescala para sistemas químicos complejos", y "porque han hecho posible el mapeo de los misteriosos caminos de la química utilizando computadoras"[1]. Su trabajo ha contribuido sustancialmente a mejorar los métodos de simulación y predicción de procesos químicos.

Si bien todavía se utilizan métodos no computacionales para realizar algunas de las tareas de recolección de datos, clasificación, análisis, visualización y optimización, cada vez más la química computacional es utilizada para sustituir esos métodos.

El reto en este contexto consiste en construir modelos teóricos que representen un fenómeno en el mayor nivel de *granularidad* posible, es decir, con el mayor detalle posible, pero lo suficientemente simples para ser reproducidos o simulados en un computador.

A continuación se ofrece una descripción general del área de química computacional, una descripción más detallada puede encontrarse en [11].

Uno de los problemas de la química más aptos para ser resueltos mediante métodos computacionales es el análisis de las propiedades de las moléculas. Conociendo esas propiedades se puede calcular, entre otras cosas:

- Optimizaciones geométricas.
- Distribuciones de carga.
- Estructuras de transición.
- Frecuencias.
- Superficies de energía potencial.
- Anclaje.
- Las constantes para reacciones químicas.
- Calor de las reacciones.

Para modelar el comportamiento y las propiedades de moléculas, la química computacional hace uso de la Mecánica Cuántica, y en particular, de la *Ecuación de Schrödinger*[43]

Para describir la estructura de una molécula se utiliza la *fórmula molecular*, que describe el número y los tipos de átomos presentes en una molécula, y los *enlaces* o *ligas* entre esos elementos.

Un problema interesante asociado a la estructura molecular consiste en predecir la forma en la que los átomos se conectan a la molécula, dada su posición relativa respecto de los demás. Las estructuras *óptimas* necesitan menos energía para formar las conexiones. El problema de predecir una estructura molecular en condiciones ideales puede ser computacionalmente sencillo. Sin embargo, una predicción similar para una estructura molecular compleja, bajo condiciones no ideales (por ejemplo, relacionadas con la energía, o la temperatura), puede resultar sumamente intensiva computacionalmente, al grado de hacer el problema intratable.

A pesar de este problema, resolver la tarea utilizando métodos computacionales es mucho más eficiente que realizar la predicción de las estructuras con métodos de laboratorio. Para que la solución sea tratable, la estrategia de la química computacional consiste en utilizar recursos de *computación de alto rendimiento*, consistentes de recursos computacionales distribuidos y de ejecución paralela. Dos ejemplos de servicios de química computacional en la web son `folding@home`[13] y `Rosetta@home`[14]. Estos proyectos utilizan la *computación en mallas* (redes de recursos de procesamiento de datos bajo demanda) para ejecutar procesos solicitados por sus usuarios.

Debido a que el cálculo de la estructura molecular es tan demandante de recursos computacionales, las estructuras calculadas son almacenadas en bases de datos especializadas, lo que ayuda a simplificar el trabajo de otros investigadores en el resto del mundo. Los servicios computacionales disponibles, como los ejemplos mencionados arriba, pueden ser utilizados en investigaciones para identificar propiedades de las moléculas inherentes a su estructura, o para diseñar nuevas moléculas o sustancias en el área de la *medicina personalizada*.

Los métodos de la química computacional se derivan principalmente de la teoría de la *computación cuántica*. Estos métodos se conocen como métodos *ab initio* y pueden procesarse utilizando solamente la ecuación de Schrödinger. Otros tipos de método necesitan parámetros de datos obtenidos empíricamente para poder utilizar los modelos matemáticos. Estos métodos se conocen como *semiempíricos*. Finalmente, se pueden también encontrar métodos de la *mecánica molecular*, que utilizan la física clásica para explicar la dinámica molecular.

En todos los casos los métodos buscan aproximar el modelo al fenómeno real sin pretensión de ser exactos, lo que, como se explicó anteriormente, es una característica de cualquier modelo. Los métodos de la mecánica molecular son los menos exigentes computacionalmente, mientras que los métodos *ab initio* son los más demandantes. Los primeros se utilizan para procesar sistemas moleculares muy grandes y los segundos para sistemas pequeños o de menor complejidad. Los métodos semiempíricos son útiles para dar cuenta de sistemas de tamaño intermedio, y requieren de una capacidad computacional mayor a los de la mecánica molecular, pero menor a los *ab initio*. Con más capacidad computacional es posible hacer cálculos más precisos, lo que hace apropiada la aplicación de máquinas y técnicas de la computación de alto rendimiento en la química computacional.

Para terminar, es importante notar la diversidad de aplicaciones de *software* desarrollados para la química computacional, en particular para la ejecución de métodos semiempíricos, para la modelación y el diseño de moléculas, para cálculos de química cuántica, para la física de estados sólidos, y para la mecánica molecular. Algunos de los más utilizados son Gaussian[25], GAMESS (General Atomic and Molecular Electronic Structure System)[34], MOPAC (Molecular Orbital PACkage)[49], Spartan[26] y Sybyl[9].

1.3.3. Física Computacional

La física computacional se ocupa de la aplicación, implementación, desarrollo y estudio de los métodos computacionales para la resolución de problemas de la Física. Los orígenes de esta disciplina no son claros, pero por cientos de años se han realizado cálculos de la dinámica del sistema solar y otros sistemas complejos modelados y estudiados teóricamente.

Uno de los ejemplos más recientes del desarrollo de la física computacional es el Gran Colisionador de Hadrones (LHC) construido por la Organización Europea para la Investigación Nuclear (CERN)[19], que genera un volumen aproximado de 30 Petabytes de datos anualmente. Para poder procesar toda esa información, existe la necesidad de contar con modelos y computadoras con enormes capacidades de procesamiento.

Tradicionalmente la física ha sido dividida en la física *experimental*, que observa y estudia los fenómenos que ocurren en el mundo real, y la física *teórica*, que utiliza métodos matemáticos y modelos simplificados para explicar lo que se ha observado experimentalmente, para poder realizar predicciones en experimentos

futuros. La física computacional combina los métodos de la física experimental y la teórica.[42].

La diversidad de las aplicaciones de la física computacional es enorme. Es una disciplina en crecimiento, y nuevas áreas fuera de sus fronteras comunes están siendo desarrolladas continuamente. La modelación y la simulación son dos de los métodos más utilizadas en esta área, que se basa principalmente en métodos numéricos y matemáticos para representar sistemas físicos del mundo real.

La meteorología, por ejemplo, es un campo de la física que se ha beneficiado mucho de esta metodología. Se han desarrollado herramientas de simulación climática que ayudan a hacer estudios y predicciones basadas en un modelo o conjunto de modelos que representan cada uno de los subsistemas que están involucrados en el complejo sistema meteorológico: sistemas de presión atmosférica, humedad, temperatura, entre muchos otros.

La atmósfera puede entenderse como el contenedor de un fluido. Desde esta perspectiva, es posible aplicar modelos de ecuaciones de dinámica de fluidos y de la termodinámica para predecir un estado futuro del fluido a partir de uno conocido previamente. Estos modelos se usan para hacer predicciones climáticas basadas en datos reales recolectados.

En esta área, el modelo de investigación y pronóstico meteorológico WRF (*Weather Research and Forecast Model* [18]) es utilizado en gran cantidad de aplicaciones meteorológicas, desde simulaciones locales con escalas de decenas de metros, hasta modelos globales con escalas de miles de kilómetros.

Otro ejemplo es la clase de modelos de simulación oceanográfica, como ROMS (*Regional Ocean Modeling System* [44]). ROMS es una combinación de algoritmos numéricos para realizar simulaciones oceánicas dinámicas y de alta resolución. Los modelos de circulación oceánica se basan usualmente en las ecuaciones de Boussinesq[6], en la hidrodinámica y los balances de masa.

1.3.4. Biología Computacional

La biología computacional se refiere al estudio de la vida a través de métodos computacionales. Esta disciplina tuvo su origen formalmente entre las décadas de los años 50 y 60[23] del siglo XX. Se dieron al menos tres eventos que marcaron su inicio:

- La base de datos de secuencias de aminoácidos en crecimiento proveía tanto una fuente de datos así como problemas interesantes que no podían ser resueltos sin una computadora (en un tiempo aceptable).
- La idea de que las macromoléculas acarrean información tomó fuerza como concepto central de la biología molecular. Esto permitió establecer un vinculo conceptual entre la biología molecular y las ciencias de la computación, en particular haciendo uso de la teoría de la información[3].
- La comunidad académica vio la llegada de los computadores digitales de "alta velocidad" que fueron desarrollados por los programas militares durante la segunda guerra mundial, y su aplicación en la modelación de procesos biológicos.

Luego del hallazgo de la estructura tridimensional de la mioglobina por John Kendrew en 1962, Margaret Dayoff, directora asociada del *National Biomedical Research Foundation* utilizó una serie de programas escritos en FORTRAN para determinar la secuencia de aminoácidos en proteínas[31]. Usando fragmentos de péptidos que se traslapan de la digestión parcial de una proteína, los programas de Dayoff calcularon todas las posibles secuencias que era consistentes con los datos, llegando a la secuencia correcta en unos minutos.

Todos estos datos sobre proteínas fueron utilizados para crear un atlas de proteínas, el *Atlas of Protein Sequence and Structure*, que más tarde, en 1983, se convertiría en el *Protein Information Resource*. Ambas bases de datos son extensivamente utilizados en investigación básica en biología.

A partir de ahí, se han hecho muchos descubrimientos biológicos utilizando la capacidad de las computadoras, en particular métodos de búsqueda por homología, modelación y visualización de proteínas, y alineado de secuencias, entre otros.

La Biología Computacional no se da exclusivamente en el nivel microscópico. Existen problemas en el nivel macroscópico que pueden ser enfrentados computacionalmente, como por ejemplo estudios sobre nichos ecológicos[46][52][29], investigación en neurociencia[21][7], y estudios de filogenia[30][47].

La Biología Computacional no debe confundirse con la Bioinformática, aunque ambos campos están estrechamente relacionados. La Bioinformática es una disciplina de la informática y la computación que

busca proveer herramientas para procesar, almacenar, categorizar y visualizar datos e información biológica. Se construyen algoritmos para resolver problemas biológicos pero el enfoque está en el método mismo y no tanto en la aplicación. El método debe ser validado por la comunidad científica en informática y computación, utilizando las métricas del campo.

Existe una gran cantidad de programas de computador para la investigación en la biología computacional, que cabe destacar, La mayoría esta disponible para alguna versión del Sistema Operativo GNU/Linux. Algunos ejemplos son:

- *Rosetta*[41]: paquete de software para la modelación y el análisis computacional de la estructura de proteínas. Este software ha permito avances importantes en el campo, como el diseño *de novo* de proteínas y de enzimas, el acoplamiento de ligandos y la predicción estructural de macromoléculas.
- *openModeller*[33]: ambiente para realizar experimentación de modelación de nichos ecológicos. Permite ejecutar tareas tales como el muestreo de puntos para proyectar modelos en diferentes ambientes, la lectura de ocurrencia de especies y de datos ambientales. Dispone de más de diez implementaciones de algoritmos como GARP, Maxent, ENFA y máquinas de soporte vectorial.
- *BLAST*[36]: herramienta para encontrar similitudes locales entre secuencias biológicas, sean estas de nucleótidos o aminoácidos. BLAST compara estas secuencias contra bases de datos (remotas o locales) y calcula estadísticamente los mejores emparejamientos. También se puede utilizar para inferir la relación funcional y evolutiva entre secuencias, así como identificar miembros de una misma familia de genes.
- *QIIME*[8]: paquete para la comparación y el análisis de comunidades microbianas. Permite al usuario realizar tareas tales como la elección de OTUs, designaciones taxonómicas, la contracción de arboles filogenéticos, entre otros.
- *PHYLIP*[16]: software para inferir árboles evolutivos. Dentro de los métodos implementados cuenta con matrices de distancias, el principio de parsimonia y verosimilitud.

Uno de los ejemplos concretos en el área de la Biología Computacional es la actividad del Centro para Biología Computacional (CCB) [51], localizado en la Universidad de California. El CCB mantiene uno de los archivos de imágenes cerebrales más grandes del mundo, así como sus meta datos asociados información genética e imágenes derivadas. Estos, y otros datos, deben ser procesados por computadoras con algoritmos diseñados especificamente para tratar grandes cantidades de datos. Dichos algoritmos son tambien desarrollados por el CCB.

1.4. Comentarios finales

Las ciencias computacionales son el resultado de satisfacer las necesidades de procesamiento de datos de las ciencias tradicionales con las herramientas de la computación de alto rendimiento.

El potencial de producción de conocimiento de estas ciencias es por el momento, limitado únicamente por los recursos computacionales disponibles: espacio de almacenamiento, y velocidad de procesamiento.

La aparición de estas nuevas formas de hacer ciencia hace necesario que el científico de cualquier disciplina incorpore a sus conocimientos los principios y métodos fundamentales de la ciencia de la computación, en particular para el análisis de la complejidad de los problemas en estudio, para la creación de modelos cada vez más finos y detallados de esos problemas, y para la implementación y puesta en marcha de métodos para resolver las tareas básicas de las ciencias computacionales: la simulación, la clasificación, la visualización y la optimización, que la ciencia necesita para responder a sus preguntas relevantes. Estas tareas son descritas en el capítulo siguiente: Taxonomía de las tareas de las ciencias computacionales.

Capítulo 2

Taxonomía de las tareas de las ciencias computacionales

2.1. Tareas

2.1.1. Introducción

En esta sección se exponen algunos de los tipos de tareas más comunes de las ciencias computacionales. La lista no pretende ser exhaustiva pero si dar un vistazo general a los aspectos metodológicos que son comunes en el área.

2.1.2. Tipos de tarea

Modelación

Un *modelo* es una representación simplificada de un fenómeno del mundo real que se conforma con el propósito de estudiar ese fenómeno. Esta simplificación es necesaria debido a que muchos de estos fenómenos son muy complejos para analizarlos en su totalidad y, debido a esto, es difícil predecir sus comportamientos en el futuro.

Una de las formas más comunes de modelar un fenómeno con fines científicos es con el uso de las matemáticas. Se parte del supuesto de que un fenómeno puede ser interpretado como una serie de elementos que se relacionan entre sí por medio de reglas (operaciones matemáticas) y que existen bases evidentes a partir de las cuales podemos sustentar estas relaciones (axiomas).

El modelo matemático es entonces la herramienta del científico para comprender el fenómeno desde un punto de vista e interpretación particular, por lo cual no debe confundirse con el fenómeno en sí. Su utilidad radica en que un modelo bien planteado puede ser utilizado para comprender mejor el fenómeno que se estudia y realizar predicciones que pueden ser luego corroboradas observando el fenómeno real.

Al ser el modelo una creación interpretativa de una o más personas, inspirada en un fenómeno real, este puede tener diversas características que lo colocan en una categoría en especial. Según [45] un modelo puede clasificarse en las siguientes categorías:

1. El modelo puede ser *probabilístico*, cuando existe algún componente de aleatoriedad, o puede ser *determinístico*, cuando los resultados siempre son los mismos dadas las mismas condiciones iniciales.

2. El modelo puede ser *estático*, cuando su definición no incluye o no necesita la variable de tiempo, o puede ser *dinámico*, cuando la variable tiempo es crucial para realizar predicciones sobre un fenómeno.

3. El modelo puede ser *continuo*, cuando el tiempo es representado como un fenónemo continuo, o *discreto*, cuando el tiempo es modelado en unidades discretas.

Un modelo puede encontrarse dentro de alguna o varias de esas categorías según su concepción y según el fenómeno que desea representar.

A grandes rasgos, según [45], los pasos para crear un modelo son los siguientes:

1. *Analizar el problema*: es necesario saber, en primera instancia, si el problema se puede modelar y, si lo es, si tiene sentido modelar el fenómeno para solucionar el problema. Este proceso involucra un alto grado de cuidado en especificar de forma precisa cuál va a ser el objetivo del modelo, sus elementos y sus características. La naturaleza matemática del modelo obliga a ser rigurosos en su definición, lo que es crucial para la comprensión del problema.

2. *Formular el modelo*: este paso puede variar según la naturaleza del problema, pero normalmente el proceso sigue pasos similares en la mayoría de los casos.

    a) *Recolectar datos*: la observación del fenómeno que se quiere modelar normalmente debe venir acompañada de una recolección de datos sistemática. Aquí el término *observación* tiene una connotación amplia, que no se limita a la observación con los sentidos humanos sino que involucra instrumentos y técnicas de medición y recolección de datos.

    b) *Simplificar y determinar variables*: para ayudar a resolver el problema, el modelo debe reflejar los aspectos considerados relevantes del fenómeno. También debe definir con precisión las relaciones entre las variables y determinar cuáles de ellas dependen del comportamiento de otras variables y cuáles son independientes. Finalmente, es importante determinar las unidades de medición de las variables del modelo.

    c) *Definir ecuaciones y funciones*: el modelo consiste finalmente de un conjunto de ecuaciones y funciones definidas sobre las variables identificadas en el paso anterior.

3. *Resolver el modelo*: en esta etapa el modelo debe ser implementado en un modelo computacional y puesto en ejecución. El resultado podrá ser variado: la generación de un conjunto de datos a partir una simulación, una visualización, etc. Este es uno de los pasos críticas del proceso de construcción de un modelo, pues debe asegurarse que la implementación computacional refleja fielmente el modelo matemático.

4. *Validación y verificación del modelo*: el proceso de validación del modelo nos dice si las soluciones efectivamente resuelven el problema que se propuso resolver, mientas que el de verificación nos dice si las soluciones que ofrece el modelo son correctas. Con base en esos resultados, se determina si el modelo debe ser refinado o extendido. Por ejemplo, un modelo de aproximación numérica puede ser verificado mediante una metodología analítica; si los resultados no coinciden se debe considerar la posibilidad de que la implementación computacional realice cálculos sin la precisión necesaria. De ser cierto, esto obliga a volver al segundo paso para revisar las variables del modelo, sus unidades, relaciones con otras variables, y las consideraciones computacionales con las cuales se hizo la implementación.

5. *Documentación y comunicación de resultados*: como en cualquier proyecto científico, el proceso debe ser adecuadamente documentado y presentado a un público que pueda evaluar el proceso y las conclusiones de la implementación y predicciones del modelo. Un modelo exitoso puede luego ser reutilizado en problemas similares, o más generales, para complementar el trabajo de otros investigadores.

6. *Mantenimiento del modelo*: el proceso de investigación científica es continuo y permanente, y debe ser siempre abierto al debate, la rectificación, y la inclusión de nuevas fuentes de información. Los datos o variables con base en los cuales se creó un modelo pueden luego ser reemplazados por otros datos más precisos, o variables más relevantes, que no fueron identificadas originalmente, ya sea por limitaciones de conocimiento o de naturaleza tecnológica. De la misma forma, ya que el modelo es una interpretación particular de un fenómeno, un investigador puede encontrar una interpretación alternativa del mismo fenómeno que se enfoque en otras variables y en otras metodologías de recolección de datos, y adaptar su modelo correspondientemente. Por otra parte puede ser que la implementación del modelo se vuelva obsoleta por un cambios en estándares de hardware o software, lo que implica que el modelo debe ser adaptado, o renovado, para las características de hardware y software actuales. El modelo es, de esta forma, una representación dinámica que debe adaptarse a circunstancias y contextos específicos.

Comúnmente un modelo matemático puede ser implementado en una computadora. Los modelos matemáticos pueden considerarse modelos computacionales o convertirse en modelos computacionales. Un modelo computacional, a diferencia de uno matemático, debe considerar las limitaciones de las máquinas computacionales como la representación *discreta* de datos, el espacio disponible de memoria, el tiempo de procesamiento, o la arquitectura de la máquina. Por ejemplo, la implementación computacional de un modelo matemático que describa un proceso continuo tendrá que comprometerse con un modelo de representación discreta del proceso y del tiempo.

Simulación

En muchos casos la realización de experimentos con medios físicos es muy compleja o imposible, por lo que el uso de simulaciones computacionales es la mejor opción.

El modelo matemático provee una formalización de lo que se desea representar del fenómeno, mientras que el modelo computacional provee una versión del modelo matemático que puede ser transcrita a un lenguaje de programación y puesta a funcionar en una computadora: una simulación.

Las características del modelo usualmente son transferibles a la simulación. Por ejemplo un modelo estático y probabilístico se traducirá a una simulación con características estáticas y probabilísticas. Usualmente las simulaciones agregan un elemento de aleatoriedad en las entradas o parámetros del modelo, lo que les permite generar cientos o miles de experimentos que ayuden a hacer los resultados más confiables.

Una simulación bien diseñada puede revelar al usuario aspectos del fenómeno que no había notado anteriormente y múltiples usuarios pueden llegar a tener diferentes tipos de revelaciones según su área de experiencia. Así, un experto en química puede descubrir algo muy distinto a lo que podría observar un experto en biología o en física en una misma simulación. Esto es una propiedad que comparten los procesos de simulación y visualización, y no es extraño ver casos en los que la simulación y la visualización se integran en un mismo modelo computacional.

Las simulaciones pueden tener muchos usos más allá de los resultados numéricos que produce o las revelaciones que promueve. En [48] podemos encontrar algunos ejemplos de uso de las simulaciones:

1. *Entrenamiento.* Las simulaciones pueden ser utilizadas para entrenar personas en múltiples áreas como la aviación y la operación de maquinaria peligrosa.

2. *Apoyo en el análisis estadístico.* Una vez validada una simulación, esta puede ser utilizada para probar múltiples entradas y salidas y validar predicciones estadísticas. Este es uno de los usos más comunes de las simulaciones en las ciencias.

3. *Guía de animaciones por computadora.* La simulación puede combinarse con la visualización para observar cómo una animación se comporta a partir de los parámetros de entrada y su configuración.

4. *Control de procesos en línea.* Para un proceso que se está llevando a cabo en un momento dado, es a veces necesario predecir su comportamiento en el futuro inmediato. En estos casos la simulación se utiliza en paralelo con el proceso y la predicción debe ser constantemente actualizada.

5. *Predicción de resultados.* La predicción de resultados puede ser incierta debido a la incertidumbre asociada al modelo y a su implementación, como en el caso de problemas complejos o caóticos.

6. *Prueba y evaluación de sistemas nuevos.* Para sistemas o fenómenos nuevos o de los cuales se conoce poco, la simulación puede ser utilizada para su prueba y evaluación.

7. *Apoyo en el análisis bajo incertidumbre del comportamiento de un sistema.* Cuando no es posible saber *a priori* cómo funciona un sistema, se puede utilizar la simulación para ganar conocimiento sobre el mismo sin enfocarse necesariamente en los resultados.

8. *Mejora de la enseñanza y la educación.* La popularidad de la modelación y de la simulación en la enseñanza va en aumento. En algunos casos el proceso de modelación puede utilizarse como metodología didáctica si se utiliza la simulación para calibrar modelos de fenómenos reales. La metodología obliga a los estudiantes a *trabajar con las manos* en un acercamiento más empírico a la comprensión del fenómeno.

No siempre realizar simulaciones es garantía de que se tendrán las soluciones deseadas o correctas, la complejidad de los fenómenos reales hace que las simulaciones sean confiables hasta cierto punto, por lo que es imprescindible tener presentes sus limitaciones. Según [45] algunas limitaciones de la simulación son:

1. *Toma mucho tiempo o es muy costosa.* Incluso cuando la experimentación con simulaciones salga más barata que la experimentación real el proceso de crear la simulación que puede empezar con todo el proceso de modelación puede resultar excesivamente costosa.

2. *Usualmente es imposible probar todas las alternativas*, dada la gran cantidad de variables y las múltiples combinaciones de sus valores. Para solventar esto usualmente se utilizan heurísticas que guían los valores que se utilizarán para las variables y las combinaciones que parezcan más relevantes o reales. Esto puede dar buenos resultados, pero no es posible garantizar que sean óptimos.

3. *Las conclusiones son inciertas.* Dado que las simulaciones integran múltiples elementos que pueden fallar o presentar errores, es necesario validar con conocimiento experto las conclusiones que se deriven de la simulación.

4. *No se dispone de datos para la verificación.* Si una simulación hace predicciones de fenómenos con los que no puede experimentar o recolectar datos directamente, el proceso de verificación del modelo puede verse limitado. significado o relevancia.

A pesar de estas limitaciones, la simulación es una herramienta que se ha vuelto indispensable en muchos procesos de investigación científica, y ha llegado a sustituir o complementar el proceso de experimentación tradicional en el método científico.

Clasificación

La clasificación es una tarea común para las personas, y se define como el proceso de asignar un conjunto de *tuplas* a una *categoría* o *clase* previamente definida. Cada tupla representa un objeto a clasificar, y es el conjunto de los atributos relevantes de ese objeto. La *dimensión* de una tupa es la cantidad de atributos que describe.

Los problemas de este tipo están presentes en prácticamente todos los campos del quehacer humano, pero dada la complejidad de la tarea o por las condiciones del entorno donde se debe desempeñar, la clasificación debe echar mano de la computación. Existe muchos ejemplos de esto en problemas de la Microbiología [40], el análisis de imágenes [22], o la visión por computadora [32], entre otros.

Formalmente, la tarea de clasificación se define mediante una función $f : x \to y$, que mapea una tupla x a una categoría o clase predefinida y. Esta función también se conoce como *modelo de clasificación* [50].

La clasificación es particularmente útil para crear modelos descriptivos o predictivos de un conjunto de datos:

- Modelo descriptivo: un herramienta explicativa para diferenciar diferentes tuplas de diferentes clases. Los modelos descriptivos explican qué atributos de los objetos los identifican como miembros de una clase.

- Modelo predictivo: un herramienta que permite asignar una tupla a una clase, ya sea con base basado en una descripción analítica (un modelo) o en una descripción empírica (basada en datos, observaciones) del objeto a clasificar.

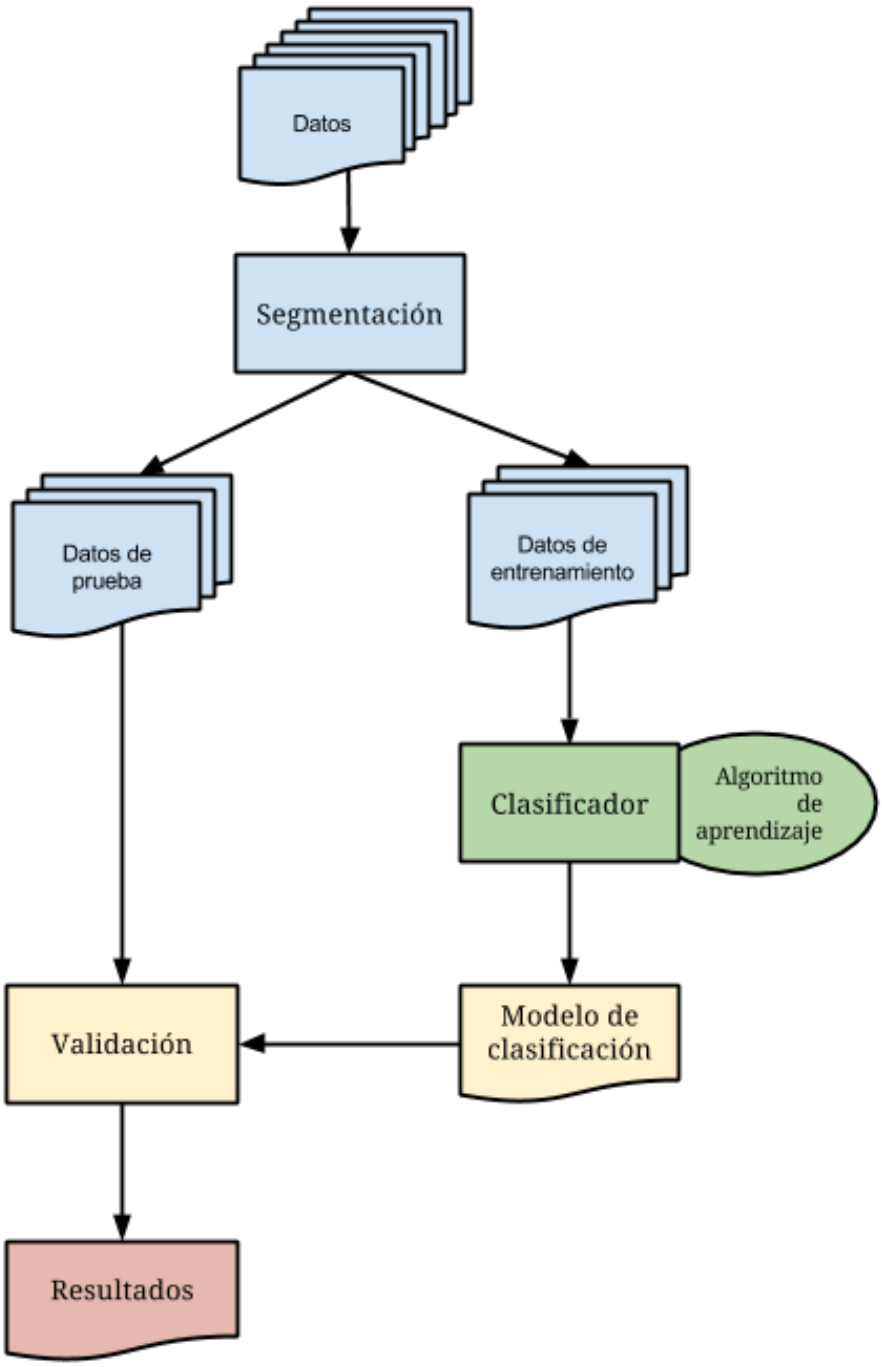

Figura 2.1: Modelo general de clasificación

Una *técnica de clasificación* o *clasificador* es una aproximación sistemática para la creación de modelos de clasificación a partir de datos de entrada. Existen muchos y diversos clasificadores que pueden ser utilizados en diferentes problemas de clasificación con resultados variables: árboles de decisión, modelos basados en reglas, redes neuronales artificiales, máquinas de soporte vectorial, modelos de inferencia bayesiana, de agrupamiento o de *clustering*, entre otros. Cada uno de ellos tiene características particulares que los hacen mejores o peores dado el problema a tratar. Dos de esas características son claves y están relacionan con los modelos

mencionados anteriormente: su *capacidad para ajustarse* a los datos existentes y su *capacidad para predecir* clases de tuplas que nunca había visto antes.

De forma general, los problemas de clasificación se pueden abordar de la siguiente manera:

1. Hacer una *recopilación* de los datos que se desean clasificar.
2. Realizar una *segmentación* de los datos en dos conjuntos: uno de *entrenamiento* y otro de *prueba*, por ejemplo asignando 60 % de los datos para el entrenamiento y 40 % para las pruebas.
3. Generar un modelo de clasificación mediante la *aplicación* del clasificador a los datos de entrenamiento. Es en esta etapa donde se observa la capacidad del modelo para ajustarse a los datos existentes.
4. *Evaluar* el modelo con los datos de prueba. Aquí se evalúa la capacidad del modelo para predecir la clasificación.

La figura 2.1 muestra un modelo general para abordar los problemas de clasificación.

La evaluación del *desempeño* del clasificador se puede basar en el porcentaje de datos clasificados correctamente. Estos se pueden representar en una tabla llamada *matriz de confusión*, como se muestra en la tabla 2.2. Existen también otras métricas que se utilizan para medir el desempeño de un clasificador, como la *tasa de error* y la *precisión*.

		Predicción		
		Clase 0	Clase 1	Clase 2
Realidad	Clase 0	30	5	5
	Clase 1	0	40	0
	Clase 2	15	0	25

Figura 2.2: Ejemplo de matriz de confusión

Muchas veces la escogencia de un clasificador depende de la configuración de los datos a procesar: sus dimensiones, tipos y tamaños, entre otros. A partir de estas características un clasificador puede ser mejor que otro [39].

Visualización

La visualización es la tarea de seleccionar, ordenar y presentar datos en una representación visual[53]. La presentación debe ayudar a caracterizar los datos y las relaciones entre ellos. El propósito de la visualización es ayudar para que la interpretación de la información presentada sea más rápida y clara, y que facilite su abstracción.

Una propiedad fundamental de toda visualización es su *adecuación*. La visualización en las ciencias computacionales se basda en diversas *metáforas* para el ordenamiento y presentación de la información, es decir, símiles con objetos o procesos naturales o de producción humana, que por su familiaridad facilitan la interpretación adecuada de los datos.

Un ejemplo del poder que puede tener la metáfora en la que se basa una visualización son los *mapas*. Considere el fragmento de datos de la figura 2.3, estos son tomas de temperatura de la superficie oceánica desde varios satélites orbitando el planeta. El conjunto de datos es mucho mayor de lo que muestra en este fragmento. Interpretar esta tabla puede ser difícil por la cantidad de datos.

VARIABLE : Analysis Temperature Deg. C
DATA SET : SST 100KM GLOBAL
FILENAME : CLASS_Descriptor.des
FILEPATH : /class_dirs/HTML/DocRoot/VisUser/www/product/23913
BAD FLAG : 999.9000244140625
SUBSET : 361 by 141 by 2 points (LONGITUDE-LATITUDE-TIME)
TIME : 03-AUG-2014 04:00:00 to 04-AUG-2014 04:00:00

	0E	1E	2E	3E	4E	5E	6E	7E
8E	9E	10E	11E	12E	13E	14E	15E	16E
17E	18E	19E	20E	21E	22E	23E	24E	25E
26E	27E	28E	29E	30E	31E	32E	33E	34E
35E	36E	37E	38E	39E	40E	41E	42E	43E
44E	45E	46E	47E	48E	49E	50E	51E	52E
53E	54E	55E	56E	57E	58E	59E	60E	61E
62E	63E	64E	65E	66E	67E	68E	69E	70E
71E	72E	73E	74E	75E	76E	77E	78E	79E
80E	81E	82E	83E	84E	85E	86E	87E	88E
89E	90E	91E	92E	93E	94E	95E	96E	97E
98E	99E	100E	101E	102E	103E	104E	105E	106E
107E	108E	109E	110E	111E	112E	113E	114E	115E
116E	117E	118E	119E	120E	121E	122E	123E	124E
125E	126E	127E	128E	129E	130E	131E	132E	133E
134E	135E	136E	137E	138E	139E	140E	141E	142E
143E	144E	145E	146E	147E	148E	149E	150E	151E
152E	153E	154E	155E	156E	157E	158E	159E	160E
161E	162E	163E	164E	165E	166E	167E	168E	169E
170E	171E	172E	173E	174E	175E	176E	177E	178E
179E	180E	179W	178W	177W	176W	175W	174W	173W
172W	171W	170W	169W	168W	167W	166W	165W	164W
163W	162W	161W	160W	159W	158W	157W	156W	155W
154W	153W	152W	151W	150W	149W	148W	147W	146W

Figura 2.3: Toma de datos satélites de temperatura de la superficie oceánica. Tomado de `http://www.class.ngdc.noaa.gov/saa/products/search?datatype_family=SST100`.

Ahora considere el mapa de temperaturas oceánicas 2.4 generado a partir de los datos de la figura 2.3. Este mapa abstrae de buena manera los datos de la tabla y los muestra en una representación gráfica que transmite de forma directa la información que contiene.

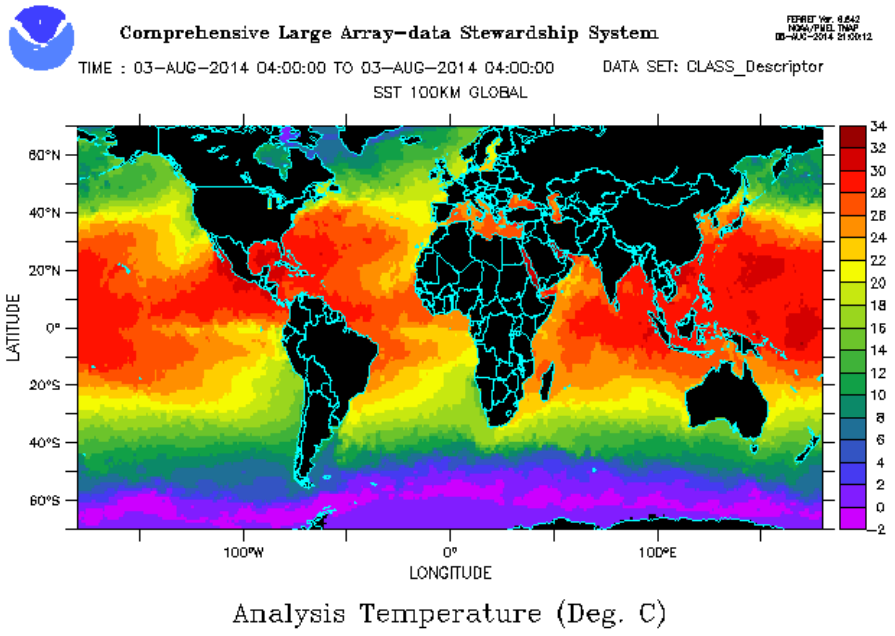

Figura 2.4: Mapa de temperatura superficial del océano. Tomado de `http://www.class.ngdc.noaa.gov/saa/products/search?datatype_family=SST100`.

Existen tres componentes de toda visualización que son críticos para su éxito: la selección de *datos relevantes*, el *mapeo* de datos *a elementos gráficos* y su *ordenamiento espacial*[50].

Si un conjunto de datos es numeroso, y cada dato es representado por una gran cantidad de atributos, su visualización puede resultar difícil. Para esto es necesario hacer una *selección*, ya sea eliminando o restando importancia a algunos objetos de la visualización. Para escoger un subconjunto de los atributos se utilizan técnicas de *reducción de la dimensionalidad*, como el análisis de componentes principales (PCA), la regresión o las redes neuronales artificiales[17]. Con los atributos más relevantes es posible producir una visualización con más significado. Por otro lado, cuando la cantidad de datos es muy alta, es posible que algunos sean obstruidos y ocultados por otros, lo que hace difícil su despliegue visual. En este tipo de situación es útil prescindir de algunos de estos datos, por ejemplo haciendo un muestreo o un acercamiento de los datos.

Los objetos a representar en la visualización deben ser transformados a elementos gráficos como puntos, líneas, colores o formas. Dependiendo del tipo de objeto a representar, se pueden utilizar varias estrategias. Por ejemplo, si se desea visualizar un solo atributo categórico de los objetos, éstos pueden ser agrupados en una categoría y ser desplegados como una entrada en una tabla o en una área especial de la pantalla, como se hace con los gráficos de barras. Si el objeto tiene varios atributos, estos pueden ser mostrados como una fila o columna en una tabla o una arista en un grafo. Es posible también que los objetos se presenten como puntos en un eje de coordenadas; estos puntos puede ser representados como formando figuras geométricas.

Los atributos de los objetos a visualizar pueden ser *nominales* (nombran o denotan un objeto), *ordinales* (hacen referencia a un objeto dentro de un conjunto ordenado), o continuos (se refieren a los valores de variables continuas del modelo).

Los atributos ordinales y continuos puede ser representados como características con orden, como puntos en ejes de coordenadas, intensidad, color, distancias de radio, anchura o altura. Por cierto, las variables *categóricas* del modelo también pueden ser representadas usando esa misma estrategia. Debe considerarse el caso de las variables nominales que no consideran un orden preestablecido, tales como lugares de nacimiento, o los nombres de los miembros de una población.

Las relaciones entre atributos también debe mostrarse gráficamente, ya sea de forma explícita o implícita. Por ejemplo, en un grafo las relaciones entre nodos se denota con una arista entre los mismos. Si se estuviera visualizando carreteras entre ciudades, y las ciudades fueran nodos, el ancho de la arista, la distancia entre los 2 nodos y el diámetro de los nodos pueden representar la afluencia de trafico, la distancia entre ciudades y la cantidad de población de las ciudades respectivamente.

En muchos casos las relaciones entre atributos u objetos se dan implícitamente. Si los objetos se representan como puntos en un eje de coordenadas en tres dimensiones, los puntos que se agrupan visualmente (sin necesidad de un componente gráfico entre ellos) muestran que los valores de sus atributos son similares.

Es difícil asegurar que las relaciones sean fácilmente observadas entre elementos gráficos; este es uno de los retos más grandes de las técnicas de visualizaciones.

La importancia del ordenamiento espacial de los elementos gráficos se puede mostrar por medio del siguiente ejemplo [50]: considere la figura 2.5 en la cual se muestra dos veces el mismo grafo. Del lado izquierdo se despliega una vista del grafo, opuesto la lado derecho, una vista diferente, que separa espacialmente los componentes conectados.

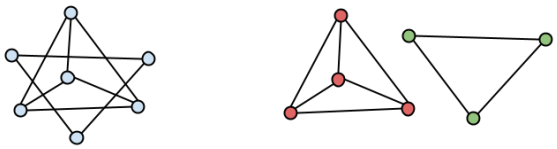

Figura 2.5: Interpretación visual de un grafo.

Utilizando esta líneas como bases es posible crear desde visualizaciones sencillas como gráficos circulares 2.6, diagramas de caja y bigotes 2.7 o histogramas 2.8, los cuales son muy utilizados para visualizar información estadística de una dimensión.

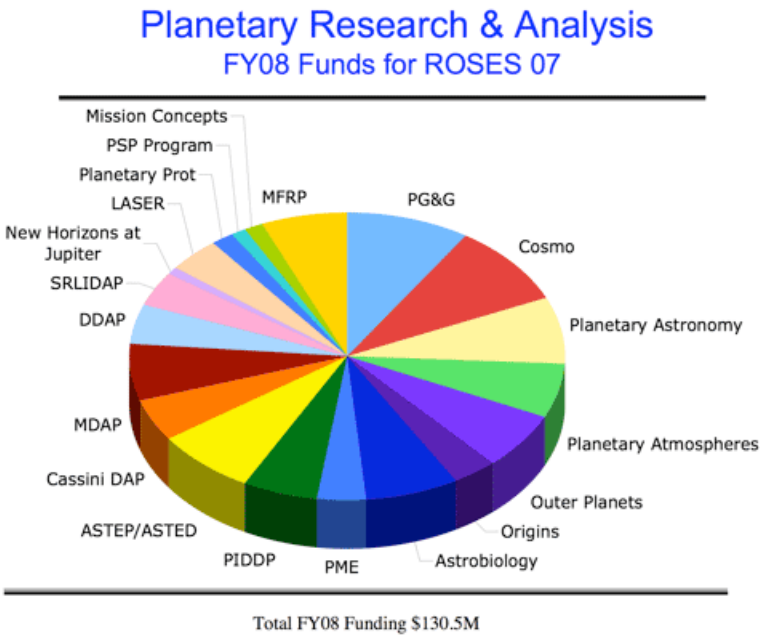

Figura 2.6: Gráfico circular: asignación de fondos para Investigación y Análisis Planetario 2007-2008, NASA, E.E.U.U. Tomando de `http://science1.nasa.gov/researchers/sara/division-corner/planetary-science-division-corner/`.

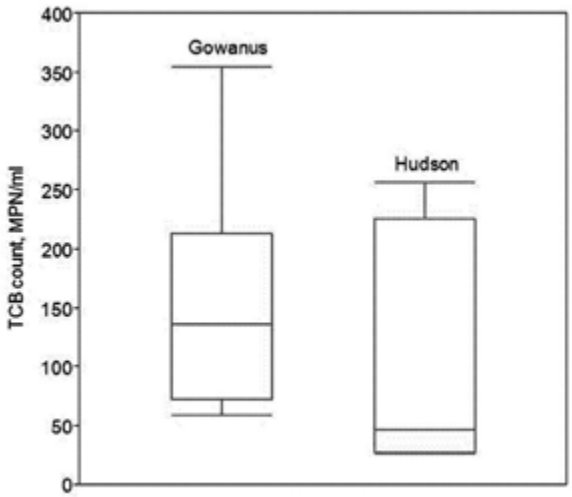

Figura 2.7: Diagrama de caja y bigotes: Total de bacterias cultivables en las aguas del río Hudson y el canal Gowanus. http://seceij.net/seceij/winter12/bio-math_mappin.html

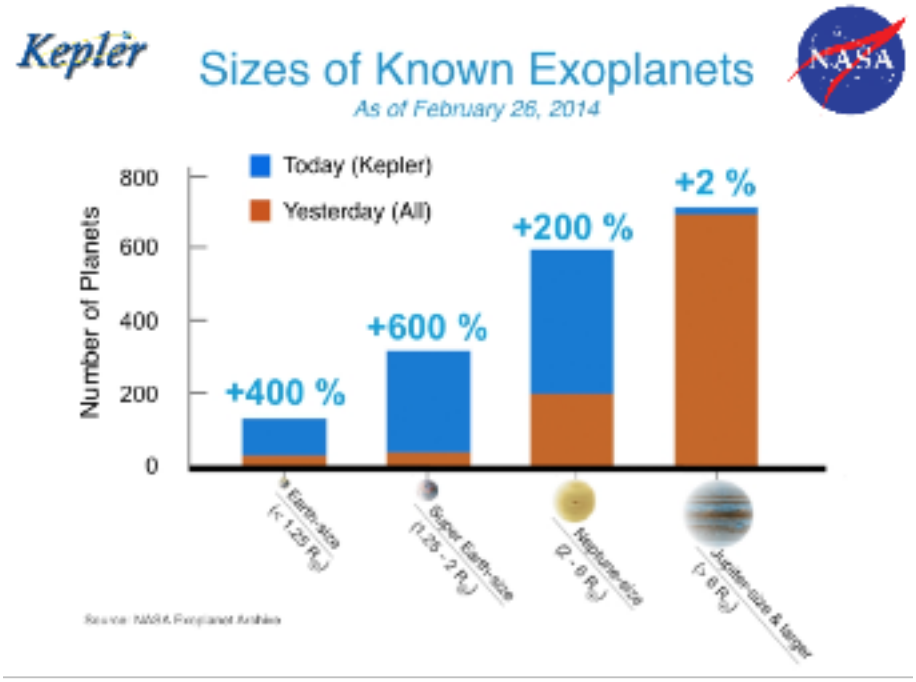

Figura 2.8: Histograma: tamaño de exoplanetas conocidos, descubiertos por la Misión Kepler de la NASA `http://www.nasa.gov/content/sizes-of-known-exoplanets`

.

Otras visualizaciones más elaborados como mapas de calor 2.9 o *Treemaps* 2.10 son utilizados para visualizar variables en dos dimensiones y son comunes en áreas como biología o informática.

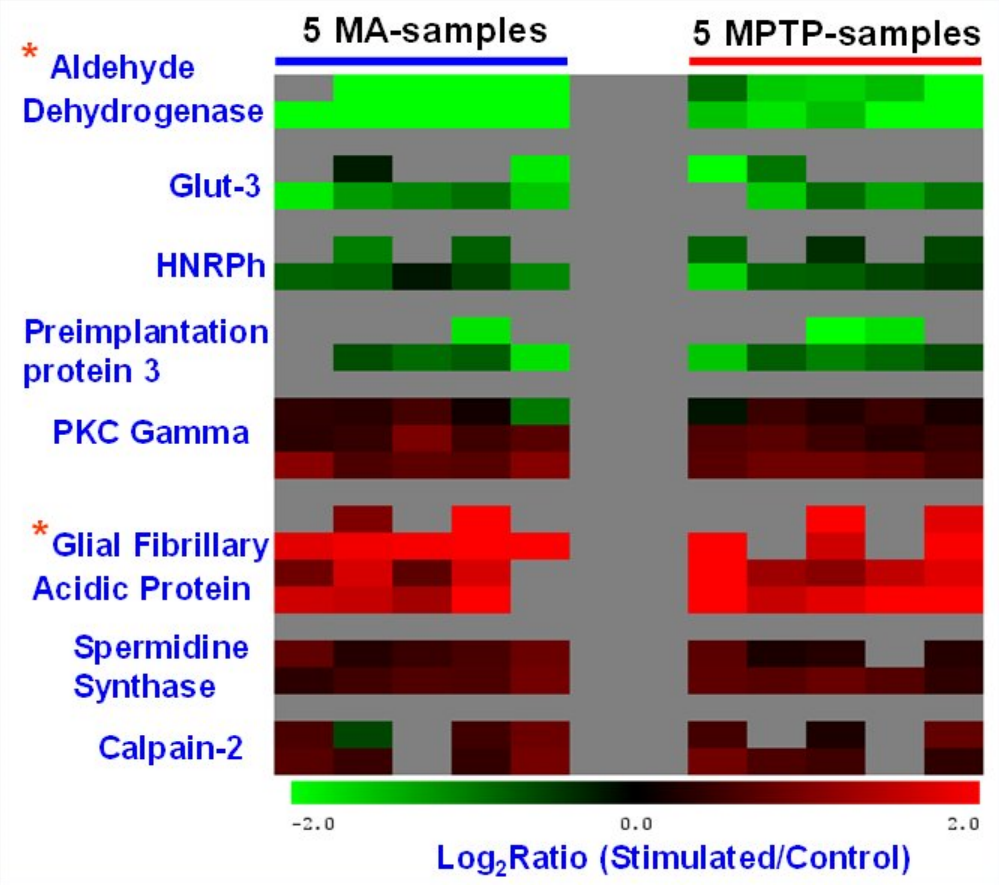

Figura 2.9: Mapa de calor: Péptidos utilizados en tratamientos de neurotoxinas en ratones de laboratorio `http://www.pnnl.gov/science/highlights/highlight.asp?id\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{6\global\mathchardef\accent@spacefactor\spacefactor}\accent226\egroup\spacefactor\accent@spacefactor64`

.

Visualizaciones mucho más complejas como animaciones, imágenes generadas por computadora (*render*) de superficies o volúmenes 2.11, o figuras en tres dimensiones 2.12 son muy llamativas y pueden comunicar mucha más información en un solo vistazo.

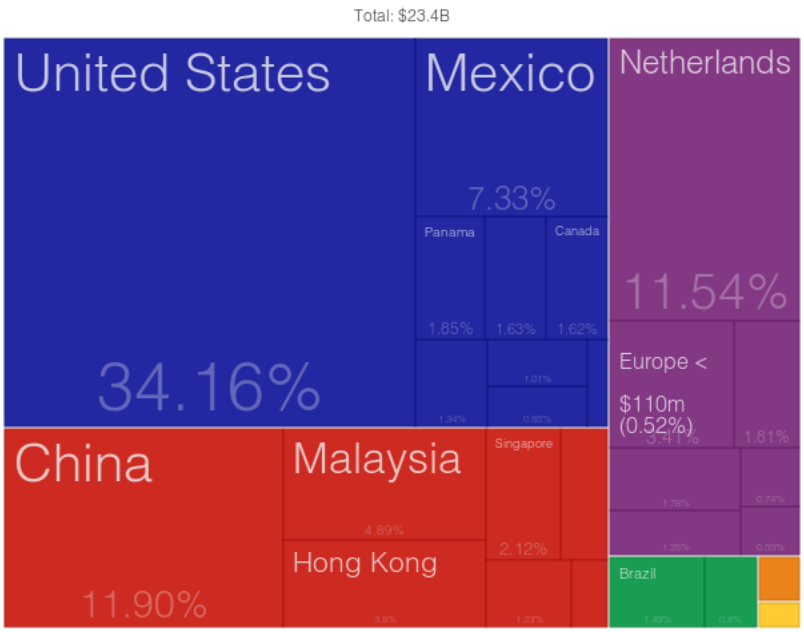

Figura 2.10: *Treemap*: exportaciones de chile en el 2010 `http://atlas.media.mit.edu/explore/tree_map/hs/export/chl/show/all/2010/`

.

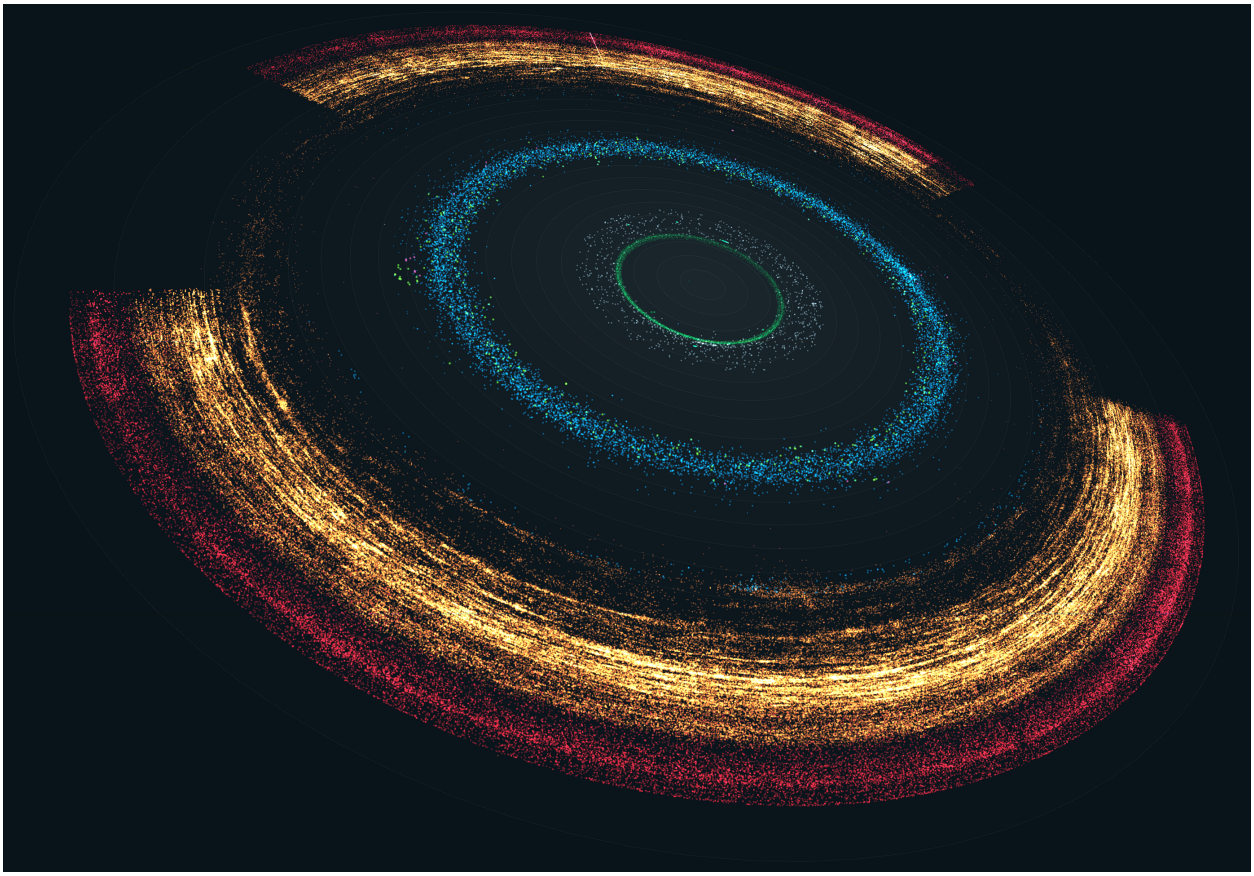

Figura 2.11: Mapa del universo conocido, cada punto representa un cuerpo astronómico identificado. Escala logarítmica `http://www.visualizing.org/galleries/ars-electronica-big-picture`

.

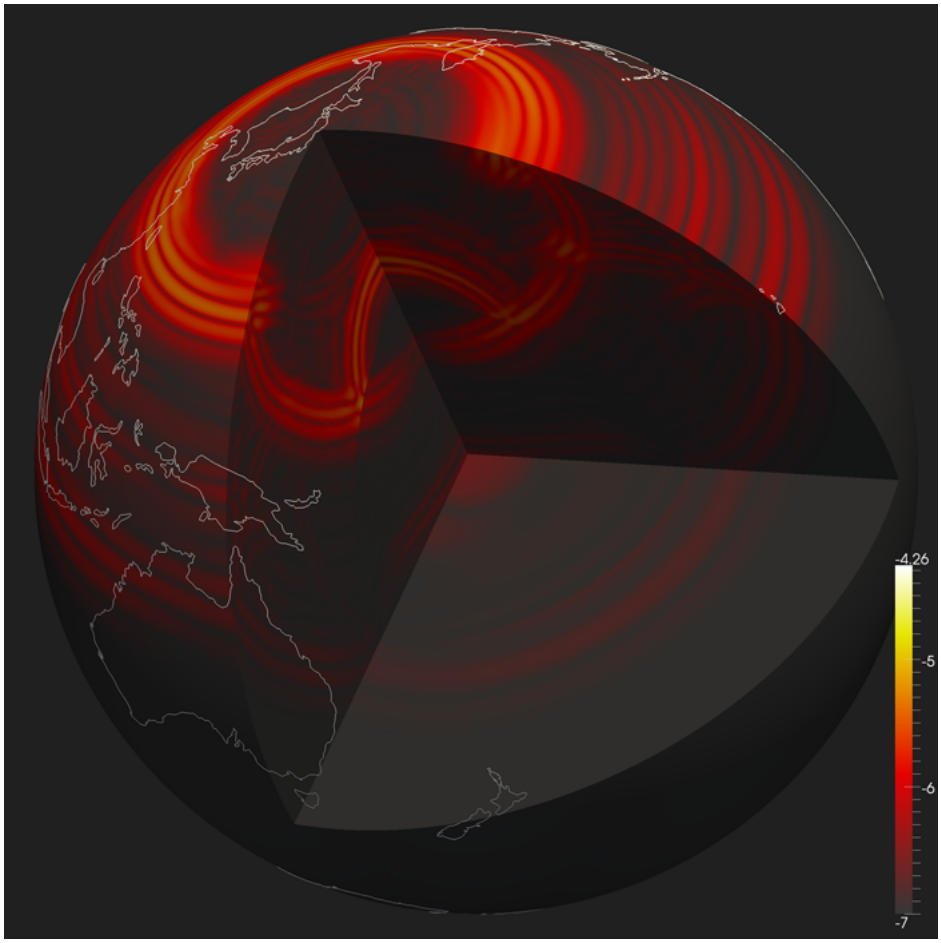

Figura 2.12: Generación de imagen por computadora en 3D: Modelo de propagación de ondas sísmicas en la Tierra `https://www.tacc.utexas.edu/scivis-gallery/seismic-wave`

.

Con frecuencia es necesario que la visualización de un proceso o problema no sea una imagen estática, sino una secuencia de ellas, o un vídeo. Esta forma de visualización genera un mayor impacto en el observador y puede ofrecer una visión más amplia del proceso o problema.

La creación de una animación o vídeo incluye la generación de series de imágenes y, cuando es posible, una pista de audio que acompaña a las mismas.

Dos ejemplos de animaciones son las producidos por el *Argonne National Laboratory*. La figura 2.13 muestra una captura de pantalla del estado intermedio de una simulación de la distribución de la materia en el universo, tomando en cuenta la influencia de la energía oscura. La figura 2.14 muestra una toma de pantalla del flujo de glóbulos en sangre, diferenciando los saludables de los enfermos.

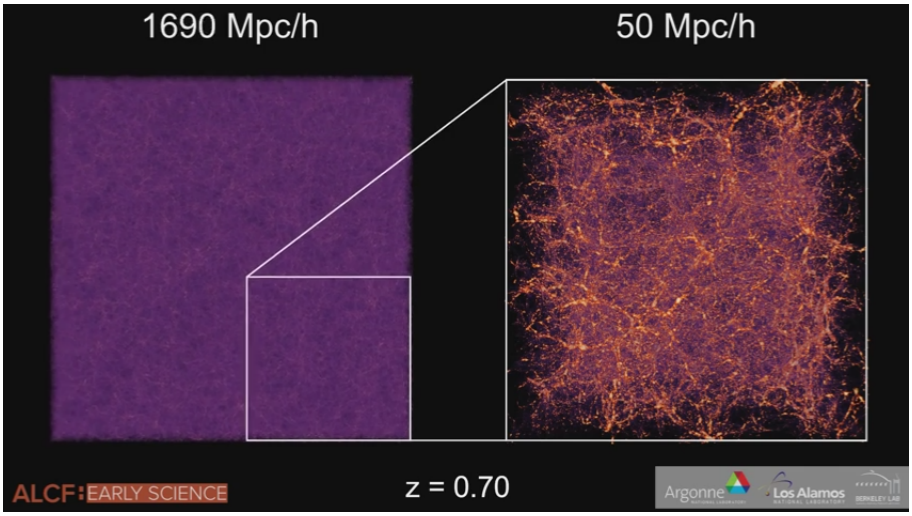

Figura 2.13: Secretos del universo oscuro: Simulando el cielo en Blue Gene/Q `http://www.youtube.com/watch?v=t-o7DU3W7kw`

.

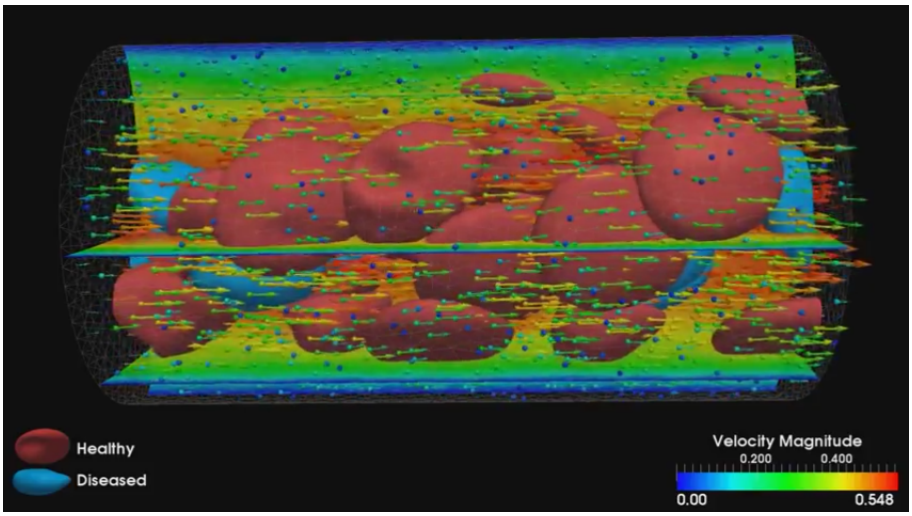

Figura 2.14: Flujo sanguíneo: modelación y visualización multi-escala `http://www.youtube.com/watch?v=0hibGZi8TWs`

.

Finalmente las visualizaciones científicas, deseablemente, deberían satisfacer los criterios expuestos en [27]:

- Enfoque científico.
- Representación del error y la incertidumbre.
- Interacción eficiente.
- Uso de puntos de vista globales y locales, según el contexto.

Optimización

Desde la perspectiva de la computación científica y de la Computación de Alto Rendimiento (CAR), se puede hablar de *optimización* del hardware y del software. Esta expresión se refiere al objetivo de mejorar la utilización de los recursos computacionales para *maximizar* alguna variable de interés en el proceso de modelación: mejoramiento de la precisión o de la velocidad en los cálculos, ahorro de energía, etc.

Estas estrategias de optimización incluyen el mejoramiento del rendimiento del código de un algoritmo mediante la incorporación de mejoras en el código fuente, o el uso de técnicas de computación paralela, o el uso de componentes de hardware con mayores capacidades, entre otras.

El objetivo de un problema de optimización es encontrar un conjunto de valores de entrada de una función que maximizan o minimizan su valor. Es decir, que dada la función

$$f : V \to R$$

se busca un valor x_i en V tal que $f(x_i) \geq f(x)$ para cualquier otro x, esto es, el máximo de la función (o, correspondientemente, se busca un valor x_i en V tal que $f(x_i) \leq f(x)$ para cualquier otro x, el mínimo de la función.

Esta definición general puede aplicarse en muchas áreas para diversos problemas. Dependiendo del contexto donde se aplique f, puede llamarse una función *objetivo* (cuyo valor es máximo para la mejor solución posible al problema(, una función *de costo* (cuyo valor es mínimo para la mejor solución), una función de *utilidad* (cuyo valor es máximo), una función de *aptitud* (también máximo) o una función de *energía* (para la que se busca el mínimo). [50].

Un problema de optimización puede tener múltiples objetivos. Por ejemplo, puede ser necesario buscar una función que calcule una reacción química que genere la mayor cantidad de calor, pero que sea estable en un ambiente determinado. Cualquiera de esas restricciones puede provocar que la otra cambie de forma contraria a lo que se busca, por lo que puede resultar necesario combinarlas para buscar la solución óptima. De la misma forma hay problemas de optimización para los que no existe una única solución, lo que hace necesario escoger una o unas pocas con base en otros criterios.

En general los métodos de optimización pueden clasificarse en aquellos *basados en el algoritmo Simplex*[12], en *algoritmos iterativos*, que buscan aproximar poco a poco una solución óptima y en ocasiones dependen de un criterio de convergencia de la solución, o en *algoritmos heurísticos*, que dependen de experiencias anteriores para guiar la búsqueda de una buena solución, si no es posible dar con la óptima.

Los problemas de optimización son muy frecuentes en áreas como la Ingeniería, la Economía, la toma de decisiones, y la modelación molecular.

Bibliografía

[1] Nobel Media AB. The nobel prize in chemistry 2013. `http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/`.

[2] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer, Nueva York, EE.UU., 1996.

[3] RB Ash. *Information Theory*. Interscience, Nueva York, EE.UU., 1965.

[4] James Binney and Scott Tremaine. *Galactic Dynamics*. Princeton Series in Astrophysics. Princeton University Press, 2000.

[5] Margaret A. Boden, editor. *The Philosophy of Artificial Intelligence*. Oxford University Press, 1990.

[6] J. Boussinesq. *Théorie de l'intumescence liquide, applelée onde solitaire ou de translation, se propageant dans un canal rectangulaire*. Comptes Rendus de l'Academie des Sciences 72: 755–759, 1871.

[7] James M. Bower. *20 Years of Computational Neuroscience*. Springer, 2013.

[8] J. Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D. Bushman, Elizabeth K. Costello, Noah Fierer, Antonio G. Pena, Julia K. Goodrich, Jeffrey I. Gordon, Gavin A. Huttley, Scott T. Kelley, Dan Knights, Jeremy E. Koenig, Ruth E. Ley, Catherine A. Lozupone, Daniel McDonald, Brian D. Muegge, Meg Pirrung, Jens Reeder, Joel R. Sevinsky, Peter J. Turnbaugh, William A. Walters, Jeremy Widmann, Tanya Yatsunenko, Jesse Zaneveld, and Rob Knight. QIIME allows analysis of high-throughput community sequencing data. *Nat Meth*, 7(5):335–336, May 2010.

[9] CERTARA. SYBYL: Molecular modeling from sequence through lead optimization. `https://www.certara.com/products/molmod/sybyl-x`.

[10] Eugene Charniak and Drew McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, Reading, Massachusetts, EE.UU., 1985.

[11] Christopher J. Cramer. *Essentials of Computational Chemistry: Theories and Models*. John Wiley & Sons Inc., 2003.

[12] George B. Dantzig and Mukund N. Thapa. *Linear programming 1: Introduction*. Springer-Verlag, 1997.

[13] Universidad de Stanford. Folding@home. `https://folding.stanford.edu`, 2014.

[14] Universidad de Washington. Rosetta@home. `https://boinc.bakerlab.org`, 2014.

[15] D. Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society (Londres) 400: 97–117*, 1985.

[16] J Felsenstein. Phylip: a free package of programs for inferring phylogenies. `http://evolution.genetics.washington.edu/phylip.html`.

[17] Imola Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National, 2002.

[18] National Center for Atmospheric Research (NCAR), National Centers for Environmental Prediction (NCEP), Forecast Systems Laboratory (FSL), Air Force Weather Agency (AFWA), the Naval Research Laboratory (NRL), University of Oklahoma (OU), and Federal Aviation Administration (FAA). WRF: The weather research and forecasting model. `http://www.wrf-model.org/`.

[19] European Organization for Nuclear Research. The worldwide lhc computing grid. `http://wlcg.web.cern.ch/`.

[20] Dann Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. 2a. edición. Academic Press, 2001.

[21] Karl J. Friston, Klaas Enno Stephan, Read Montague, and Raymond J. Dolan. Computational psychiatry: the brain as a phantastic organ. *The Lancet Psychiatry*, 1(2):148–158, 2014.

[22] Zhenhua Guo, Zhongcheng Zhang, Xiu Li, Qin Li, and Jane You. Texture classification by texton: Statistical versus binary. *PLoS ONE*, 9(2):1 – 13, 2014.

[23] Joel Hagel. The origins of bioinformatics. *Nature Reviews*, 1:231–236, dec 2000.

[24] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[25] Gaussian Inc. Gaussian 09. `http://www.gaussian.com`.

[26] Wavefunction Inc. Spartan: molecular modeling package. `https://www.wavefun.com/products/spartan.html`.

[27] Chris Johnson. Top scientific visualization research problems. *IEEE Comput. Graph. Appl.*, 24(4):13–17, July 2004.

[28] D. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, 1968.

[29] Honnavalli N. Kumara, Mohammed Irfan-Ullah, and Shanthala Kumar. Mapping potential distribution of slender loris subspecies in peninsular india. *Endangered Species Research*, 7:29–38, 2012.

[30] Bret Larget. The estimation of tree posterior probabilities using conditional clade probability distributions. *Systematic Biology*, 62(4):501–511, 2013.

[31] David Lipman. Margaret dayhoff and molecular evolution in the 21st century. In *7th Georgia Tech - Oak Ridge National Lab International Conference: Genome Biology and Bioinformatics*, 2009.

[32] Jing Lu, Jianwei Zhao, and Feilong Cao. Extended feed forward neural networks with random weights for face recognition. *Neurocomputing*, 136:96 – 102, 2014.

[33] M.E.S. Muñoz, R. Giovanni, M.F. Siqueira, T. Sutton, P. Brewer, R.S. Pereira, D.A.L. Canhos, and V.P. Canhos. openmodeller: a generic approach to species' potential distribution modelling. *GeoInformatica. DOI: 10.1007/s10707-009-0090-7*, 2009.

[34] M.W.Schmidt, K.K.Baldridge, J.A.Boatz, S.T.Elbert, M.S.Gordon, J.H.Jensen, S.Koseki, N.Matsunaga, K.A.Nguyen, S.Su, T.L.Windus, M.Dupuis, and J.A.Montgomery. General atomic and molecular electronic structure system. *Journal of Computational Chemistry, 14, 1347-1363*, 1993.

[35] Nils J. Nilsson. *The Quest for Artificial Intelligence*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[36] National Laboratory of Medicine. BLAST: Basic local alignment search tool. `https://blast.ncbi.nlm.nih.gov/Blast.cgi`.

[37] The Joint Task Force on Computing Curricula: Association for Computing Machinery (ACM) and IEEE Computer Society. Computer science curricula 2013. *Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, 2013.

[38] Randall C. O'Reilly and Yuko Munakata. *Computational Explorations in Cognitive Neuroscience. Understanding the Mind by Simulating the Brain.* The MIT Press, Cambridge, Massachusetts, EE.UU., 2000.

[39] Fabrice Pierre and Robert Colas. *Data mining scenarios for the discovery of subtypes and the comparison of algorithms.* PhD thesis, Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University, 2009.

[40] Jonathan Quiton, Claire Rinehart, Joseph Chavarria-Smith, and Nancy Rice. Gene classification for microarray data with multiple time measurements. *BMC Bioinformatics*, 9(7), 2008.

[41] Rosetta. Chemistry toolkit rosetta. `https://www.rosettacommons.org`.

[42] P.O.J. Scherer. *Computational Physics: Simulation of Classical and Quantum Systems.* SpringerLink: Springer e-Books. Springer, 2010.

[43] Erwin Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Physics Review 28, 1049*, 1926.

[44] Alexander F. Shchepetkin and James C. McWilliams. The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9:347–404, 2005.

[45] Angela B. Shiflet and George W. Shiflet. *Introduction to computational science.* Princeton University Press, 2006.

[46] Daniel P. Silva, Victor H Gonzalez, Gabriel A.R. Meloe, Mariano Lucia, Leopoldo J. Alvarez, and Paulo De Marco Jr. Seeking the flowers for the bees: Integrating biotic interactions into niche models to assess the distribution of the exotic bee species lithurgus huberi in south america. *Ecological Modelling*, 273:200–209, Febrero 2014.

[47] Mark P. Simmons. Misleading results of likelihood-based phylogenetic analyses in the presence of missing data. *Cladistics*, 28(2):208–222, 2012.

[48] John A. Sokolowski and Catherine M. Banks. *Principles of Modeling and Simulation: A Multidisciplinary Approach.* John Wiley & Sons Inc., 2009.

[49] James Stewart. MOPAC: Molecular Orbital PACkage. `http://openmopac.net`.

[50] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining.* Pearson Education Inc., 2006.

[51] Arthur W. Toga, Ivo D. Dinov, Paul M. Thompson, Roger P. Woods, John D. Van Horn, David W. Shattuck, and Douglas Stott Parker Jr. The center for computational biology: resources, achievements, and challenges. *Journal of the American Medical Informatics Association*, 19(2):202–206, 2012.

[52] Tiago S. Vasconcelos, Miguel Á Rodríguez, and Bradford A. Hawkins. Species distribution modelling as a macroecological tool: a case study using new world amphibians. *Ecography*, 35(6):539–548, 2012.

[53] Benjamin Vrolijk. *Interactive visualisation techniques for large time-dependent data sets.* PhD thesis, Technische Universiteit Delft, 2007.