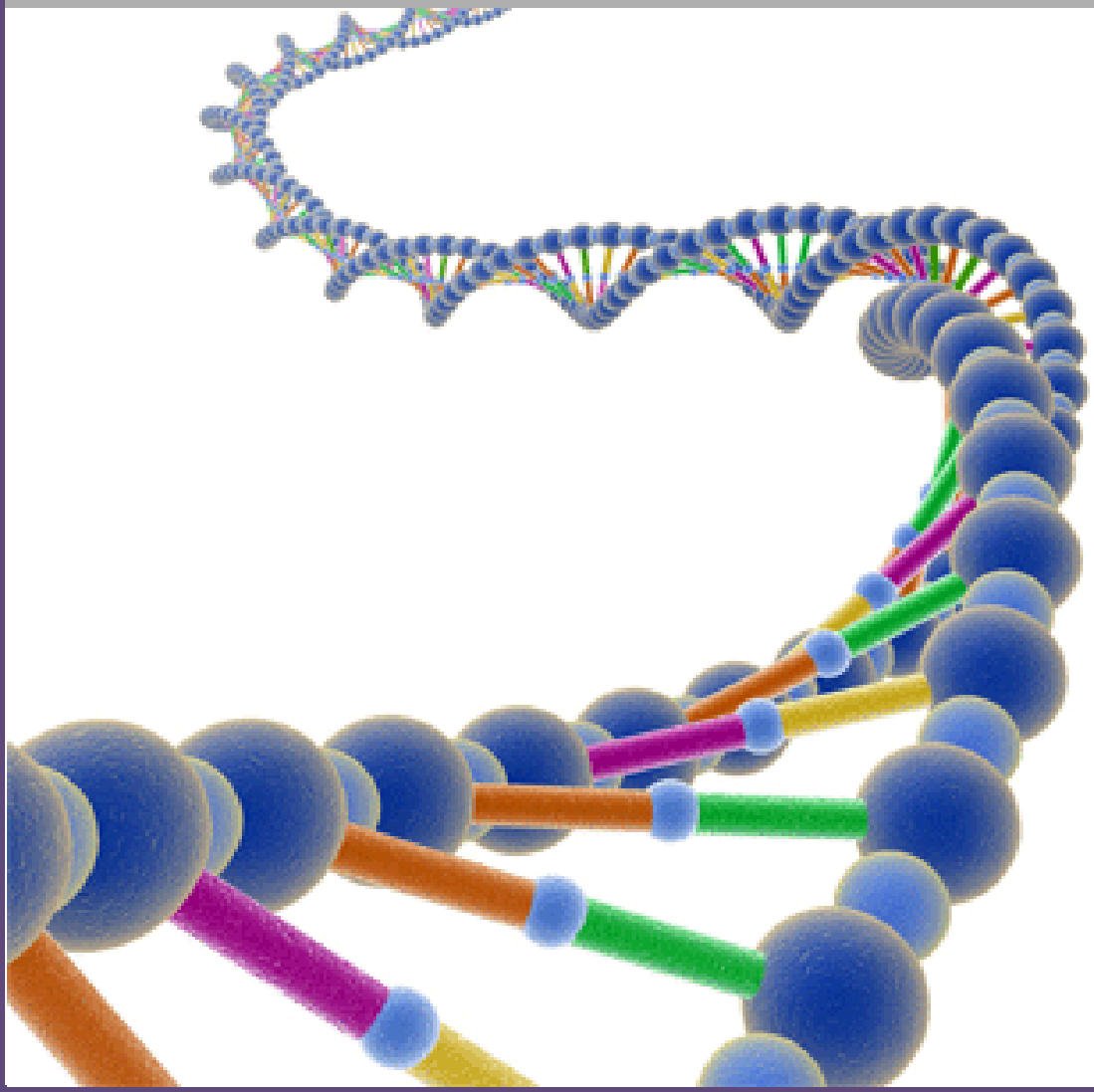
PROS
Centro de Investigación en Métodos de Producción de Software

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

JULIO 2011

INTEGRACIÓN DE BASES DE DATOS GENÓMICAS: UNA APROXIMACIÓN BASADA EN MODELADO CONCEPTUAL

Ainoha Martín Mayordomo



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Centro de Investigación en Métodos de Producción de Softwre (ProS)

Tesis de Master

Integración de bases de datos genómicas: una aproximación basada en modelado conceptual

Presentada por:

Ainoha Martín Mayordomo

Dirigida por:

Dra. Matilde Celma Giménez

VALENCIA, 8 DE JULIO DE 2011

CONTENIDOS

AGRADECIMIENTOS

En primer lugar quiero agradecer al Centro de Investigación en Métodos de Produccion de Software (ProS) y en concreto a todos los miembros del grupo Genoma por su colaboración en esta tesis y por hacer que los días de trabajo juntos sean especialmente agradables.

Mi más sentido agradecimiento a mi directora Mati por todo su tiempo dedicado a corregirme esta tesis y a ayudarme a mejorar en todo lo posible el trabajo realizado. De la misma manera agradezco también a Oscar Pastor Director del Centro ProS, por darme la oportunidad de trabajar en su grupo de investigación y por descubrirme un mundo tan apasionante como lo es la genómica. Y hablando de genómica, también agradecer a nuestra bióloga Ana Levin por compartir conmigo sus conocimientos y experiencia en este campo.

Por otra parte quiero hacer una mención especial al Centro de Investigación Principe Felipe en el que he estado colaborando durante el periodo de realización de esta tesis y que ha sido el punto de partida que me ha llevado a decantarme por este tema. En especial a Matthijs, miembro de Genoma pero que ha estado colaborando conmigo en el CIFP, a Nacho y a Joaquín, pero también al resto del equipo de biólogos que han ayudado a que este trabajo haya sido posible de realizar.

Mi mayor y eterno agradecimiento a mis padres que han estado conmigo en todo momento, apoyándome y dándome su cariño para que en los momentos difíciles siempre viese esa luz que a veces es tan complicada de encontrar.

A Nacho, mi peque, por estar a mi lado en todo momento, bueno y malo, durante la realización de esta tesis prestándome todo su apoyo y cariño haciendo que todo fuese más fácil.

Quiero agradecer también este periodo de mi vida a mis compañeras y amigas Ani y Vero por su ayuda, por hacerme pasar tan grandes momentos de confidencias y risas y porque estar trabajando con personas como ellas hace que madrugar valga la pena.

Como no, dar las gracias a Larry y Malonda por sus tardes y noches de cenitas en su piso ayudándome a desconectar de la tensión acumulada, a mi Mellas por lo mucho que me ha ayudado durante todo este tiempo y por lo importante que es para mí, a Héctor porque siempre me saca una sonrisa hasta en los momentos más complicados, a Conqui, mi compi de piso, por esos momentos de desconexión y desestrés viendo series y por esa tranquilidad transmitida en mis momentos de nerviosismo, y por supuesto, al resto de mis amigos que por haber estado ahí también siempre que lo he necesitado.

Por último, agradecer a mis excompañeros de laboratorio Clara, Mario, Carla, Sergio y Arthur por todos los momentos compartidos y las risas que hemos pasado juntos, a mis compañeros del máster y a todos mis profesores.

1. INTRODUCCIÓN

1.1 Motivación

Hoy en día, está ampliamente aceptado que el desarrollo de sistemas de información (SI) de calidad exige el uso de metodologías basadas en el modelado conceptual [1]. El uso de modelos conceptuales permite a los diseñadores realizar su trabajo a un alto nivel de abstracción, permitiéndoles conocer y comprender el dominio del problema antes de abordar su solución. Asimismo, el desarrollo por fases, iniciado por el modelado conceptual, permite el uso de herramientas de desarrollo dirigido por modelos, que representan la tecnología actual para el desarrollo de sistemas de software [1, 2].

Los modelos conceptuales han sido utilizados en muchos ámbitos de aplicación [3] con un resultado muy satisfactorio. Aún así, existen dominios, alejados de los clásicos dominios organizacionales, en los que a pesar de su complejidad y de los grandes volúmenes de datos que manejan, no se hace uso de ellos. Un ejemplo de este tipo de dominios es el de la Bioinformática. Hasta ahora, la Ingeniería del Software (IS) en este campo se ha centrado principalmente en el diseño de potentes y eficientes algoritmos de análisis que solucionen problemas concretos para los laboratorios, prestando poca atención al desarrollo de sistemas de información que proporcionen datos genómicos de calidad.

La Genómica, disciplina que estudia el genoma de los organismos, es un campo en constante evolución. Desde 1970, las técnicas de secuenciación, alineamiento de secuencias y análisis biológico han avanzado rápidamente produciéndose una gran cantidad de datos en los laboratorios. Este ingente volumen de información disponible ayuda, obviamente, al investigador en su trabajo, pero al mismo tiempo, es un hecho que su constante crecimiento dificulta cada vez más la búsqueda de la información más adecuada en cada momento. Esta dificultad se incrementa debido a que la información está dispersa en numerosos repositorios, sitios web, bancos de datos, ficheros públicos, etc, por lo que la tarea de encontrar alguna información dentro de este caos se convierte en una tarea tediosa para los biólogos, que incluso a veces puede llegar a ser un objetivo imposible de alcanzar. Si a esta situación, de crecimiento constante del volumen de datos y del número de repositorios disponibles, se añade que la misma disciplina, es decir los conceptos biológicos sobre los que se asienta, está en continua evolución, la necesidad de abordar la construcción de sistemas de información genómica desde una aproximación metodológica se convierte en una necesidad.

En el año 2009, surge una colaboración entre el centro de investigación PROS de la UPV y el Centro de Investigación Príncipe Felipe de Valencia, con el objetivo de analizar su base de datos genómica (Infrared), con la intención de determinar su validez en lo que se refiere a la representación de los conceptos biológicos del dominio, así como su capacidad para integrar información procedente de las fuentes de datos genómicas de más impacto en el área. El análisis de esta base de datos, condujo a la conclusión de que el desarrollo seguido había sido un desarrollo ad-hoc, es decir, se iban creando nuevas tablas para cada una de las bases de datos que se deseaba cargar, con lo que la presencia de información redundante se multiplicaba continuamente. No se estaba siguiendo una estrategia integradora sino una estrategia de extensión de la base de datos original. Se disponía de toda la información en la misma base de datos, pero las búsquedas, debido a la redundancia, seguían siendo muy costosas y tediosas.

De la necesidad de superar esta situación, surgió el proyecto de desarrollo de un sistema de información que contemplase la integración de toda la información genómica dispersa actualmente en numerosas fuentes de datos, un sistema que fuese flexible a la evolución del dominio, así como a la evolución e incremento de dichas fuentes. Éste es el punto de partida del trabajo que ahora se presenta.

1.2 Objetivos

El objetivo general de esta tesis es el diseño y desarrollo de un sistema de información genómica.

El dominio de la Genómica es un dominio nuevo y en continua evolución, por lo que una parte importante del trabajo ha consistido en estudiar y recopilar todo el conocimiento actual en el dominio. Esta labor ha sido costosa porque al ser la Genómica un campo científico de reciente aparición, los conceptos biológicos no están suficientemente asentados, admiten distintas definiciones y en muchos casos se contradicen o solapan. Esta situación exige que la construcción de sistemas de información en este ámbito pase por el uso de aproximaciones metodológicas basadas en un riguroso modelado conceptual del dominio.

Otra característica relevante de este campo científico es el gran volumen de datos existente, resultado de las investigaciones, y su dispersión en numerosas y heterogéneas fuentes de datos. Este hecho convierte el diseño, implementación y carga de la base de datos del sistema, en un problema de mayor complejidad que la usual en otros ámbitos de aplicación.

Así, de acuerdo a lo anterior, los objetivos específicos en los que se refina el objetivo general son:

1. Diseño[1] de un modelo conceptual del genoma que represente y unifique el conocimiento actual en el dominio.
2. Diseño de una base de datos, de acuerdo al modelo conceptual, que permita la integración de la información genómica residente en múltiples y heterogéneas fuentes de datos.
3. Diseño e implementación de un módulo de carga y mantenimiento, flexible a la aparición de nuevas fuentes de datos y a su evolución.
4. Demostración de la validez de nuestra propuesta al extender el modelo conceptual diseñado, integrando la propuesta del estándar BioPax para rutas metabólicas.

1.3 Estructura de la tesis

A fin de cumplir con los objetivos descritos, la tesis se estructura como se detalla a continuación. En el capítulo 2 se realiza una revisión del estado del arte, analizando los principales trabajos existentes en modelado conceptual en el ámbito bioinformático. Se analizan también los diferentes estándares aceptados en esta ciencia. El capitulo 3 es un breve resumen realizado con el objetivo de que se comprendan mejor los conceptos más relevantes de esta ciencia. En el capítulo 4 se presenta el diseño y el

[1] El modelo conceptual propuesto ha sido diseñado por el grupo de investigadores del grupo Genoma, grupo del que la autora forma parte y en cuyas discusiones ha participado

desarrollo de un sistema de información genómica: el modelo conceptual que plasma el conocimiento del dominio, la base de datos implementada en SQL, las correspondencias entre los repositorios de datos externos y la base de datos diseñada y finalmente un módulo de carga que, mediante el uso de una estrategia ETL, realiza el proceso de carga y actualización de la base de datos. En el capítulo 5, se estudia la propuesta BioPax, a petición de los biólogos que han colaborado en el trabajo. En este capítulo se explica detalladamente qué pretende BioPax con su propuesta y se analizan sus desventajas y nuestra propuesta de solucionar estos problemas mediante técnicas de modelado conceptual. Una vez diseñado el modelo conceptual para BioPax, se muestran las correspondencias de ambas representaciones con sus similitudes y diferencias y, se realiza una propuesta sobre la integración del modelo BioPax en el modelo conceptual del genoma modelado en el capitulo 4.

2. ESTADO DEL ARTE

El uso del modelado conceptual en el campo de la Bioinformática no ha sido muy frecuente. Es cierto que existen numerosos repositorios de datos que almacenan información genómica, pero la mayoría de ellos crecen de forma desestructurada y no es usual encontrar un riguroso y sólido modelo conceptual como base de ellos.

A pesar de esta carencia, existen algunas propuestas realizadas en este ámbito. El pionero fue Paton [4-6] que introdujo los primeros trabajos sobre el modelado conceptual del genoma desde diferentes perspectivas. Paton presentó modelos que describían el genoma de la célula eucariota, la interacción entre proteínas, el transcriptoma…, sin embargo su trabajo no tuvo una clara continuación.

Por otro lado Ram et al [7] también aplicaron principios de modelado conceptual pero en el contexto de las proteínas. La consulta de datos voluminosos y de estructura compleja, como es el caso de las proteínas 3D, requiere el uso de modelos expresivos que soporten explícitamente y capturen la semántica de este tipo de datos. En su trabajo Ram muestra cómo la comparación y búsqueda en la estructura de una proteína en 3D se facilita con el modelado conceptual. A pesar de ser un dominio "pequeño", se demuestra que el modelado conceptual ayuda a manejar datos de manera efectiva. Es importante destacar que el trabajo presentado por Ram es simplemente una parcela de lo que pretende ser el trabajo aquí propuesto, por lo que podría estar perfectamente embebido en él.

Existen otras propuestas que surgen con el objetivo de modelar tipos particulares de genomas como el proyecto e-Fungi [8, 9], que permite realizar análisis comparativos y sistemáticos de los genomas de los hongos. Para soportar este análisis se ha desarrollado la base de datos e-Fungi, que integra datos de más de 30 genomas de hongos. Para acceder a los datos, un conjunto de tareas de análisis está disponible a través de una interfaz web. Las tareas de análisis vienen motivadas por recientes estudios genómicos comparativos y tratan de soportar la rápida evolución del estudio de la biología así como los esfuerzos de la comunidad para mejorar las anotaciones de los genomas. Esta herramienta está desarrollada para soportar únicamente el dominio de los hongos, pero muestra un claro ejemplo de esfuerzo cuyos resultados podrían ser perfectamente proyectados en nuestro proyecto del genoma completo de todas las especies.

Continuando con otro tipo de propuestas existentes basadas en técnicas de modelado conceptual, podemos destacar también un conjunto importante de implementaciones bioinformáticas favorablemente aceptadas por la comunidad científica en las que en mayor o menor grado también han sido utilizadas estas técnicas. Un ejemplo de esto es el trabajo de [5] el cual es una aproximación dirigida por modelos para la generación parcial de interfaces de usuario cuyo objetivo es realizar búsquedas en repositorios de datos bioinformáticos. Este trabajo demuestra que los modelos conceptuales pueden ser usados para generar aplicaciones futuras y no únicamente para representar un dominio. Cuando comparamos este trabajo con el nuestro, cabe destacar que las técnicas de modelado se utilizan únicamente para solventar una parte del gran problema al que nos enfrentamos, las interfaces de usuario, mientras que a nosotros nos gustaría proporcionar una amplia y unificada vista de modelado conceptual que abarque todos los aspectos del sistema de información.

Como propuestas más cercanas al trabajo realizado en esta tesis, cabe estacar las primeras propuestas realizadas por el equipo para modelar el genoma [10-12]. Estas propuestas constituyen el punto de partida

del trabajo presentado en este proyecto. Poco a poco, con la ayuda de biólogos y el avance de la ciencia, el conocimiento adquirido es mayor, por lo que las necesidades en el esquema conceptual van cambiando de manera que nuevas vistas aparecen así como información existente es modificada.

Otro enfoque propuesto para representar conceptos relacionados con el genoma, son los intentos de unificación de términos realizados por expertos en el campo de las ontologías. Un ejemplo de este tipo de representación la proporciona GeneOntology [13], iniciativa que nació con el objetivo de estandarizar la representación de los genes y sus atributos. El proyecto proporciona un vocabulario controlado de términos para describir todas las características de los genes y de los datos anotados en el genoma a través de una herramienta de acceso a ellos. Pero a pesar de esto, este tipo de solución que proporciona va más orientada a solucionar un problema que a solucionar una situación global que afecta al mundo de la bioinformática.

Por otro lado, los expertos en rutas metabólicas también han generado sus propios estándares tratando de definir los diferentes conceptos que componen las reacciones que se producen en el organismo. De la recopilación de todos ellos se ha creado un nuevo glosario de términos llamado BioPax [14] que actualmente cuenta con una gran aceptación dentro del campo de la bioinformática. Otras iniciativas que se han generado en torno a este dominio son: PSI-MI [15], SBML [16], CellML [17] y SBGN [18], cada una de ellas con un objetivo diferente.

PSI-MI es un formato de intercambio de datos entre interacciones moleculares. La iniciativa de estándares en proteómica (PSI) trata de definir estándares útiles en la comunidad de biólogos expertos en proteómica para representar datos sobre interacción moleculares de manera que se facilite su comparación, intercambio y verificación. Está considerado un buen estándar pero no soporta todas las rutas metábolicas existentes en el genoma. Por otro lado, SMBL es un lenguaje de intercambio de modelos computacionales orientado hacia la descripción de procesos biológicos. Los aspectos dinámicos y cuantitativos de los procesos biológicos incluyendo aspectos temporales de bucles de realimentación y ondas de calcio son algunos de los aspectos con los que trata. Como sucedía con el estándar anterior, no soporta toda la información existente sobre rutas metabólicas. CellML, al igual que SMBL, es un lenguaje estándar cuyo propósito es almacenar e intercambiar modelos matemáticos computacionales y finalmente SBGN es un formato para estandarizar la notación gráfica usada en los mapas de procesos biológicos. Igual que el resto de estándares, solo cubren uno de los aspectos de las rutas metabólicas

BioPax es un lenguaje de intercambio para representar rutas biológicas a nivel molecular y celular que sirve para facilitar el intercambio de datos entre rutas metabólicas. El rápido crecimiento del volumen de datos respecto a este tipo de reacciones moleculares ha estimulado el crecimiento de bases de datos y herramientas computacionales para lograr su interpretación. BioPax, glosario que fue creado con el beneplácito de la comunidad científica y en el que están involucrados expertos de las principales bases de datos existentes en la actualidad en rutas metabólicas, resuelve este problema representando todo tipo de interacciones moleculares mediante un lenguaje. De esta manera los tipos de datos de rutas metabólicas están estructurados siguiendo el mismo formato en la mayoría de bases de datos relevantes en estos momentos respecto a esta parcela concreta del dominio bioinformático. Este formato es muy útil debido a que la comunidad científica lo tiene considerado en alta estima, pero a pesar de ello como pasaba en otros casos arriba citados no tiene en cuenta la totalidad el genoma, sino que simplemente cubre una parte. A pesar de esto, debido al interés que ha generado para los biólogos su estudio se hace necesario para nosotros.

Una vez se han estudiado los trabajos relacionados existentes en el ámbito bioinformático basados en técnicas de modelado conceptual, así como estándares biológicos, vocabularios y ontologías, salta a la vista la poca cantidad de ellos que existen. La mayoría como se ha explicado, no proporcionan como base de fondo un modelo conceptual, por lo que los datos llegan a ser un caos. Tao y Embley [19] han analizado cómo diferente información genómica es almacenada en diferentes fuentes de datos biológicas. Este estudio ha sido muy útil ya que han detectado una gran cantidad de datos inconsistentes, datos redundantes o información incompleta que dificulta el trabajo de los expertos. Este hecho es corroborado por las fuentes de datos estudiadas en este trabajo: Ensembl [20], HapMap [21, 22], Uniprot [23, 24]… y podrían ser eliminados usando una aproximación basada en modelos como nosotros proponemos.

Otra cosa a tener en mente es que los datos son almacenados en ficheros, como por ejemplo HapMap o en desestructuradas bases de datos, como por ejemplo Ensembl y que estos repositorios han sido implementados únicamente para almacenar información específica, pero ¿qué ocurre cuando se quiere unir información entre repositorios para realizar conexiones entre elementos existentes en cada uno de ellos? Este es un trabajo manual y tedioso y que a veces puede llegar a ser imposible.

Ejemplos de este tipo de fuentes de datos son por ejemplo Uniprot, que centra su trabajo principalmente en el estudio de proteínas, Cosmic que centra su trabajo en el estudio de mutaciones relacionadas con el cáncer, HapMap que centra su atención principalmente en variaciones comunes dentro del genoma humano contenidas en más del 1% de la población… y así sucesivamente con todas las fuentes de datos estudiadas. Este hecho corrobora lo anteriormente citado sobre la heterogeneidad de las fuentes y la información específica que corresponde a cada una de ellas.

Con nuestra aproximación propuesta basada en modelos conceptuales, un esquema de bases de datos del genoma es automáticamente derivado para almacenar toda la información disponible de diversas fuentes.

3. ESTUDIO BIOLÓGICO

La herencia genética ha sido un tema que ha cautivado el interés del hombre a lo largo del tiempo. Muchas son las preguntas que surgen respecto a este tema y mucha la información que se conoce gracias a los avances de la ciencia, quién no se ha preguntado alguna vez... ¿por qué los hijos se parecen a los padres o a los abuelos?, ¿qué o quiénes son los causantes de dicho parecido?, ¿dónde se almacena esa información?, ¿cómo se interpreta?, ¿por qué mi abuelo es daltónico y yo también? … Estos interrogantes y sus primeras repuestas marcaron el inicio del conocimiento sobre la herencia y de la ciencia, que más tarde, sería conocía como Genética.

La Genética nace como una rama de la Biología con el objetivo de comprender la herencia biológica que se transmite de generación en generación. En 1865, Gregor Mendel, al que podríamos denominar el padre de la Genética ya que realizó los primeros trabajos en esta ciencia, describió por medio de trabajos llevados a cabo con diferentes variedades del guisante, las hoy llamadas leyes de Mendel que rigen la herencia genética y que más tarde fueron ampliadas y generalizadas a un gran número de organismos vivos:

1ª Ley de Mendel o principio de la uniformidad.

2ª Ley de Mendel o principio de la segregación

3ª Ley de Mendel o principio de la combinación independiente.

El principal objeto de estudio de la genética es el ADN o ácido desoxirribonucleico. El cuerpo humano está formado por 10 billones de células que se conocen como las unidades funcionales de los seres vivos. Cada una de ellas, posee una zona llamada núcleo donde se almacena la información genética en forma de ADN. Este ácido es la molécula que controla todos los procesos celulares como la alimentación y la reproducción celulares o la transmisión de caracteres de padres a hijos. El conocimiento actual del genoma es tan amplio que por razones de comprensión en esta tesis se va a presentar por vistas de acuerdo a su funcionalidad: primero se realiza una introducción a la parte estructural que describe cómo está formado el ADN y los elementos que lo constituyen, en segundo lugar se definen los procesos de transcripción y traducción mediante los cuales el ADN se codifica en proteínas, en tercer lugar se presentan los fallos ocasionados en la maquinaria de la célula a la hora de reproducirse y las consecuencias que este proceso puede ocasionar y por último se describen las rutas metabólicas o interacciones biológicas entre procesos existentes dentro de las células y su funcionalidad.

3.1 Estructura del genoma

La molécula de ADN (Fig. 1) se encuentra formada por dos cadenas muy largas enrolladas entre sí formando una estructura helicoidal alrededor de un eje que da lugar a una doble hélice parecida a una escalera de caracol. La parte lateral de esta escalera está formada por fosfatos y azucares orientados hacia el exterior de la molécula y los peldaños son pares de bases. En esta estructura, la adenina se empareja con la timina (A-T, T-A) y la citosina se empareja con la guanina (C-G, G-C). Ya que el esqueleto azúcar-fosfato es siempre igual, la manera de escribir la información genética se realiza mediante un alfabeto de 4 letras en el cual se tiene en cuenta el tipo de nucleótidos y el orden en que se disponen. Esta disposición de la información de manera habitual es denominada secuencia. Esta molécula de ADN tiene la capacidad de desdoblarse y dar lugar a otra molécula idéntica, así es como se pasa la información.

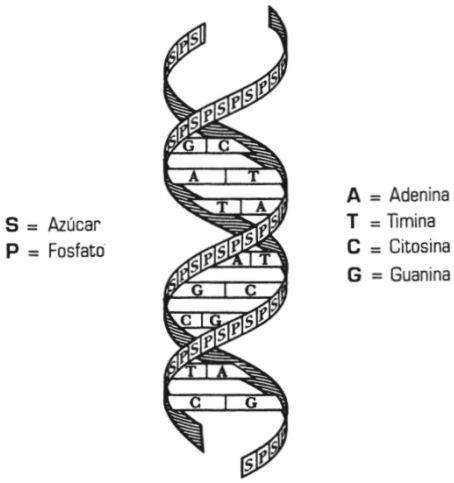


Fig. 1 Molécula de ADN

Si estirásemos el ADN, llegaría a medir hasta 1,8 metros, es decir, unas 300000 veces más que el núcleo. Para evitar este problema, el ADN esta plegado formando unas estructuras denominadas cromosomas. Cada cromosoma es una única molécula de ADN, que a su vez está formada por miles de nucleótidos.

Los cromosomas (Fig. 2) son pequeños cuerpos en forma de bastoncillos en que se organiza la cromatina del núcleo celular durante las divisiones celulares. Su número es constante para una especie determinada, en el hombre 46; de ellos un par son cromosomas sexuales (determinan el sexo del sujeto) y 44 son autosómicos (no sexuales).

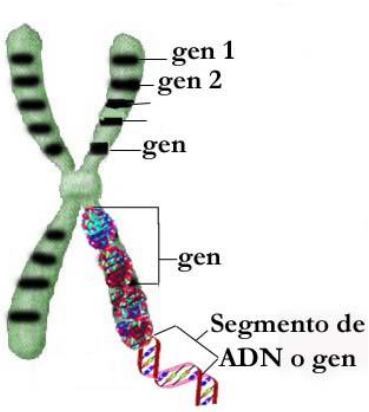


Fig. 2 Cromosoma y genes

En un cromosoma se encuentran muchos elementos, entre ellos los genes (Fig. 2). Por ejemplo, en cada célula del cuerpo humano hay aproximadamente 30.000 genes y cada uno de ellos ocupa en el cromosoma una posición determinada llamada locus [25].

En general, se denomina gen [26] a la secuencia lineal de nucleótidos de ADN que es esencial para una función específica, da lugar a un ARN (ácido ribonucleico) a través de un proceso de transcripción y lleva la información para sintetizar una proteína. Como el ADN, el ARN también está formado por una cadena de nucleótidos, pero a diferencia de éste, la molécula de ARN contiene un átomo de oxigeno que el ADN no tiene y contiene la base de uracilo U en lugar de la timina T. La secuencia de bases presente en el ARN determina la secuencia de aminoácidos de la proteína por medio del código genético. Es importante resaltar que, si bien el ADN es donde se almacena la información genética de un organismo, las proteínas son las que ejecutan dicha información porque son las moléculas esenciales para todos los aspectos de estructura y actividad celular.

Pero no todos los genes codifican proteínas sino que algunos de ellos cumplen su función en forma de ARN, como por ejemplo regular post-transcripcionalmente otros genes. Entre estos encontramos genes de ARN transferente, micro ARN, ARN ribosómico, ribozimas y otros ARN pequeños de funciones diversas.

Por otro lado, existen otros elementos que también forman parte del cromosoma, son los elementos reguladores y las regiones conservadas.

Los elementos reguladores determinan la expresión de los genes así como del proceso post-transcripcional y entre ellos destacan: los TFBS o puntos en la región de ADN donde los factores de transcripción se unen produciendo un efecto en la transcripción del gen, cpg islands o regiones del ADN que aparecen al comiendo o cerca del inicio de la transcripción y contienen una alta frecuencia de Cs y Gs, tríplex o regiones del ADN en las que secuencias de ADN se intercalan en la doble hélice de ADN, o microARN o puntos en la región de ADN donde los genes de tipo microRNA se unen produciendo un efecto regulador del transcrito.

Las regiones conservadas son secuencias biológicas similares o idénticas que pueden encontrarse dentro de múltiples especies de organismos. Estas regiones sirven como evidencias de conservación estructural y funcional y son mantenidas evolutivamente. Un ejemplo de región conservada es la secuencia promotor "caja TATA" que se encuentra en la mayoría de los eucariotas.

3.2 Trascripción y traducción del ADN

El proceso por el cual el ADN se traduce en una secuencia de proteínas, pasa por dos fases:

- **Transcripción:** primera fase del proceso que utiliza como patrón la secuencia de ADN y sintetiza el ARN.
- **Traducción:** segunda fase del proceso que utiliza como patrón la secuencia de ARN y sintetiza la proteína.

3.2.1 Transcripción

La transcripción del ADN (Fig. 3) es el primer proceso de la expresión génica, mediante el cual se transfiere la información contenida en la secuencia del ADN hacia la secuencia de proteína utilizando diversos ARN como intermediarios.

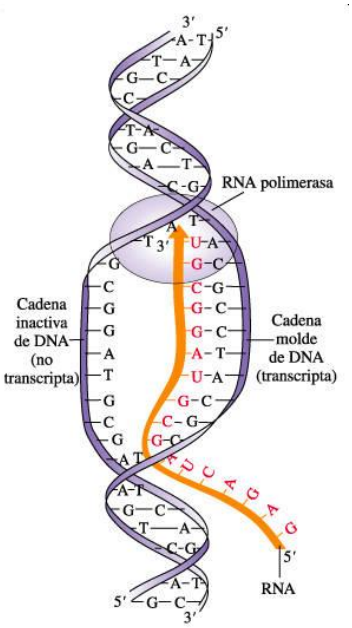


Fig. 3 Proceso de transcripción de ADN en ARN

Cuando comienza, en la fase de iniciación, el ADN se separa para poder ser copiado ya que ha de ser asequible para la enzima ARN-polimerasa. En la siguiente fase, fase de elongación, las materias primas que forman la molécula de ARN: ATP, GTP, CTP, UTP, quedan enlazadas a lo largo de una cadena sencilla de ADN. Durante la fase de maduración las secuencias intrónicas, aquellas que no contienen información para la síntesis de proteínas, son eliminadas de la secuencia, dando lugar a una secuencia formada únicamente por exones que forman la región codificante del gen y transportan la información para producir la proteína. El proceso de retirada de los intrones y conexión de los exones se llama Splicing y da lugar al ARNm (ARN mensajero) maduro. Es importante mencionar que un mismo gen puede producir diferentes proteínas gracias al fenómeno conocido como Splicing Alternativo en el que algunos exones o parte de ellos pueden ser eliminados junto con los intrones que los flanquean y algunos intrones o parte de ellos pueden no ser eliminados durante el proceso. De esta manera se crean diversos ARNm que son traducidos a su vez en distintas proteínas. Cabe destacar que este Splicing Alternativo, no es de ninguna manera un proceso aleatorio sino que ha evolucionado de manera que las diferentes proteínas así creadas sean todas funcionales [27].

Este proceso de traducción se produce en el núcleo de la célula y dará lugar a la molécula de ARNm. Tras la formación de dicha molécula, ésta se desplazará hasta el citoplasma de la célula a través de los poros de la membrana nuclear para participar en el proceso de traducción o síntesis de proteínas.

3.2.2 Traducción

La información genética contenida en el ARNm deberá ser traducida en el citoplasma por una fábrica de proteínas: el ribosoma. En este proceso (Fig. 4) se pueden distinguir cuatro fases: en primer lugar la activación de los aminoácidos, en segundo lugar la iniciación de la síntesis de proteínas con el codón de iniciación, en tercer lugar, el alargamiento de los polipéptidos durante la traducción y por último, finalización de la cadena proteínica.

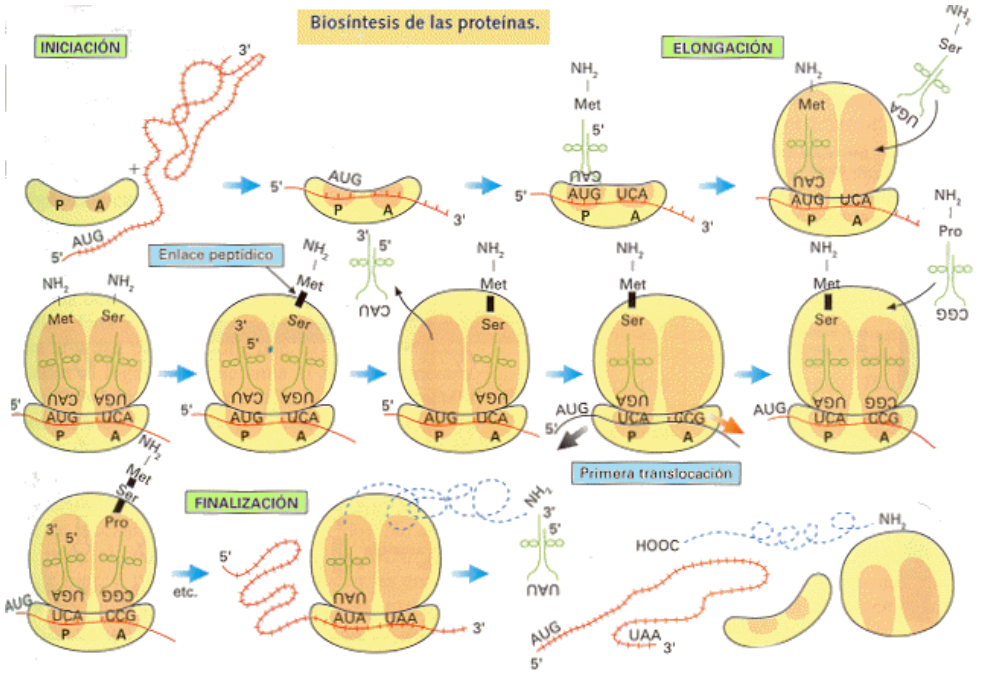


Fig. 4 Proceso de traducción de ARN en proteína

Se denomina codón al grupo de tres nucleótidos adyacentes que codifican para un aminoácido (Fig. 5). Cabe destacar que la iniciación de la síntesis de proteína se realiza mediante el codón AUG que da la señal de salida para el proceso de síntesis y que por contraposición también existen codones de finalización que son UGA, UAG y UAA y que debido a que no identifican ningún aminoácido detienen la síntesis.

Existen una serie de ARN que intervienen durante el proceso de traducción y son: ARNm que es el portador de la información genética

Segunda Letra

Primera letra	U	C	A	G	
U	UUU UUC Fenilalanina UUA UUG Leucina	UCU UCC UCA UCG Serina	UAU UAC Tirosina UAA UAG Código de parada (stop codon)	UGU UGC Cisteína UGA Codigo de parada (**) UGG Triptófano	U C A G
C	CUU CUC CUA CUG Leucina	CCU CCC CCA CCG Prolina	CAU CAC Histidina CAA CAG Glutamina	CGU CGC CGA CGG Arginina	U C A G
A	AUU AUC AUA Isoleucina AUG Metionina (Iniciación)	ACU ACC ACA ACG Treonina	AAU AAC Asparagina AAA AAG Lisina	AGU AGC Serina AGA AGG Arginina	U C A G
G	GUU GUC GUA GUG Valina	GCU GCC GCA GCG Alanina	GAU GAC Acido Aspartico GAA GAG Acido Glutámico	GGU GGC GGA GGG Glicina	U C A G

Fig. 5 Código de las proteínas

del ADN, ARNribosomico que se encuentra asociado a las proteínas formando los ribosomas que son los encargados de leer la secuencia de tripletes de las bases del ARNm y el ARNtransferencia que transporta los aminoácidos hasta los ribosomas para cederlos a la cadena proteínica que se está formando.

Una vez finalizados los procesos de transcripción y traducción, la secuencia proteínica se crea. Las proteínas son macromoléculas de peso molecular elevado formadas por aminoácidos que desempeñan un papel fundamental en la vida, formando por una parte la estructura básica de los tejidos y por otro desempeñando funciones metabólicas y reguladoras.

3.3 Variaciones en la secuencia de ADN

Una célula tiene una maquinaria muy sofisticada que permite hacer copias muy precisas de la molécula de ADN, incluso existen diversos sistemas de reparación en caso de ocurrir algún fallo en ellas. A pesar de esto, dichos fallos ocurren y se traducen en variaciones o errores genéticos dentro de la cadena.

Un simple cambio en la secuencia de un gen, puede conllevar desde la indiferencia hasta las consecuencias más drásticas. Un cambio, o variación que es como lo llamaremos a partir de ahora, en la secuencia de ADN es el responsable de que cada uno de nosotros tenga los ojos de diferente color, sea más alto o más bajo, que dos plantas iguales tengan flores diferentes..., pero no toda variación es buena, en ocasiones puntuales también es la causante de ciertas enfermedades como el cáncer o la fibrosis quística que puede llegar a ocasionar a veces incluso la muerte. Esto es debido a que una variación en la reproducción de una célula puede interrumpir la actividad normal de un gen dejando a éste inerte de realizar sus funciones originales.

Dado que las variaciones (Fig. 6) son cambios en una posición concreta de la secuencia de ADN y pueden clasificarse según su descripción en:

- **Inserción:** una o más bases de nucleótidos adicionales se introducen en la secuencia de ADN.
- **Borrado:** una o más bases de nucleótidos de la secuencia de ADN son eliminadas.
- **Sustitución:** uno o más nucleótido en la secuencia de ADN son sustituido por otra secuencia de nucleótidos de menor, igual o mayor tamaño.
- **Inversión:** uno o varios nucleótidos de la secuencia de ADN invierten su posición.

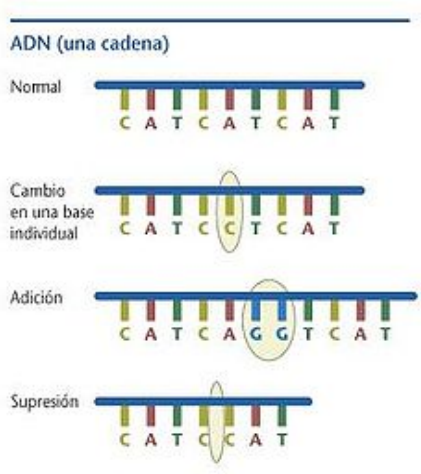


Fig. 6 Ejemplo de cadena de ADN y posibles variaciones

La posición en la que se haya producido la variación es un detalle muy importante, ya que si una inserción, un borrado o una sustitución de diferente tamaño ocurre dentro de un gen, se produce un cambio en la pauta de lectura. Como se ha comentado en el apartado anterior, durante el proceso de traducción, la lectura del ARNm se realiza en grupos de tres nucleótidos, por lo que el hecho de insertar o borrar nucleótidos de la secuencia implica una interpretación totalmente distinta de la secuencia de ARNm y produce una secuencia de aminoácidos totalmente distinta a la inicial. En caso de ser una inserción o un borrado de tres nucleótidos, la variación se llama in-frame e implica el borrado o inserción de un aminoácido que puede tener

consecuencias igual de graves que las de alteración en la pauta de lectura. Este tipo de variaciones que producen un efecto dañino en la salud de las personas son conocidas por nosotros con el nombre de mutaciones debido a su efecto nocivo y su baja frecuencia de aparición.

Por otra parte, existen variaciones que suelen afectan a más del 1% de individuos de una misma población, son conocidas como polimorfismos y son las principales causantes de cambios en el fenotipo de un individuo como por ejemplo el color de la piel. Los polimorfismos suelen ser la sustitución de un nucleótido por otro y se conocen como SNPs aunque a veces pueden ser más complicados, como por ejemplo una variación en el número de copias de una secuencia de ADN determinada en un porcentaje de individuos que toma el nombre de CNP. Este tipo de variaciones son más frecuentes y forman hasta el 90% de todas las variaciones genómicas.

3.4 Rutas metabólicas

Una ruta metabólica (pathway) (Fig. 7) es una sucesión de reacciones químicas que ocurren dentro de una célula y de las que a partir de un sustrato inicial se obtiene un producto metabólico. Las enzimas catalizan estas reacciones y a menudo requieren minerales, vitaminas y otros factores para funcionar correctamente. Debido a la cantidad de químicos que se ven involucrados, la mayoría de estas rutas son elaboradas e implican una modificación paso a paso de la sustancia inicial para darle la forma del producto con la estructura química deseada. Además muchas de estas rutas metabólicas coexisten dentro de una célula y son importantes para mantener el proceso de homeostasis o equilibrio interno del organismo.

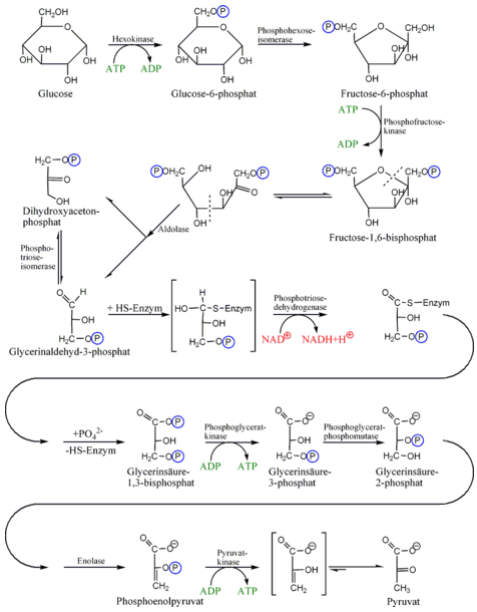


Fig. 7 La glucólisis, un ejemplo de ruta metabólica

Un pathway implica la modificación paso a paso de de una molécula inicial para formar otro producto. Este producto resultante puede ser utilizado de diversas maneras:

- Para ser usado inmediatamente como producto final del pathway.
- Para iniciar una nueva sucesión en el proceso de pathway (generando un flujo de pasos).
- Para ser almacenado en la célula.

Los pathways fluyen a menudo en una única dirección debido a que a pesar de que las reacciones químicas puedan producirse en ambos sentidos, las condiciones de la célula a menudo son termodinámicamente más favorables en una sola dirección del flujo. Por ejemplo un pathway puede ser el responsable de la síntesis de un aminoácido en particular, pero el desglose de ese aminoácido se puede producir a través de otro pathway distinto. Un ejemplo de excepción a esta regla es el metabolismo de la glucosa, la glicolisis, que tiene reacciones reversibles.

4. DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN GENÓMICA

La evolución de la tecnología informática y su uso extendido han convertido los sistemas de información en el pilar básico de las organizaciones. Aunque el campo de aplicación más frecuente es el organizacional, los sistemas de información, y su soporte tecnológico, las bases de datos, se han extendido rápidamente a otros ambitos de aplicación: sistemas bioinformáticos, geográficos, multimedia,…

Según Olivé [1], un sistema de información es un sistema que recoge, almacena, procesa y distribuye información. Siguiendo con el este autor, el modelado conceptual de estos sistemas de información, o en otras palabras el diseño de estos sistemas, es uno de los problemas más importantes de los que se encarga la Ingeniería del Software y, como bien dice al autor, es condición necesaria aplicar esta ciencia para la construcción de sistemas de información de calidad.

Un modelo conceptual es la representación de un dominio según los requisitos de los usuarios; esta representación se expresa en un lenguaje de modelado y es independiente de todo criterio de implementación. El modelo conceptual ayuda a los diseñadores y a los desarrolladores a conocer y comprender el dominio.

El sistema de información que se presenta en este trabajo es un sistema de información genómica, es decir un sistema que recoge, almacena, procesa y distribuye información sobre el genoma de los seres vivos. Las diferencias principales con un sistema de información organizacional son: la necesidad de manejar datos muy grandes (las secuencias de ADN), la continua evolución del dominio (los conceptos subyacentes) y la desestructuración y dispersión de la información existente en este campo.

Como bien dice Olivé el diseño del modelo conceptual debe ser el paso previo en el desarrollo del sistema de información, es por esto que en este capítulo se presenta el modelado conceptual del genoma.

Dado que el dominio genómico es muy complejo, identificar y unificar los conceptos del dominio no es tarea fácil. Existe además una dificultad añadida, y es que el conocimiento en este campo se encuentra todavía en desarrollo, evolucionando diariamente, lo que complica la estabilidad del modelo.

Por otro lado, los repositorios de información genómica existentes son muy heterogéneos. Cada repositorio usa conceptos distintos, o versiones distintas del mismo concepto, por ejemplo el concepto de variación en la cadena de ADN. Una variación en la secuencia de ADN puede ser referida como: variación, mutación, polimorfismo o SNP [28], lo que no es lo mismo, existen ligeras diferencias entre estos conceptos, por lo que son términos confusos a la hora de interpretar los datos. Como algunos trabajos [29] afirman, esta situación es un problema, y por ello el uso de metodologías de modelado conceptual es imprescindible.

Gracias a las ventajas de los sistemas de información como son la modularidad, su capacidad de evolución y el uso de modelos conceptuales, la evolución del dominio así como las diferentes representaciones de los mismos conceptos quedan solventadas así como el resto de problemas que surgen en los sistemas de

información tradicionales. El lenguaje utilizado para representar el modelo conceptual en este trabajo es UML.

Una vez definido el modelo conceptual del dominio, a partir de él se diseña una base de datos cuya función es almacenar la información procedente de los distintos repositorios de datos genómicos. Dado que nosotros no somos biólogos, distinguir la calidad de la información es para nosotros un trabajo imposible, es por eso que el trabajo se realiza junto a un conjunto de expertos biólogos especializados cada uno en un área diferente del genoma. Una vez seleccionados los repositorios, se define la correspondencia entre ellos y el esquema de la base de datos para así analizar cada fuente qué información proporciona y de qué manera casa con nuestro esquema. Finalmente, para almacenar la información necesaria, se implementa un módulo de carga siguiendo una estrategia ETL (extracción, transformación y carga) que proporciona flexibilidad ante los cambios en los repositorios o en nuestra base de datos.

4.1 Modelado conceptual del genoma[2]

En el año 2008, el grupo Genoma dentro del Centro de Investigación ProS de la UPV, inicia una nueva línea de investigación sobre el desarrollo y evolución del genoma que pretende potenciar el uso de los sistemas de información en el ámbito de la bioinformática, con la intención de solucionar problemas hasta ahora existentes y mejorar el trabajo de los expertos. Desde el inicio, varias son las versiones del modelo conceptual del genoma humano que se han desarrollado dentro del grupo de investigación. La primera de ellas, fue detalladamente descrita en [10]. Con el paso del tiempo nueva información se iba conociendo, y por supuesto con la ayuda de expertos, la primera descripción evolucionó a versiones posteriores [11, 12, 30] haciendo hincapié principalmente en aspectos relacionados con las mutaciones y la relación genotipo-fenotipo (Fig. 8).

En este apartado se presenta el estado actual del modelo conceptual del genoma, versión 3, evolucionado a partir de las versiones anteriormente citados. Este modelo ha sido diseñado por el grupo de investigadores del grupo Genoma[3]. Se ha extendido con nuevas vistas, rutas metabólicas (pathways), y se han completado las ya existentes, gracias a la colaboración de la bióloga del grupo Ana Levin y los colaboradores del Centro de Investigación Príncipe Felipe que cada jueves durante más de dos años han dedicado su tiempo a trabajar con nosotros. Es imprescindible destacar que este modelo no abarca únicamente la secuencia del genoma humano, sino que contempla el genoma de cualquier especie.

Para una mejor comprensión de este evolucionado modelo se divide en cinco vistas, todas ellas relacionadas entre sí, pero lo suficientemente independientes de las demás como para poder definirlas por separado sin perder información. Dichas vistas son: la vista estructural, la vista de transcripción, la vista de variaciones, la vista de rutas metabólicas y la vista de bibliografía y fuentes de datos. Para destacar las clases UML de intersección entre dichas vistas, éstas aparecen sombreadas en los diagramas.

[2] Los nombres de las clases y sus atributos aparecen en el documento en letra cursiva.

[3] El modelo conceptual propuesto ha sido diseñado por el grupo de investigadores del grupo Genoma, grupo del que la autora forma parte y en cuyas discusiones ha participado.

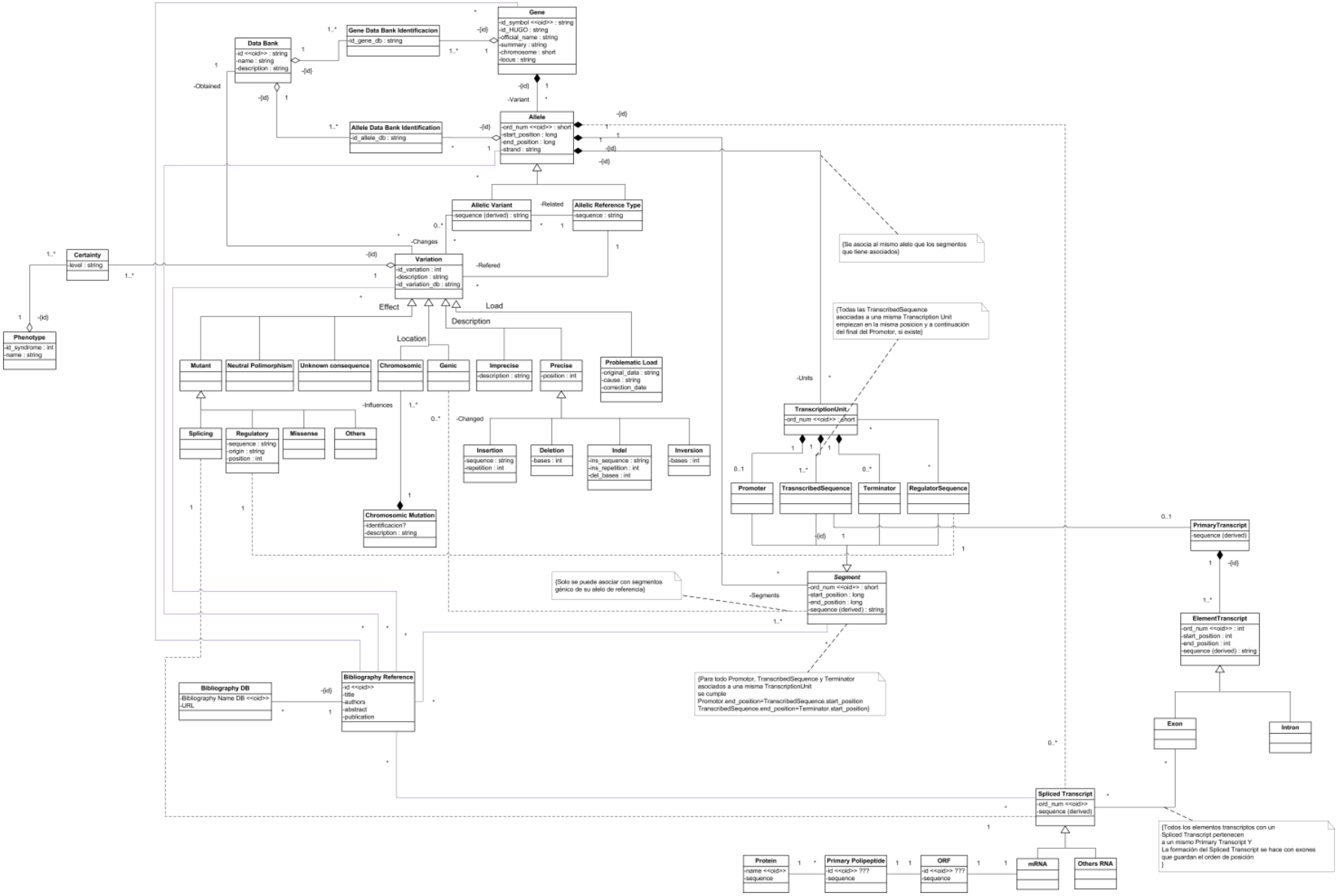


Fig. 8 Modelo conceptual del genoma humano v2

4.1.1 Vista estructural

Esta vista (Fig. 9), como su nombre indica, describe la estructura del genoma que a grandes rasgos se puede describir como la información genómica de un organismo distribuida en 23 pares de cromosomas compuestos de distintos tipos de elementos como por ejemplo, genes que codifican proteínas, secuencias reguladores, etc. Cada cromosoma pertenece a una única especie y contempla zonas calientes o hotspots [31] y subregiones llamadas citobandas que se hacen visibles microscópicamente después del tintado. A continuación se describen cada una de las clases que forman esta vista:

Chromosome

Chromosome es la clase principal de esta vista, y se define como una estructura organizada y única dentro del ADN donde genes, elementos reguladores y otras secuencias de nucleótidos son localizados. Además, un cromosoma tiene una serie de atributos por los cuales es identificado:

- ***name*:** nombre e identificador del cromosoma en la fuente de datos de la que se ha extraído la secuencia.
- ***sequence*:** secuencia de referencia del cromosoma.
- ***long*:** campo longitud que indica el número de nucleótidos que tiene la secuencia.

Hay que tener en cuenta, que debido a la cantidad de información genómica existente que va a ser almacenada en la base de datos, diferentes versiones del mismo genoma serán almacenadas en distintas versiones de la bases de datos y que además la secuencia de referencia almacenada no corresponde a ningún individuo en particular sino que se obtiene de una de las principales organizaciones de secuencias genómicas actuales.

Specie

Como se ha comentado anteriormente, esta nueva versión del modelo conceptual no está diseñada únicamente para la especie humana, sino que contempla todas las especies conocidas en la actualidad. Por lo tanto la clase *specie*, sirve para determinar a qué familia pertenece cada uno de los cromosomas. Sus atributos son:

- ***scientific_name*:** nombre científico e identificador por el cual se conoce la especie por ejemplo, homo sapiens.
- ***common_name*:** nombre común por el cual se conoce la especie. Por seguir con la analogía anterior el ejemplo aquí sería ser humano.
- ***ncbi_taxon_id*:** identificador dado a una especie por la organización de NCBI.
- ***assembly*:** identificador de la versión utilizada como secuencia genómica de referencia de dicha especie.
- ***date_assembly*:** fecha de la versión utilizada como secuencia genómica de referencia de dicha especie.
- ***source*:** fuente de la cual se obtiene la secuencia genómica de referencia.

Hotspot

La clase *hotspot* representa otra característica del cromosoma, la información sobre los puntos en la secuencia de ADN donde existe mayor probabilidad de que se produzca la recombinación durante el proceso de meiosis. Tiene dos atributos que son:

- ***hotspot_id*:** identificador interno del cruce de recombinación.
- ***position*:** punto dentro de la secuencia de ADN en la que se produce el proceso de recombinación.

Cytoband

La clase *cytoband*, conocida también con el nombre de banda citogenética, representa otra característica del cromosoma, la información sobre las subregiones de un cromosoma que llegan a ser visibles microscópicamente después del tintado durante una fase específica del ciclo celular. Una citobanda es representada mediante:

- ***name*:** el nombre de la citobanda sigue un formato definido: "q" o una "p", dependiendo del brazo del cromosoma, seguida de uno, dos o tres números separados por puntos dependiendo de la resolución utilizada i.e. (q24.22).
- ***score:*** indica la intensidad de tintado, que puede tomar cinco valores diferentes proporcionales a la presencia de A y T.
- ***start_position:*** posición inicial en la secuencia de referencia del cromosoma.
- ***end_position:*** posición final en la secuencia de referencia del cromosoma.

Chromosome element

La clase *chromosome element* representa información sobre fragmentos relevantes dentro del cromosoma. Tiene cuatro atributos:

- ***chromosome_element_id:*** identificador interno de cada uno de los elementos del cromosoma.
- ***start_position:*** posición inicial del elemento en la secuencia de referencia del cromosoma.
- ***end_position:*** posición final del elemento en la secuencia de referencia del cromosoma.
- ***strand:*** hebra dentro de la doble hélice en la que se encuentra el elemento dentro del cromosoma.

Los elementos del cromosoma pueden ser de tres tipos dependiendo de la función que desempeñen: *transcribable element*, *regulatory element* and *conserved region*.

Transcribable element

La clase *transcribable element* representa una región del ADN que se puede transcribir, o en otras palabras un elemento del que se crea un ARN complementario a partir de la secuencia de ADN. Este tipo de regiones pueden especializarse en dos tipos: *gene* and *exon*.

Gene

La clase *gene* representa una región de ADN que contiene la información necesaria para la síntesis de una macromolécula con una función celular específica, es decir contiene elementos reguladores que controlan el proceso de transcripción, normalmente sintetiza proteínas, pero también otro tipo de ARNs. Esta clase tiene cinco atributos:

- ***ensemble_gene*:** nombre del gen proporcionado por Ensembl[4].
- ***description:*** descripción del gene al que se hace referencia.
- ***biotype:*** especialización del tipo de gen dependiendo de las funciones que realiza, puede tomar valores como por ejemplo: snRNA, miRNA, protein coding, etc.
- ***status:*** determina el estado de validez en el que se encuentra cada elemento en la actualidad, puede tomar valores como: obsoleto, nuevo, etc.
- ***gc_percentage:*** a diferencia del resto de regiones de la secuencia de ADN, ha sido comprobado que las regiones transcribibles tienen mayor alto contenido de Gs y Cs en su secuencia y que dicho contenido es directamente proporcional a la longitud de la secuencia codificante. Este atributo almacena el porcentaje de pares de bases Cs y Gs que existen en el elemento.

Un gen, además, dependiendo del valor de su atributo biotype puede especializarse en diversos tipos de genes, dependiendo como se ha dicho anteriormente de las función que desempeñe. Existen muchos tipos de genes que podrían ser modelados, pero por simplificar el modelo se decide ilustrar un solo ejemplo, los factores de transcripción que se describen a continuación.

Tf

La clase *tf* (factor de transcripción) representa aquellos genes que codifican una proteína cuya función es regular la transcripción de otros genes o incluyo la suya propia y se define mediante el atributo:

- ***cons_seq:*** este atributo hace referencia a la secuencia de nucleótidos que una vez acoplada a las regiones de unión de la cadena de ADN realizará una función reguladora para el gen.

Exon

La clase *exon* representa un elemento transcribible que forma parte del gen, y que es además la unidad básica de los transcritos. Cada exón codifica una porción específica de la proteína completa, de manera que el conjunto de exones forma la región codificante del gen.

Regulatory element

La clase *regulatory element* representa regiones del ADN que realizan una función reguladora controlando ciertos procesos existentes dentro el ADN. Los elementos reguladores se especializan en dos clases dependiendo de si es un elemento regulador del gen o del transcrito: *gene regulator* y *transcript regulator*.

Gene regulator

La clase *gene regulator* representa los elementos reguladores del gen, entre los cuales se encuentran: *tfbs*, *cpg_island* y *triplex*.

Tfbs

La clase *tfbs* (transcription factor binding sites) son regiones de unión de los factores de transcripción que producen un efecto en la transcripción del gen bien sea de activación o represión. Esta clase tiene cinco atributos:

- ***name:*** nombre que toma el sitio de unión de los factores de transcripción.
- ***type:*** los sitios de unión de los factores de transcripción pueden ser de dos tipos dependiendo de la función que desempeñen: activador o inhibidor.

[4] El proyecto Ensembl proporciona información de bases de datos del genoma para vertebrados y otras especies eucariotas y ofrece todos los datos de manera gratuita a través de la red.

- ***description:*** descripción del tfbs.
- ***score:*** grado de similitud entre la secuencia consenso y el tfbs.
- ***cons_seq***: secuencia consenso la cual enlaza el tfbs.

Cpg island

Las *cpg island* conforman aproximadamente un 40% de promotores de los genes de mamíferos. Son regiones donde existe una gran concentración de pares de Cs y Gs enlazados por fosfatos. La "p" en CpG representa que están enlazados por un fosfato y simboliza un conjunto de repeticiones de las bases CG que están cerca del promotor y son objetivos para la metilación que es otra manera de alterar la expresión del gen. La definición formal de una isla CpG es una región con al menos 200 pares de bases, con un porcentaje de GC mayor de 50 y con un promedio de CpG observado/esperado mayor de 0,6.Tiene un atributo que la define:

- ***cg_percentage:*** representa el porcentaje de GC en el elemento.

Triplex

Los *triplex* son secuencias de ADN que se intercalan en la doble hélice de ADN de las células, pasando a tener éste tres cadenas, de tal manera que se impide el proceso de transcripción causando un efecto negativo en el individuo.

Transcript regulator

La clase *transcript regulator* representa regiones reguladoras del transcrito. Existen muchas especializaciones de elementos reguladores del transcrito, pero por razones de simplificación en este modelo se representan únicamente dos: *mirna target* y *splicing regulator*.

Mirna target

La clase Mirna target representa una región reguladora del transcrito a la que se unirá post-transcripcionalmente un miRNA.

Splicing regulator

La clase *splicing regulator* representa un elemento regulador del transcrito que regula el proceso de splicing y tiene dos atributos:

- ***type:*** indica el tipo de regulación y puede tomar dos valores, desactivar (silencer) o promover (enhancer).
- ***regulated_element:*** indica cual es el elemento regulado si se trata de un intrón o un exón.

Conserved región

La clase *conserved region* representa las regiones conservadas dentro del cromosoma, regiones que normalmente tienden a ser no codificantes, es decir, se mantienen intactas tras el proceso de evolución entre las especies. Tiene un atributo que la define:

- ***score:*** representa el grado de conservación de la región (puede tomar dos valores: o un valor estadístico indicando la probabilidad o un valor extraído de una fórmula).

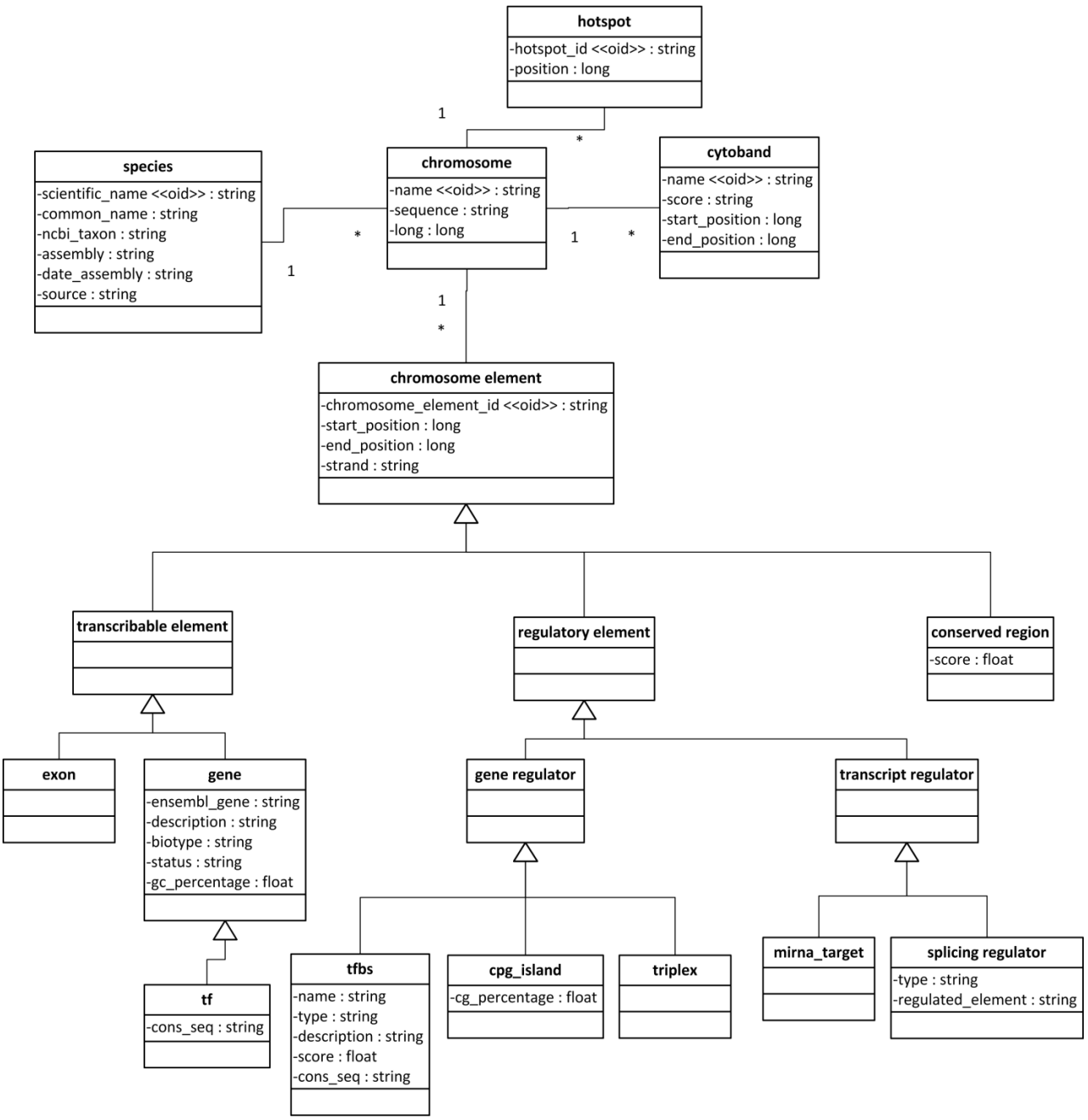


Fig. 9 Vista estructural del modelo conceptual

4.1.2 Vista de transcripción

Un gran número de genes expresan su funcionalidad a través de la producción de proteínas. La vista transcripción (Fig. 10) muestra los componentes y conceptos relacionados con la síntesis de proteínas.

La secuencia de ADN que se transcribe en una molécula de ARN codifica al menos un gen, y si el gen transcrito codifica para una proteína, el resultado de la transcripción es RNA mensajero (mRNA), el cual será entonces usado para crear esa proteína a través de un proceso de traducción.

Después de la transcripción, tiene lugar una modificación en el ARN llamada splicing, en la que los intrones son borrados y los exones se unen. Pero en muchos de los casos, el proceso de splicing no es “perfecto” y puede variar la composición de los exones del mismo ARN mensajero. Este fenómeno es entonces llamado

splicing alternativo. El splicing alternativo puede ocurrir de muchas maneras. Los exones pueden ser extendidos o saltados, o los intrones pueden ser retenidos.

A continuación se describen las clases que forman la vista:

Transcript

La clase *transcript* representa los diferentes transcritos que presenta un gen. Estos transcritos están formados por una serie de exones que lo forman. Como se ha comentado antes, existe un fenómeno llamado splicing alternativo que permite la combinación de diferentes exones, e incluso en algunos casos algún intron, formando diferentes transcritos. Tiene dos únicos atributos:

- ***transcript_id:*** identificador interno del transcrito.
- ***biotype:*** cada transcrito puede tener una función diferente representada con este atributo, que puede tomar el valor de: protein coding, trna, rrna, mirna, sirna, pirna, antisense, long noncoding, riboswitch, shrna, snorna, mitocondrial o otros.

La relación entre la clase transcrito y exón (perteneciente a la vista estructural) indica qué exones forman cada uno de los transcritos.

Protein coding

La clase *protein coding* es una especialización de la clase *transcript* y que, como su propio nombre indica, representa el primero de los biotipos citados arriba. Ya que este tipo de transcritos sintetiza para una proteína se le añaden nuevos atributos:

- ***start_position_ORF:*** indica la posición de inicio de la secuencia codificada.
- ***end_position_ORF:*** indica la posición final de la secuencia codifica.

A partir de un transcrito de tipo *protein coding*, muchas proteínas diferentes pueden ser sintetizadas.

Protein

La clase *protein* representa las miles de proteínas que se sintetizan a partir de un transcrito y es identificada por los siguientes atributos:

- ***name:*** nombre e identificador de la proteína.
- ***accesion:*** identificador que presenta la proteína en la fuente de datos de la cual ha sido extraída.
- ***sequence:*** la secuencia de la proteína.
- ***source:*** fuente de datos de la cual se ha extraído la información

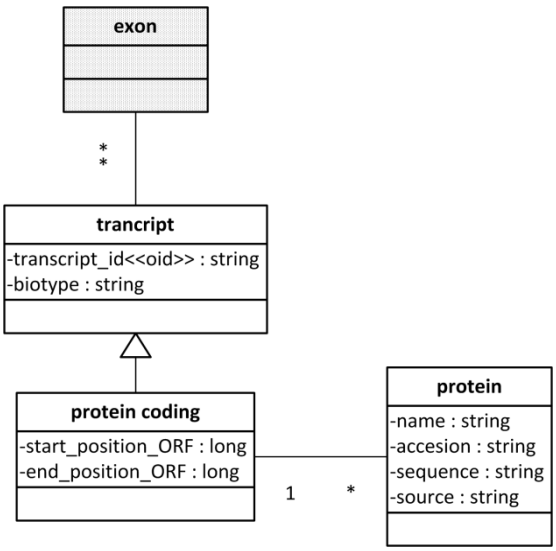


Fig. 10 Vista de transcripción del modelo conceptual

4.1.3 Vista de variaciones

La vista de variaciones (Fig. 11) modela el conocimiento relacionado con las diferencias encontradas en la secuencia de ADN de diversos individuos. A continuación se detallan las clases que la forman y la explicación de cada una de ellas:

Variation

La clase *variation* es la clase principal en esta vista, representa, como su propio nombre indica, las variaciones existentes en la cadena de ADN. Sus atributos son:

- ***variation_id:*** identificador interno de la variación.
- ***description:*** proporciona una descripción de la variación.
- ***id_variation_db:*** identificador que proporciona la fuente de datos de la cual se ha extraído la variación.

Las variaciones se especializan siguiendo dos criterios: la precisión en su descripción (ISA *description*) y su frecuencia (ISA *frequency*).

En la jerarquía *frequency*, si la variación se presenta en más del 1% de la población o es un caso puntual, una variación puede estar especializada en dos clases: *mutation* y *polimorphysm*.

En la jerarquía *description*, una variación puede estar especializada en dos clases: *precise* e *imprecise*, dependiendo de si se conocen datos al respecto de su posición.

Por otra parte, cabe destacar que la clase *variation* enlaza esta vista con la vista de la estructura del genoma, mediante una relación entre la clase *variation* y la clase c*hromosome element* que indica que una variación es un elemento que forma parte de un cromosoma.

Mutation

La clase *mutation*, especialización en *frequency*, hace referencia a las variaciones con efecto patológico que se encuentran en un bajo porcentaje de la población, es decir, en menos del 1%. Normalmente son nocivas, por lo que o la célula recobra su bienestar mediante sus mecanismos de recuperación o la persona que

padece una grave mutación no suele sobrevivir para pasarla genéticamente de generación en generación; este es el principal motivo de su poca extensión. Un tipo de estas mutaciones son las que se producen en el gen CFTR, que provocan que las personas que poseen ambas copias del gen dañado sufran fibrosis quística, enfermedad que daña muchos de los órganos del cuerpo dependiendo de la agresividad de las mutaciones heredadas. Es una de las enfermedades genéticas más frecuentes en la raza caucasiana con una incidencia en la población española, según recientes estudios de cribado neonatal, de aproximadamente 1/5.000 nacidos vivos.

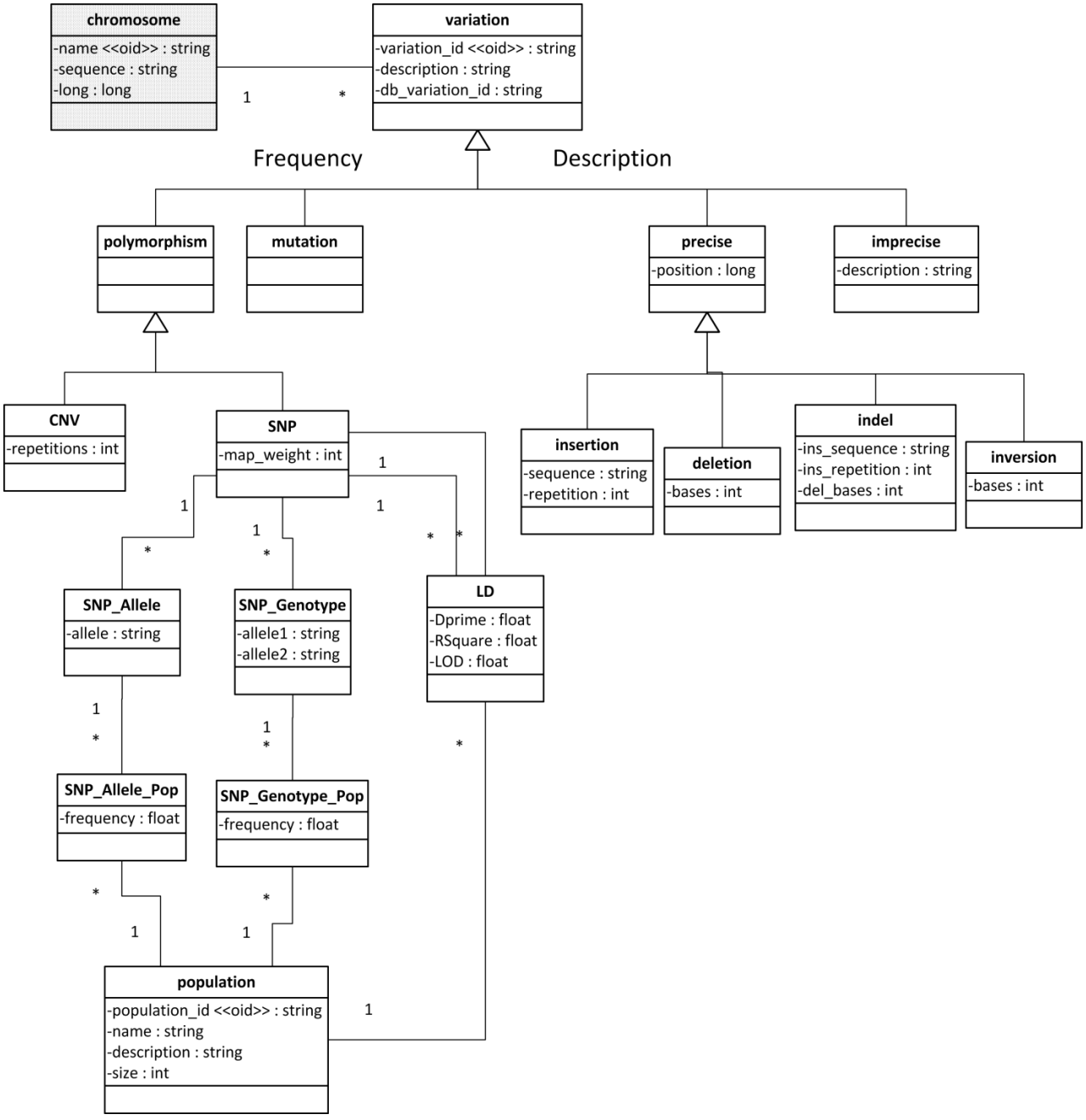


Fig. 11 Vista de variaciones del modelo conceptual

Polymorphism

La clase *polymorphism*, especialización en *frequency*, describe las variaciones que aparecen en más del 1% de la población y normalmente no tienen un diagnostico maligno, por lo que se heredan de generación en

generación. Este tipo de variaciones, puede especializarse en dos tipos: *CNV* (Copy Number Variation) y *SNP* (Single Nucleotide Polymorphism).

CNV

Un *cnv* (copy number variation) es definido como una variación que consiste en la repetición un cierto número de veces o el borrado de una pequeña región de la secuencia de ADN, se representa con el atributo:

- ***repetitions:*** este atributo almacena el número de veces que la secuencia se repite o se borra.

SNP

Un *SNP* es un polimorfismo que tiene lugar cuando un único nucleótido dentro del genoma difiere de lo habitual entre individuos de la misma especie agrupados por poblaciones. Constituyen hasta el 90% de todas las variaciones genómicas humanas, y aparecen cada 1300 bases en promedio a lo largo del genoma humano. Dos tercios de los SNP corresponden a la sustitución de una citosina (C) por una timina (T). Estas variaciones en la secuencia del ADN pueden afectar a la respuesta de los individuos a enfermedades, bacterias, virus, productos químicos, fármacos, etc. El atributo que los define es:

- ***map_weight:*** las veces que dicho SNP ha sido mapeado en la muestra del genoma de un individuo.

Un SNP es un cambio de un único nucleótido en una posición del genoma pero a su vez puede proporcionar datos relevantes: los distintos valores que puede tomar el SNP teniendo en cuenta un único alelo (*SNP_Allele*) y las diferentes combinaciones de valores que puede tomar el SNP teniendo en cuenta los dos alelos (*SNP_Genotype*). Además, existe más información de interés con respecto a los SNPs así como el linkage disequilibrium (*LD*) y que se describe como marcador que indica la relación existente entre dos SNPs dentro de una población.

SNP_Allele

La clase *SNP_Allele* representa los diferentes valores que puede tomar un SNP teniendo en cuenta un solo alelo y se define mediante el atributo:

- ***allele:*** este atributo indica el valor que puede tomar el alelo en cada caso. Su dominio es {A,T,G,C}.

SNP_Genotype

La clase *SNP_Genotype* representa los diferentes valores que pueden tomar el par de alelos de cada individuo en la posición del SNP teniendo en cuenta las dos hebras. Sus atributos son:

- ***allele1:*** este atributo indica el valor que puede tomar el alelo en una hebra. Su dominio es {A,T,G,C}.
- ***allele2:*** este atributo indica el valor que puede tomar el alelo en la otra ebra. Su dominio es {A,T,G,C}.

Como se ha comentado en la descripción de SNP, cada uno de ellos está directamente relacionado con varias poblaciones, por lo que las dos clases, *SNP_Allele* y *SNP_Genotype* tienen relación con varias poblaciones en cada caso. Para proporcionar información sobre la frecuencia de aparición de cada SNP en diferentes poblaciones, bien sea a nivel alélico o a nivel genotípico, se crean también las clases *SNP_Allele_Pop* y *SNP_Genotype_Pop*.

SNP_Allele_Pop

La clase *SNP_Allele_Pop* representa la frecuencia en la que cada SNP, teniendo únicamente en cuenta un alelo, aparece en cada población El atributo que la define es:

- ***frequency:*** frecuencia con la que cada SNP aparece en diversas poblaciones.

SNP_Genotype_Pop

La clase *SNP_Allele_Pop* representa la frecuencia en la que cada SNP aparece en cada población teniendo únicamente en cuenta los dos alelos. El atributo que la define es:

- ***frequency:*** frecuencia con la que cada SNP aparece en diversas poblaciones.

Population

La clase *population* representa conjuntos de individuos con características comunes. Los atributos que definen la clase son:

- ***name:*** nombre e identificador de cada población.
- ***description:*** descripción de cada población.
- ***size:*** cantidad de individuos pertenecientes a una población.

LD

Otro concepto modelado que hemos nombrado anteriormente es el linkage disequilibrium o *LD* que es un marcador que define la relación existente entre dos SNPs en una población específica. Los atributos que describen esta clase son:

- ***Dprime, Rsquare y LOD:*** los tres son valores matemáticos de ámbito muy biológico a los que no vamos a entrar en detalle en esta tesis.

Precise

La clase *precise*, especialización del tipo ISA *Description*, representa las variaciones detectadas con posición conocida dentro del cromosoma en la secuencia de ADN. Su definición viene proporcionada principalmente por el atributo:

- ***position:*** posición en la que se encuentra la variación dentro de la secuencia del cromosoma.

La clase Precise se especializa en cuatro nuevas entidades dependiendo de qué tipo de variación haya tenido lugar dentro del genoma: *insertion*, *deletion*, *indel* e *inversion*.

Insertion

La clase *insertion* representa variaciones que consisten en la inserción de una secuencia de nucleótidos un número de veces en la secuencia de ADN del cromosoma. Sus atributos son:

- ***sequence:*** secuencia de nucleótidos insertados en la secuencia.
- ***repetition:*** número de veces que se repite la secuencia insertada.

Deletion

La clase *deletion* representa variaciones que consisten en el borrado de un número de nucleótidos en la secuencia de ADN del cromosoma. El atributo que la define es:

- ***bases:*** número de nucleótidos borrados en la secuencia.

Indel

La clase *indel* representa variaciones consistentes en inserciones y borrados a la vez en la secuencia de ADN del cromosoma. Los atributos que la definen son:

- ***ins_sequence:*** secuencia de nucleótidos insertados en la secuencia.
- ***ins_repetition:*** número de veces que se repite la secuencia insertada.
- ***del_bases:*** número de nucleótidos borrados.

Inversion

La clase *inversion* representa variaciones que invierten el orden de una secuencia de nucleótidos en la secuencia del cromosoma. Es definida por:

- ***bases:*** número de nucleótidos invertidos en la secuencia.

Imprecise

La clase *imprecise* dentro de la jerarquía de descripción representa variaciones cuya posición es desconocida dentro de la secuencia de ADN. La única información que se conoce es una descripción en lenguaje natural y es su único atributo:

- ***description:*** descripción de la variación en lenguaje natural.

4.1.4 Vista de rutas metabólicas

En bioquímica, las rutas metabólicas (pathways) (Fig. 12), son una serie de reacciones químicas que ocurren dentro de una célula. Esta composición de procesos viene representada en el esquema por las siguientes clases:

Event

La clase *event* es la clase principal y es la que representa la combinación de procesos existentes en el organismo. Para caracterizar cada proceso se tiene los siguientes dos atributos:

- ***event_id:*** identificador interno del evento
- ***name:*** nombre que tiene el evento.

Además, la clase Event se especializa en dos clases dependiendo de la cantidad de procesos que lo formen: Process y Pathway.

Process

La clase *process* representa un único proceso atómico o dicho en otras palabras un proceso de tipo simple.

Pathway

La clase *pathway* representa un proceso complejo formado por una secuencia de otros procesos de tipo complejo o simple.

La asociación entre las clases *pathway* y *event* representa la composición de un pathway, es decir, proporciona información sobre que otros eventos anteriores forman parte de dicho pathway.

Por otra parte la relación existente de la clase *event* consigo misma cuyas aristas poseen el nombre de Pre y Post nos permite conocer el orden de la composición de los eventos dentro del pathway. El que va primero toma el valor de Pre y el que lo sigue toma el valor de Post.

Por otra parte y siguiendo con la descripción de la vista, una entidad puede participar en un proceso de varias maneras:

- Siendo el químico principal, es decir la entrada necesaria para ese proceso, a veces también llamada sustrato.
- Como resultado del proceso o en otras palabras la salida o producto final.
- Siendo un regulador del proceso, de los cuales se pueden distinguir dos tipos: activador e inhibidor. Por otro lado, existe un tipo especial de elemento regulador, la catálisis, de la que a veces se desconoce información al respecto pero es sabida su existencia en algunos procesos.

Este conocimiento es modelado con las siguientes clases: *takes_part, input, output* y *regulator*.

Takes_part

La clase *takes_part* es una clase genérica que define de que manera una entidad participa dentro de uno o varios procesos. Tiene un atributo que proporciona una pequeña descripción:

- ***notes:*** comentario sobre la relación entre las entidades que toman parte en cada proceso.

Se especializa en tres entidades diferentes dependiendo de la manera en la que dicha entidad participe en dicho proceso: *input*, *output* y *regulator*.

Input

La clase *input* representa la entidad de entrada a un proceso. Su definición viene dada por el atributo:

- ***stoichiometry:*** cantidad de la entidad que interviene en el proceso.

Output

La clase *output* representa el resultado final del proceso. Como la clase Input, su definición viene dada por el atributo:

- ***stoichiometry:*** cantidad de la entidad que interviene en el proceso.

Regulator

La clase *regulator* como su propio nombre indica los procesos reguladores existentes en las partes intermedias de la reacción, para definirlo se utiliza el atributo:

- ***type:*** se usa para distinguir de qué tipo de regulación se trata y puede tomar dos valores: inhibidor y activador.

Catalysis

La clase *catalysis*, define el proceso por el cual se aumenta o disminuye la velocidad de una reacción química. Es un tipo especial de regulador de pathways que ha sido modelada aparte debido al hecho de que se tiene constancia de que forma parte de muchos procesos pero en algunos de ellos el catalizador es desconocido. En los casos en los que el catalizador es conocido, una enzima es asociada al correspondiente proceso. Como atributo de la clase tiene:

- ***EC number:*** los números EC (Enzyme Commission numbers) son un esquema de clasificación numérica para las enzimas, basado en las reacciones químicas que catalizan. En realidad los números EC codifican reacciones catalizadas por enzimas. Enzimas diferentes (por ejemplo que procedan de organismos diferentes) que catalicen la misma reacción recibirán el mismo número EC. Cada código de enzimas consiste en las dos letras EC seguidas por 4 números separados por

puntos. Estos números representan una clasificación progresivamente más específica. Por ejemplo, la enzima tripéptido aminopeptidasa tiene el código EC 3.4.11.4.

Enzime

La clase *enzime*, es una especialización de proteína que cataliza reacciones químicas. Una enzima hace que una reacción química que es energéticamente posible pero que transcurre a una velocidad muy baja, sea cinéticamente favorable, es decir, transcurra a mayor velocidad que sin la presencia de la enzima. Está asociada con el proceso de catálisis para determinar cuál es el catalizador en caso de ser conocido. Como atributos contiene:

- ***name:*** las enzimas son usualmente nombradas de acuerdo a la reacción que producen. Normalmente, el sufijo "-asa" es agregado al nombre del sustrato (p. ej., la lactasa es la enzima que degrada lactosa) o al tipo de reacción (p. ej., la ADN polimerasa forma polímeros de ADN).

Entity

La clase *entity* es la clase genérica que representa el tipo de entidades que pueden participar en un proceso de un pathway. Como atributo contiene:

- ***entity_id:*** identificador interno de la clase *entity*.
- ***name:*** atributo genérico que proporciona información acerca del nombre de la entidad.

La clase *entity* se especializa en cuatro clases dependiendo del tipo de entidad: *complex*, *polymer*, *simple* y *entitySet.*

Complex

La clase *complex* representa entidades que están formadas por la combinación de otras entidades más simples. Se define mediante el atributo:

- ***detection_method:*** este atributo indica la técnica usada para determinar cómo se ha formado la entidad.

Component

La clase *component* representa de que manera una entidad *complex* está formada por sus entidades más simples. Los atributos que contiene esta clase son:

- ***stoichiometry:*** permite conocer cuanta cantidad del complejo está formado por cada uno de sus componentes.
- ***interaction:*** permite conocer como el complejo ha sido formado a partir de cada uno de sus componentes.

Polymer

La clase *polymer* representa entidades que son generadas por la repetición de alguna entidad, bien sea compleja o simple. Sus atributos son:

- ***min:*** representa el rango de repeticiones mínimo de la entidad que forma el polímero.
- ***max:*** representa el rango de repeticiones máximo de la entidad que forma el polímero.

Simple

La clase *simple*, representa las entidades más simples que pueden formar parte de un proceso, como por ejemplo: gen, ARN, proteína, aminoácido, nucleótido, entidad básica (agua, fósforo, etc.).

EntitySet

La clase *entitySet* representa un conjunto de entidades que participan de manera habitual conjuntamente en algunos procesos, lo que permite reducir la cantidad de procesos similares existentes.

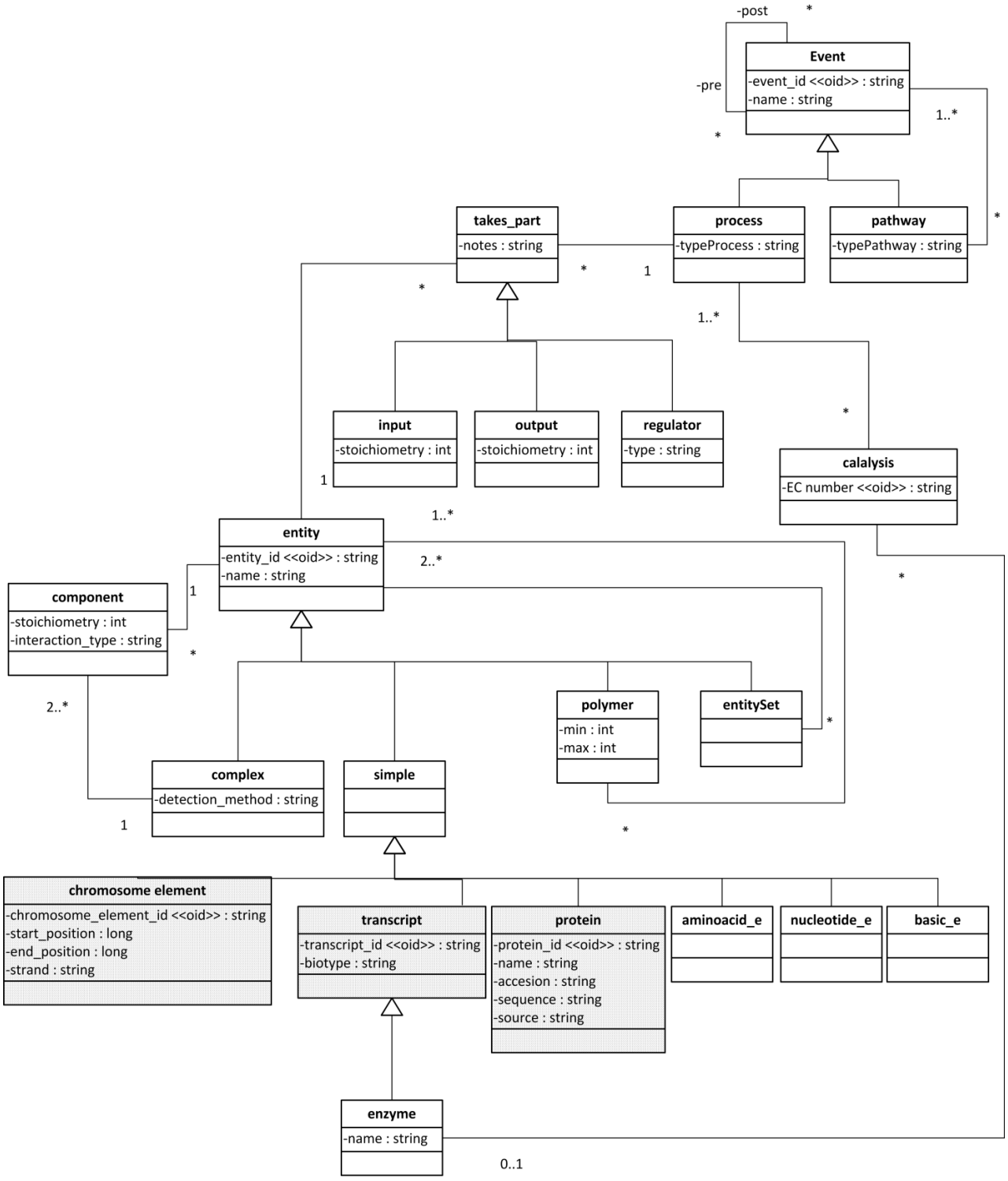


Fig. 12 Vista de rutas metabólicas del modelo conceptual

4.1.5 Vista de fuentes de datos y bibliografía

Esta vista (Fig. 13) representa la información sobre las fuentes de datos de las que se ha extraído la información que se va a almacenar en la base de datos, así como una serie de documentos bibliográficos de consulta para quien desee obtener más información con respecto a algún aspecto aquí definido.

Para mantener información sobre las fuentes de las cuales se ha obtenido la información, esta vista incluye las siguientes clases:

Data_bank

La clase *data_bank* representa la fuente de datos de la cual se extrae la información de cada uno de los elementos del modelo. Los atributos que la describen son:

- ***name:*** nombre de la fuente de datos.
- ***description:*** descripción de la fuente de datos.

Data bank version

Esta clase representa la versión de cada una de las bases de datos que se han utilizado y en qué fecha dichas bases de datos han sido actualizadas. Sus atributos son:

- ***release:*** versión de la fuente de datos.
- ***date:*** fecha en la que se actualizó por última vez la fuente de datos consultada.

La clase *variation* se relaciona con esta clase representando su procedencia.

Element data bank

La clase *element data bank* permite relacionar cada uno de los elementos del cromosoma con la fuente de datos de la que han sido extraídos y su versión. Tiene un atributo:

- ***source_identification:*** este atributo indica el identificador que proporciona cada una de las fuentes a los elementos del cromosoma.

Data Bank Entity Identification

Permite relacionar cada una de las entidades que forman las rutas metabólicas con la fuente de datos y la versión de la cual se ha extraído la información. Tiene un atributo asociado:

- ***source_identification:*** este atributo indica el identificador que proporciona cada una de las fuentes a las entidades que forman los pathways.

Por último, para mantener información sobre la bibliografía asociada a cada elemento, la vista incluye también las siguientes clases:

Bibliography DB

La clase *bibliography DB* representa las distintas fuentes de datos de la web de las que se extraen las publicaciones científicas.

- ***Bibliography Name DB:*** nombre de la base de datos de la que se extraen las publicaciones científicas.
- ***URL:*** dirección web de la base de datos de las que se extraen las publicaciones.

Bibliography reference

Bibliography reference proporciona información sobre los artículos relacionados con cada uno de los elementos almacenados si se dispone de ella. Tiene cinco atributos:

- ***bibliography_reference_id:*** identificador interno de las referencias bibliográficas.
- ***title:*** título del artículo.
- ***authors:*** autores que han escrito el artículo.
- ***abstract:*** resumen del artículo.
- ***publication:*** fecha en la cual se ha publicado el artículo.
- ***pubmed_id:*** identificador que la base de datos de pubmed proporciona al artículo.

Las clases *variation*, *process*, *transcript*, *chromosome element* y *entity* se relacionan con esta clase.

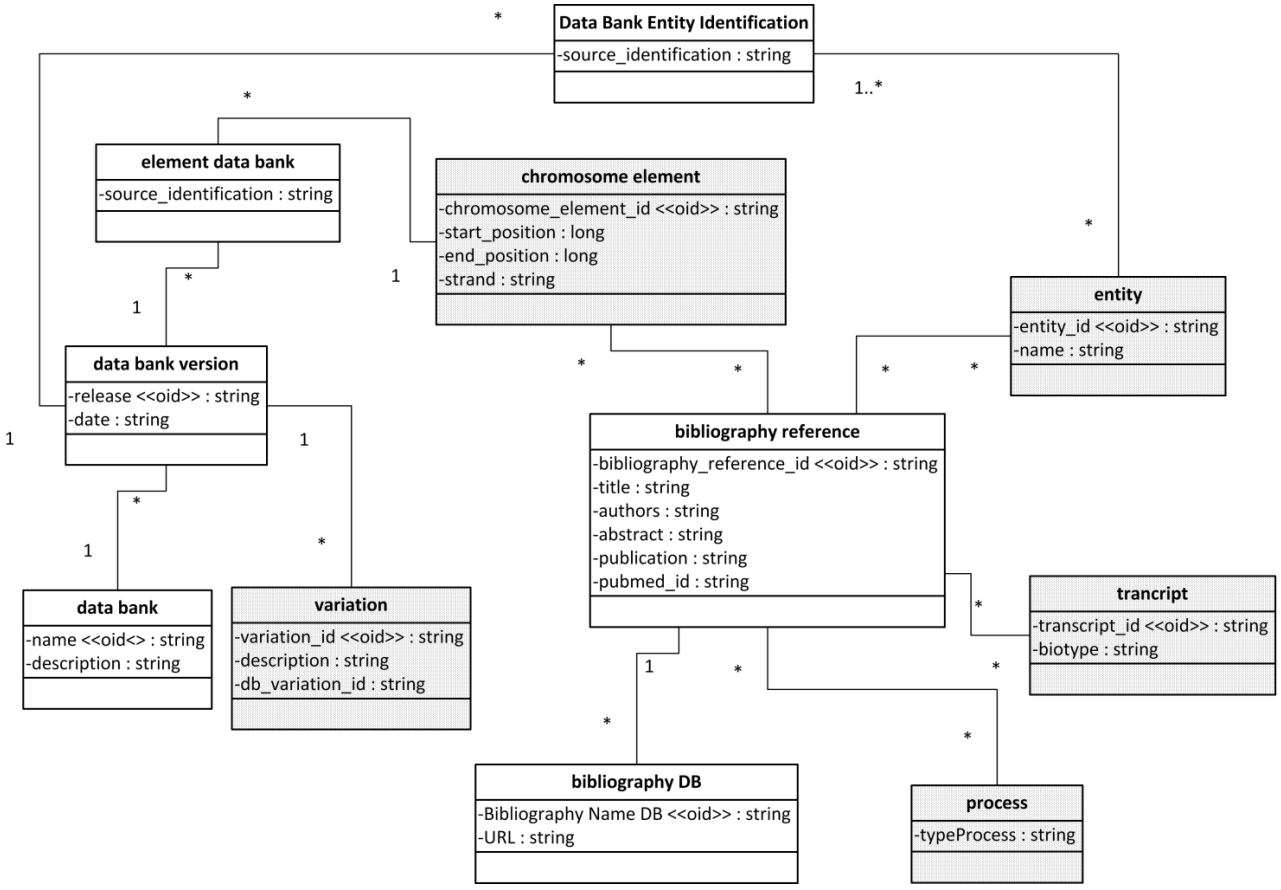


Fig. 13 Vista fuentes de datos y bibliografía del modelo conceptual

4.1.6 Modelo conceptual del genoma

Una vez presentadas una por una las vistas que componen el modelo conceptual del genoma, se muestra una imagen completa de él (Fig. 14):

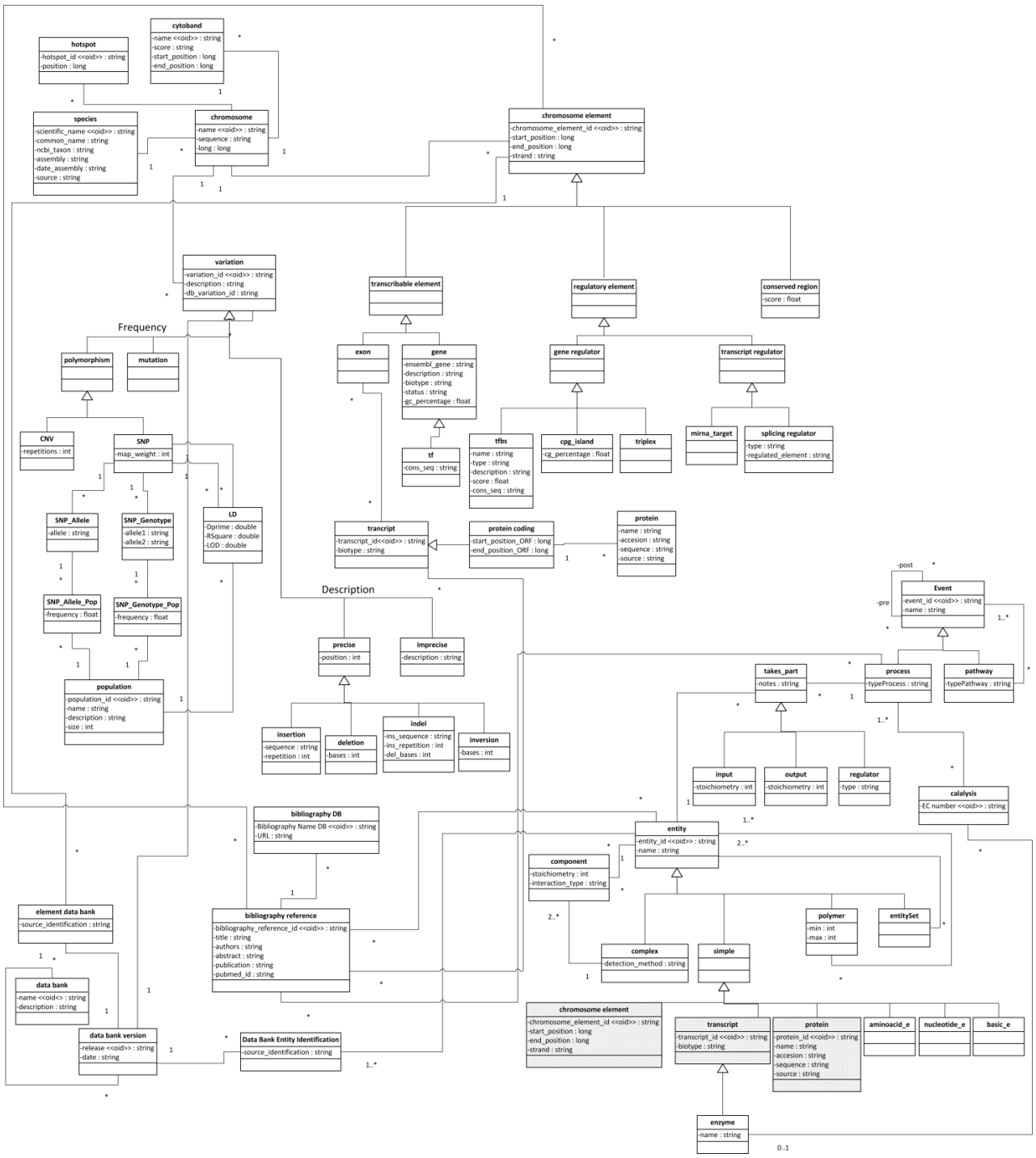


Fig. 14 Modelo conceptual del genoma

4.2 Diseño de la base de datos [5,6]

La traducción del modelo conceptual al esquema de la base de datos, es prácticamente automática. Pero obviamente son dos modelos a diferentes niveles de abstracción, por lo que las diferencias entre ellos aparecen. El modelo conceptual representa el conocimiento del dominio desde el punto de vista de los expertos, por otro lado el esquema de base de datos (modelo lógico) está enfocado al almacenamiento de datos y a las operaciones de recuperación de información. Por esta razón es importante considerar la tecnología que mas se adapta a las necesidades de cada entorno para ser usada en la implementación de la base de datos y tener en cuenta pequeños detalles de representación para mejorarla.

El esquema de la base de datos (GDB) ha sido implementado usando mySql relacional. Los sistemas de bases de datos relacionales son ampliamente usados en la actualidad, que proporcionan la tecnología necesaria para tratar con grandes volúmenes de datos, facilitan una organización de los datos muy estructurada y no redundante y utilizan el lenguaje estándar SQL para realizar las operaciones de recuperación y almacenamiento.

El diseño e implementación de la base de datos se ha realizado manualmente utilizando técnicas de generación de bases de datos a partir de modelos conceptuales. Esto es debido a que el desarrollo del sistema de información no ha sido realizado de manera lineal, sino que el modelo conceptual se iba diseñando poco a poco por vistas, ampliándolo cuando la vista que se estaba tratando en cada momento se daba por finalizada. Debido a este motivo, el diseño de la base de datos fue tomando el mismo camino y paso a paso conforme el modelo conceptual avanzaba, la base de datos iba complementándose y cargándose con su información correspondiente.

Para explicar las diferencias entre el modelo conceptual y el esquema de la base de datos, se va a utilizar la misma estrategia seguida en el capitulo anterior, es decir, dividirlo por vistas, las mismas vistas que se han utilizado para describir el modelo conceptual.

4.2.1 Vista estructural

Una de las principales diferencias entre el modelo conceptual y el esquema de la base de datos (Fig. 15), es que en dicho esquema no aparecen todas las clases especializadas que aparecen en el modelo conceptual. Estas clases representan conceptos existentes en el dominio pero que no contienen información específica,

[5] La implementación de la base de datos se ha realizado con el programa MySQLWorkbench, que utiliza la siguiente notación gráfica:

- Atributos:
  - Clave primaria
  - Clave ajena
  - Valor no nulo
- Cardinalidad:
  - Extremo de relación de cardinalidad 1…1
  - Extremo de relación de cardinalidad 0…*
  - Extremo de relación de cardinalidad 0…1
- Relaciones:
  - Relación entre dos tablas entre las que la clave primaria de una de ellas es la clave ajena de la otra.
  - Relación entre dos tablas entre las que la clave primaria de una de ellas es a su vez clave primaria de la otra.

[6] Los nombres de las tablas y atributos de la base de datos aparecen en letra cursiva.

por lo que no es necesario representarlas. Algunos ejemplos de este tipo de clases son: *transcriptable element*, *exon*, *mirna*, *regulatory element*, *gene regulator*, *triplex*, *transcript regulator* and *mirna target*. Todas estas clases se eliminan y quedan representadas en la tabla *chromosome_element* del modelo lógico con el atributo *type* añadido para diferenciar de qué tipo de elemento cromosómico se trata.

Por otro lado, debido a que la cantidad de pares de bases que contiene cada cromosoma es muy elevada, por razones de eficiencia de la base de datos se decide dividir la secuencia completa de cada uno de ellos en porciones más pequeñas de 1000 pares de bases. Para representar este consenso, se añade al esquema, una nueva tabla llamada *chromosome seq* cuya función es almacenar cada una de esas porciones en las que se ha dividido el cromosoma (atributo *sequence*) y la posición de cada una de ellas en la secuencia completa para poder unirlas (atributo *chunk_rank*).

4.2.2 Vista de transcripción

La primera diferencia que se encuentra en esta vista entre el modelo conceptual y el esquema de la base de datos (Fig. 16) es la inserción de una nueva tabla al esquema, *chromosome_element_has_transcript*, que representa la relación de cardinalidad muchos a muchos entre las clases *exon* y *transcript* del modelo conceptual. Como se puede observar, la relación en el esquema de la base de datos no es entre las tablas *exon* y *transcript*, sino entre las tablas *chromosome element* y *transcript*. Esto es debido a que la clase *exon* de la vista estructural no añade información específica, como se ha comentado en la vista anterior, y queda representada en la tabla *chromosome element* del esquema de la base de datos mediante el atributo *type*. Como se puede observar en la figura 16, se añade una nueva restricción de integridad en lenguaje natural al esquema de la base de datos que indica que dicha relación entre un elemento de un cromosoma y un transcrito se puede realizar únicamente cuando el atributo *type* de la tabla *chromosome element* tome el valor *exon*.

Por otra parte, debido a la relación de especialización entre las clases *transcript* y *protein coding* en el modelo conceptual, en el esquema lógico si se representasen las dos tablas tendrían una relación de cardinalidad 1-1, por lo que se decide prescindir de la tabla *protein coding* e incluir sus atributos, *start_position_ORF* y *end_position_ORF*, dentro de la tabla *transcript*.

Además, a la tabla *transcript* se le añade también un atributo más llamado *gene_name*. Este atributo es derivado del esquema y sirve básicamente para mejorar el rendimiento de la base de datos a la hora de obtener a que gen pertenece cada uno de los transcritos.

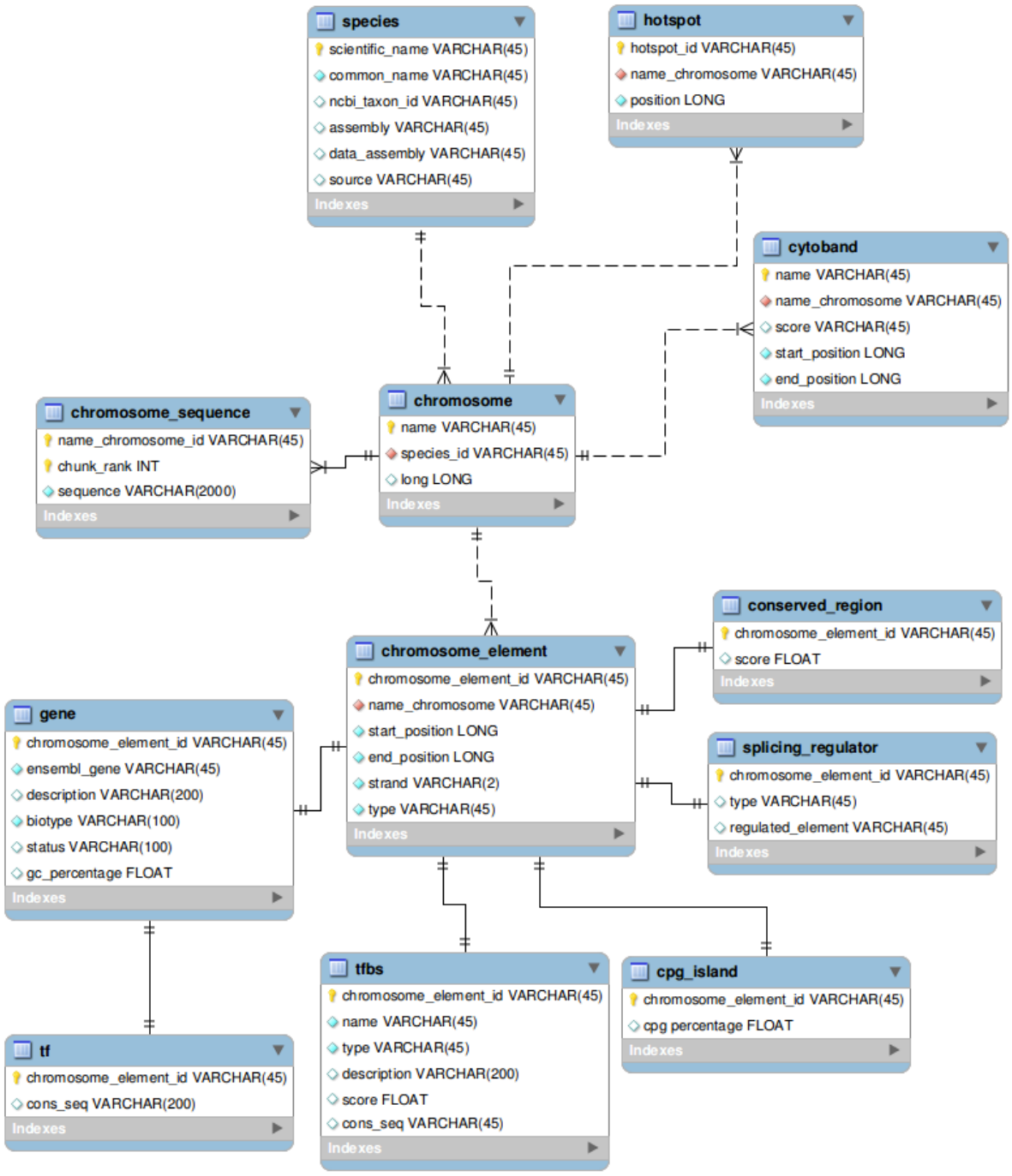


Fig. 15 Vista estructural de la base de datos

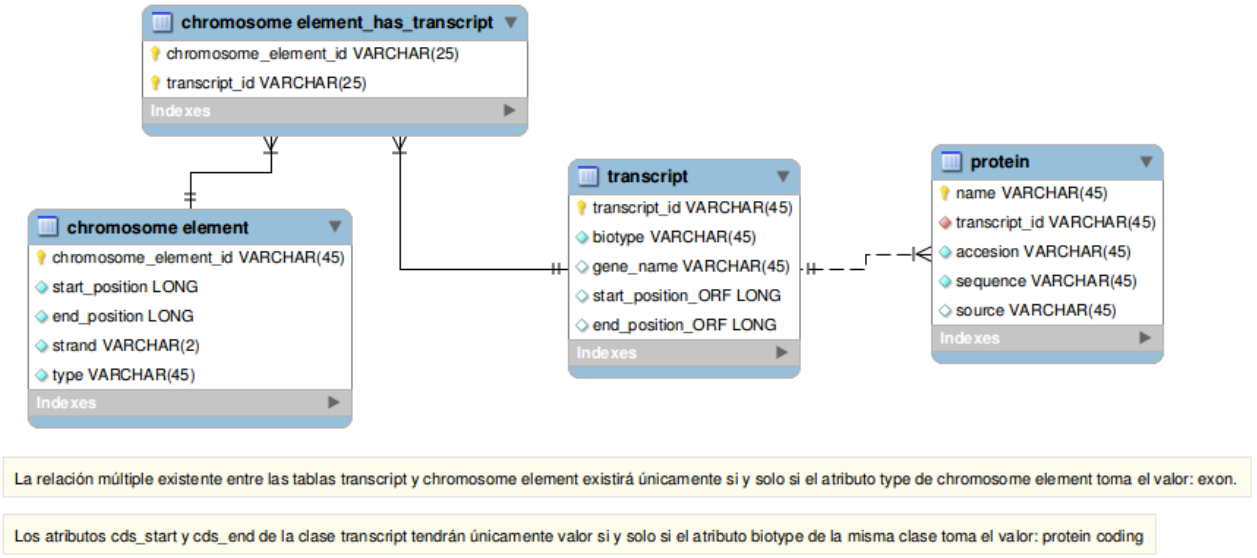


Fig. 16 Vista de traducción de la base de datos

4.2.3 Vista de variaciones

Una de las principales diferencias en esta vista entre el modelo conceptual y el esquema de la base de datos (Fig. 17), como se ha comentado antes, es que en el esquema de la base de datos no aparecen todas las clases especializadas que aparecen en el modelo conceptual. Este tipo de clases en esta vista son: *mutation*, *polymorphism*, *precise* or *imprecise*; estas clases son representadas en la tabla *variation* del modelo lógico.

El atributo *type_phenotypic_effect* es añadido a la tabla *variation* para representar la relación de especialización ISA *frequency* del modelo conceptual o en otras palabras trata de distinguir si la variación es una mutación o un polimorfismo. De esta manera, la clase *polymorphism* desaparece porque no tiene ningún atributo asociado, pero la clase *mutation* que debería desaparecer por el razonamiento explicado arriba permanece en el esquema de la base de datos para almacenar atributos derivados que mejoran el rendimiento de la base de datos. Estos atributos derivados son *cds_mutation* y *aa_mutation* y son obtenidos a partir de otros elementos almacenados en el esquema. *Cds_mutation* es la representación de la mutación en formato HGVS [32] teniendo en cuenta la región codificante, y *aa_mutation* es la representación de la mutación en formato HGVS pero teniendo en cuenta su traducción.

De la misma manera, un nuevo atributo llamado *type_description* es añadido a la tabla variación para representar la relación de especialización ISA *Description* del modelo conceptual. Este atributo es usado para representar las clases *precise* e *imprecise*. Además en ambos casos, debido a que los únicos atributos que tienen son *position* y *description* respectivamente, dichos atributos son añadidos a la clase directamente superior, la clase *variation*. El atributo position en la clase *precise* pasan a ser dos que son *start* y *end* en la tabla *variation*. *Start* representa al atributo position en el modelo conceptual y *end* es un atributo derivado obtenido de la posición y la descripción de la variación. Estos cambios reducen el número de tablas en el esquema de la base de datos. Además de estos, existen más atributos derivados en el esquema de la base de datos, específicamente en la tabla *snp* que son: *allele_string* y *ancestral_allele*. *Allele_string* define los nucleótidos que están involucrados en el cambio, representados en un formato estándar, por ejemplo: A/C, y *ancestral_allele* indica el alelo predominante antes de la evolución. Las razones para estos cambios son mejorar el rendimiento del sistema disminuyendo el tiempo de respuesta durante la explotación.

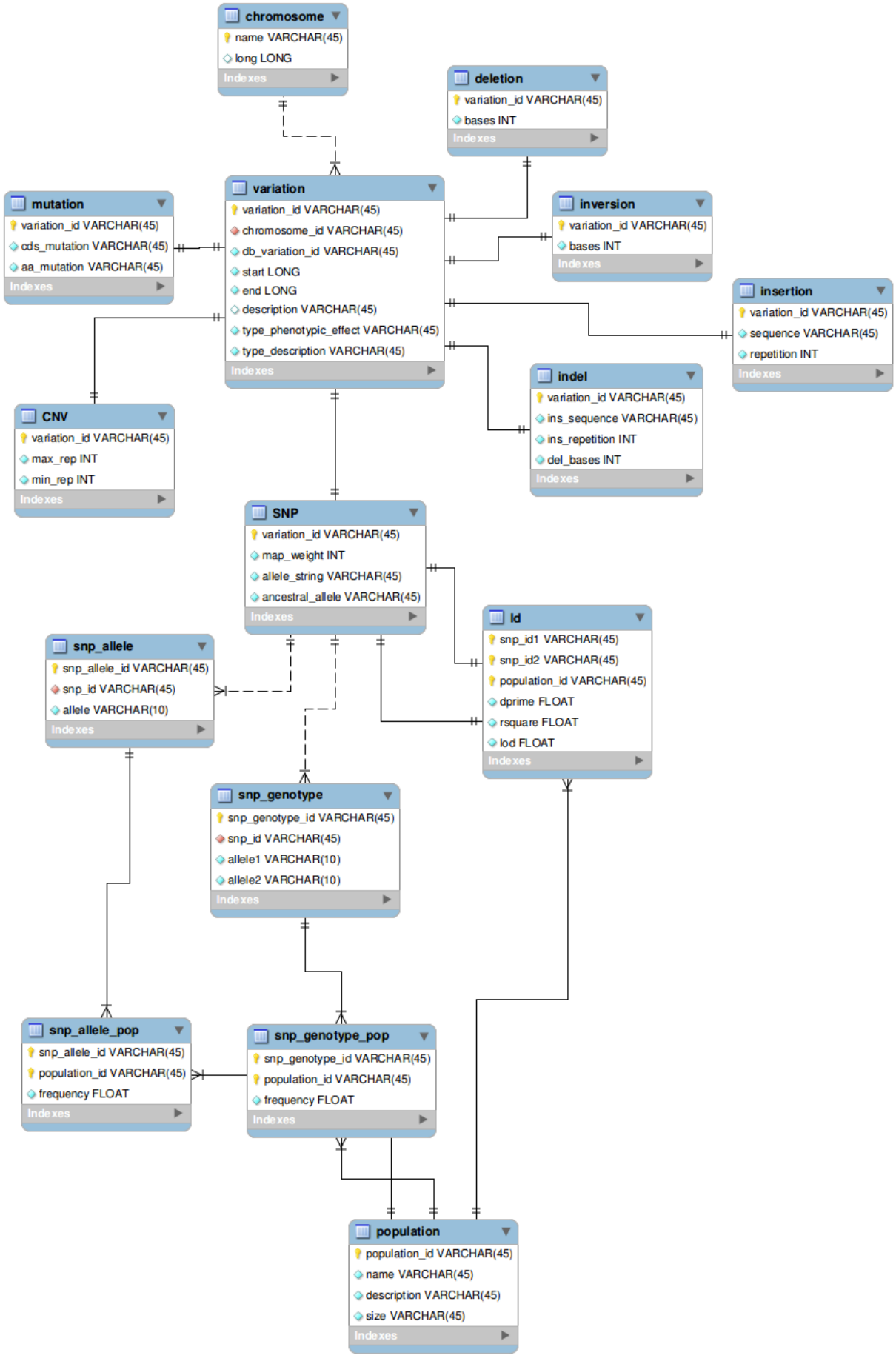


Fig. 17 Vista de variaciones de la base de datos

4.2.4 Vista de rutas metabólicas

Como en el resto de vistas descritas, existen ciertas diferencias entre el modelo conceptual y el esquema de la base de datos (Fig. 18). En primer lugar clases que no aportan información específica y que simplemente sirven para comprender el dominio en el que nos encontramos, las clases *simple* y *entitySet,* estas clases como se ha venido haciendo hasta ahora se eliminan del esquema de bases de datos.

Por otra parte, aparecen nuevas tablas resultantes de las relaciones de asociación de cardinalidad muchos a muchos. Este tipo de relaciones las encontramos entre las clases: *event* y ella misma que añaden una nueva tabla llamada *event_has_event*, *pathway* y *event* que añaden otra nueva tabla llamada *pathway_has_event* y por último, *process* y *catalysis* que añaden otra nueva tabla llamada *process_has_catalysis*.

Las claves primarias de las tablas *chromosome_element*, *protein* y *transcript* se ha decidido que a pesar de ser especializaciones de la clase *entity*, sigan conservando el identificador primario correspondiente al nombre de cada una de ellas seguido de un guión bajo y la palabra id, ya que aporta claridad a la vista completa del esquema de bases de datos.

4.2.4 Vista fuentes de datos y bibliografía

En esta vista, la única diferencia que se observa entre el modelo conceptual y el esquema de base de datos (Fig. 19), son las nuevas tablas que surgen de las relaciones de cardinalidad muchos a muchos entre: *variation* y *bibliography reference*, *entity* y *bibliography reference*, *chromosome element* y *bibliography reference* y *transcript* y *bibliography reference*. Estas tablas son: *variation_has_bibliography reference*, *bibliography reference_has_entity*, *chromosome_element_has_bibliography reference*, y *bibliography reference_has_transcript* respectivamente.

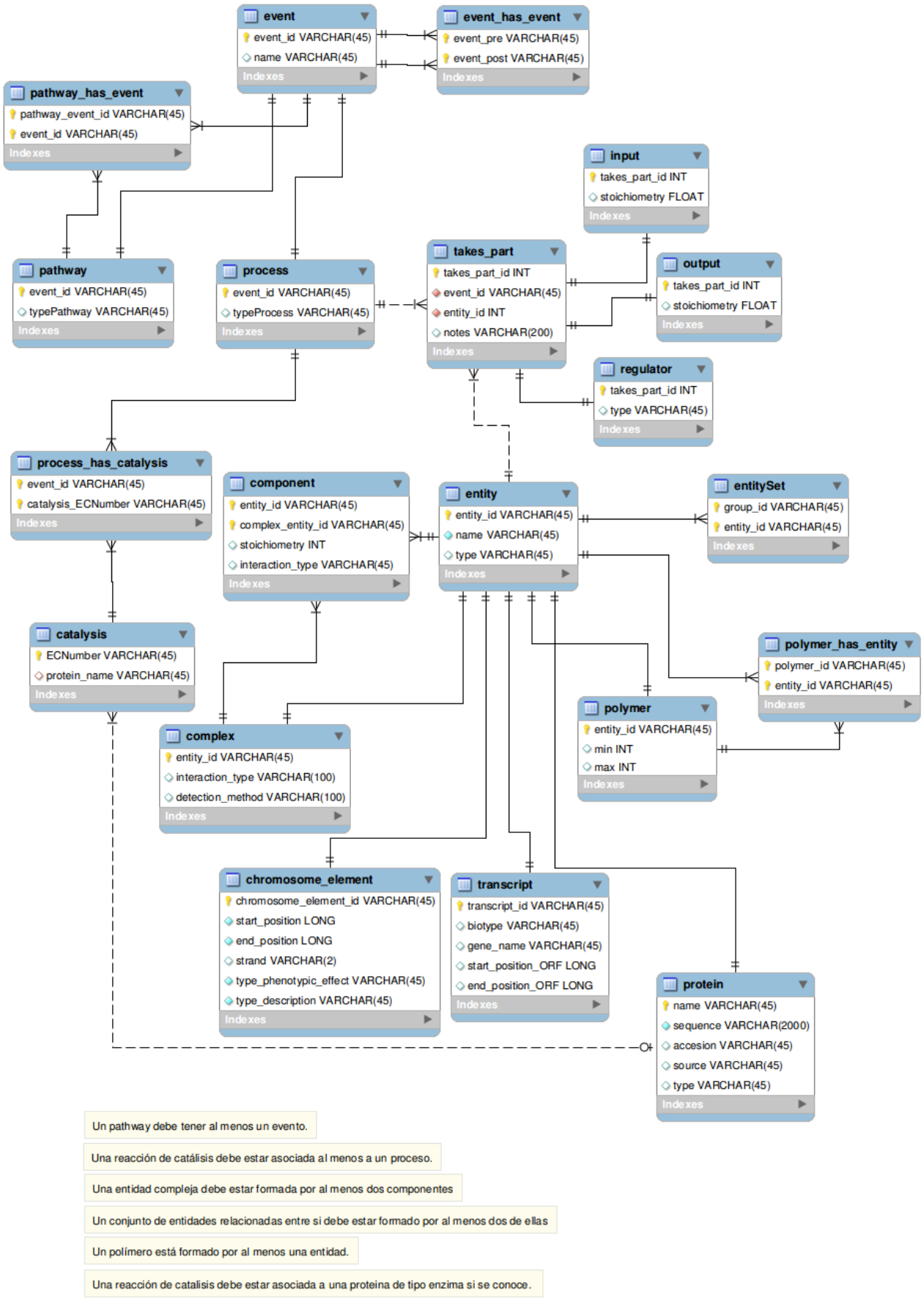


Fig. 18 Vista de rutas metabólicas en la base de datos

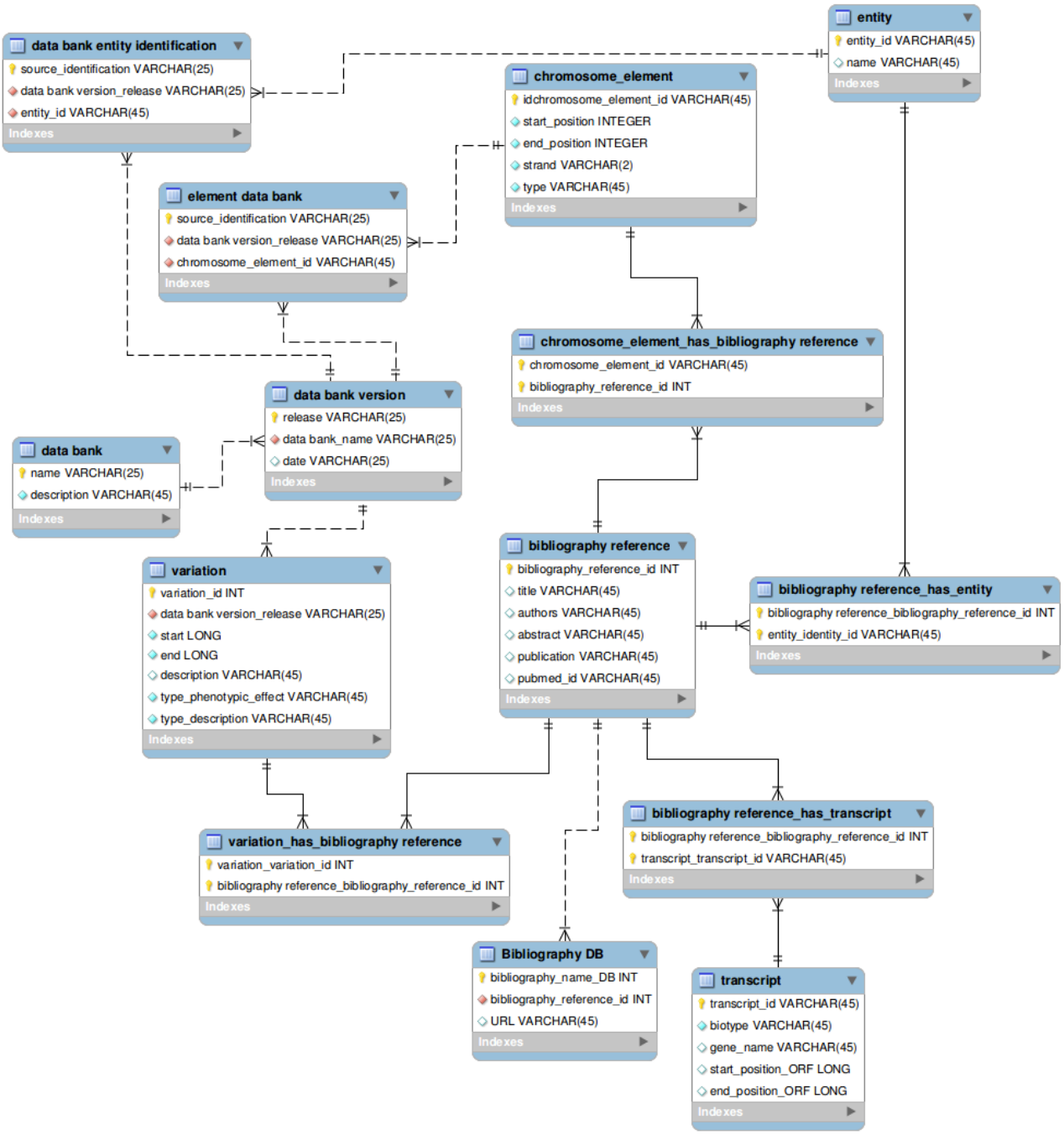


Fig. 19 Vista fuentes de datos y bibliografía en la base de datos

4.3 Definición de correspondencias entre los repositorios externos y el esquema de la base de datos

Para cargar la base de datos implementada (GDB), se han seleccionado las siguientes fuentes de datos: HapMap [21, 22], Cosmic [33], Ensembl [20], Jaspar [34], UCSC [35, 36] y Uniprot [23, 24]. La idea principal de la base de datos presentada es tener un repositorio que de manera integrada pueda contener toda la información hasta ahora existente del genoma. El proceso de selección de dichas bases de datos se realizó a través de reuniones con expertos biólogos del Centro de Investigación Príncipe Felipe especializados, cada uno de ellos, en diferentes ramas de la Genómica. Dado que dichos expertos son los usuarios finales de

dicha herramienta, conocer sus necesidades tras años de experiencia es muy importante a la hora de captar bien los requisitos e implementar una base de datos acorde a éstos que integre la información de las bases de datos que ellos utilizan. Cada uno de ellos nos comentó de donde extraía la información para realizar sus experimentos y lo útil que sería disponer de un único repositorio en el que poder hacer sus propias investigaciones de una manera más sencilla. Los motivos principales de la elección de dichas bases de datos, son su valor de estandarización en la comunidad de biólogos y su periódica actualización por cada uno de sus equipos técnicos, lo que permite tener las fuentes de datos con información más actual y relevante.

Siguiendo el formato utilizado hasta ahora, el estudio de las correspondencias entre de las fuentes de datos externas y el esquema de la base de datos presentado se presenta por vistas tal y como se ha hecho en los capítulos anteriores.

4.3.1 Vista estructural

Para cargar la vista estructural de la base de datos del genoma han sido seleccionadas cuatro bases de datos: Ensembl, Jaspar, USCS y HapMap.

La principal fuente de datos de la que se extrae información para esta vista es Ensembl. El proyecto Ensembl se inició en 1999 con el objetivo de anotar automáticamente el genoma, integrar esta anotación con datos biológicos disponibles y hacer todo esto público a través de la web. Más tarde, los datos disponibles fueron expandidos para incluir comparativas genómicas, variaciones y elementos reguladores.

Existen dos maneras de acceder a la información proporcionada por Ensembl, la primera consultando directamente su base de datos. Esta base de datos hay que descargarla desde el propio portal web de Ensembl e instalársela localmente. La segunda forma es mediante su herramienta BioMart que consiste en una interfaz de consulta y extracción de datos que ofrece los resultados de manera más sencilla.

La base de datos Ensembl es una gran base de datos, en cuanto a volumen de datos se refiere, que contiene información biológica muy variada. En ella podemos encontrar información desde la cadena de cada cromosoma, hasta las variaciones existentes en cada uno de ellos. El problema de esta base de datos radica principalmente en su mala estructura y la poca claridad que proporciona en los nombres y atributos de cada tabla dificultando su interpretación a pesar de ofrecer gran cantidad de información. Para solucionar estos problemas, Ensembl crea una interfaz gráfica con el sistema BioMart mucho más sencilla de manejar a la hora de buscar y obtener los datos, pero solo proporciona la información más relevante, lo que en muchos casos hace indispensable la instalación de la base de datos entera. Aprender a usar BioMart es una tarea sencilla, lo primero que se debe hacer una vez dentro es seleccionar la base de datos y el conjunto de datos de los cuales se quiere extraer la información y una vez seleccionadas ambas opciones, las consultas pueden comenzar. Estas consultas tienen filtros para fijar que es lo que se conoce y que atributos se quieren conocer. Cuando los filtros y los atributos son seleccionados aparecen los resultados.

A partir de la base de datos de Ensembl, sin usar la herramienta BioMart, se ha extraído información acerca de las secuencias de ADN de cada cromosoma, las citobandas de cada cromosoma y sus regiones conservadas (Tabla 1). La base de datos utilizada para la extracción de información es *homo_sapiens_core_63_37* para ser humano, gallus_*gallus_core_63_2* para el gallo… y así para cada especie, una vez dentro accedemos a la tabla *dna* y a la tabla *karyotype*. De la tabla *dna* se extrae la secuencia perteneciente a cada uno de los cromosomas y de la tabla *karyotype* se extrae las citobandas pertenecientes a cada cromosoma.

GDB database		Ensembl	
Table	Attribute	Query	Attribute
chromosome	name	dna	chrom
	species_id		Extraido a partir del nombre de la bd
	long		Cálculo a partir de la secuencia de ADN extraída
chromosome_sequence	sequence		Extraemos de toda la secuencia del cromosoma a partir del campo seq
cytoband	name	karyotype	Name
	chromosome_id		chrom
	score		%
	start_position		start_chrom
	end_position		end_chrom

Tabla 1 Correspondencia entre Ensembl y la vista estructural (1)

Por otra parte, los datos que extraemos desde la interfaz BioMart son los genes y los tfbs (transcription factor binding sites).

Para los genes se utiliza la base de datos *Ensembl Gene 62* y uno a uno se va seleccionando el conjunto de datos para cada una de las especies: *Homo Sapiens, Mus Musculus*... El formato es el mismo para cada especie, lo que cambia es la información que se obtiene (Tabla 2).

GDB database		Ensembl	
Table	Attribute	Query	Attribute
chromosome element	name_chromosome	Query dataset	Chromosome name
	start_position		Gene Start (bp)
	end_position		Gene End (bp)
	strand		Strand
	type		“gene”
gene	name		Associated Gene Name
	description		Description
	biotype		Gene Biotype
	status		Status (gene)
	gc_percentage		% GC content

Tabla 2 Correspondencia entre Ensembl (usando BioMart) y la vista estructural (2)

Para los tfbs se utiliza la base de datos *Ensembl Regulation 62* y uno a uno se va seleccionando el conjunto de datos para cada una de las especies. Una vez seleccionada la especie accedemos a la sección de *Annotated Features*. En este caso el formato también es el mismo para cada una de las especies, lo único que cambia son los datos que se obtienen (Tabla 3).

GDB database		Ensembl	
Table	Attribute	Query	Attribute
chromosome element	name_chromosome	Query dataset	Chromosome Name
	start_position		Start (bp)
	end_position		End (bp)
	strand		Por defecto se asume siempre la hebra positiva
	type		"tfbs"
tfbs	name		Feature Type
	type		Feature Type Class
	description		Featute Type Description
	cons_seq		Secuencia real a la que se une extraída a partir de las posiciones start y end

Tabla 3 Correspondencia entre Ensembl (usando BioMart) y la vista estructural (3)

Por otra parte, los factores de transcripción, que se unen a estos tfbs, se obtienen de la fuente de datos Jaspar. Jaspar es una fuente de datos revisada por expertos que obtiene su información de experimentos publicados sobre factores de transcripción para eucariotas. La calidad de esta fuente de datos radica en que es un repositorio de libre acceso y que al estar revisado por expertos garantiza una mayor calidad de información. Los factores de transcripción son un tipo de genes, por lo que se encuentran ya almacenados en la base de datos extraídos desde la interfaz BioMart de Ensembl, pero Ensembl no contiene información relacionada con la secuencia consenso de la región, por lo que se decide extraerla de Jaspar. La información en Jaspar está modelada en forma de matrices y dividida entre diferentes directorios, para el ámbito de nuestro trabajo el directorio utilizado para extraer la información es *Jaspar Core*. Si accedemos a él y seleccionamos el tipo de datos no redundantes, accedemos a una carpeta en la que se encuentran una serie de directorios divididos por especies. Dentro de cada uno de estos directorios divididos por especies existe un fichero tabulado llamado *matrix_only.txt* que representa una matriz con la siguiente información: el nombre del factor de transcripción y la matriz asociada a él. La matriz muestra en cada posición del factor de transcripción, el número de ocurrencias en ciertas muestras de que exista un nucleótido u otro. De esta información se extraen los nucleótidos que más frecuencia de aparición, y la unión de todos ellos forman lo que se llama la secuencia consenso (Tabla 4).

GDB database		Jaspar	
Table	Attribute	File	Field
tf	cons_seq	matrix_only.txt	lectura de la matrix escogiendo los valores con porcentajes más altos para A, T, C y G.

Tabla 4 Correspondencia entre Jaspar y la vista estructural

Dado que el nombre proporcionado por Jaspar para cada gen coincide con el nombre proporcionado por Ensembl, ya que es un nombre estándar, la manera de mapear los identificadores entre las dos bases de datos resulta sencilla.

Siguiendo con el resto de información, las *cpg islands* se obtienen desde el portal web de la Universidad de California, Santa Cruz (UCSC). UCSC es una de las universidades más importantes de Estados Unidos y entre sus proyectos de investigación se encuentra el de la secuenciación del genoma. Dicho proyecto es muy importante en lo que respecta al mundo de la biología y contiene las secuencias de referencia así como versiones de estudio de una gran cantidad de genomas. La UCSC proporciona una interfaz gráfica en la que se permite buscar información de una manera más visual. Dentro de dicha interfaz, al seleccionar la opción grupo: *regulation*, término: *cpg_island*, especie y la versión del genoma para la que quieres hacer la consulta, UCSC proporciona un fichero de tipo texto tabulado con la información requerida. El fichero consta de los siguientes campos: el cromosoma al que pertenece (*chrom*), la posición dentro del cromosoma donde empieza (*chromStart*), la posición en el cromosoma en la que termina (*chromEnd*) y el porcentaje de Cs y Gs que tiene la región (*perCpg*). El resto de datos que proporciona el fichero son otros cálculos que se quedan fuera del ámbito de estudio (Tabla 5).

GDB database		UCSC	
Table	Attribute	File	Field
chromosome element	name_chromosome	cpgTables.txt	chrom
	start_position		chromStart
	end_position		chromEnd
	strand		Por defecto se asume siempre la hebra positiva
	type		“cpg_island”
cpg_island	cpg_percentage		perCpg

Tabla 5 Correspondencia entre UCSC y la vista estructural

Finalmente, la información referente a los hotspot se obtiene de la base de datos de HapMap. HapMap es un proyecto cuyo principal objetivo es identificar y catalogar las similitudes y diferencias genéticas entre individuos humanos, ayudando a los investigadores a asociar las variaciones a su efecto fenotípico. Este proyecto es de dominio público y se trata de una colaboración entre científicos de muchos países. Como se ha comentado su principal objetivo son las variaciones dentro del genoma, más concretamente los SNPs y sus frecuencia de aparición en cada población, pero existe cierta información acerca de los puntos de recombinación o hotspots dentro del cromosoma que también es de su interés. Dichos hotspots son los principales responsables de la diversidad humana que permiten a las especies adaptarse y evolucionar en un entorno cambiante y que, tras una serie de investigaciones científicas, se ha demostrado que en parte depende de los SNPs. La información que nos interesa en este caso se encuentra dentro del directorio *recombination* y más concretamente el fichero correspondiente a la última versión de datos que fecha del año 2011 llamado *2011-01_phaseII_B37/*. Una vez dentro del directorio se encuentra el archivo comprimido: *genetic_map_HapMapII_GRCh37.tar.gz*, que al descomprimirlo contiene información acerca de las cpg_island existentes separadas por el cromosoma al que pertenecen, representado por la letra X, debido a la cantidad de información disponible siguiendo el formato *genetic_map_GRCH37_X.txt*. Los

campos que contiene este fichero son: el cromosoma al que pertenece la recombinación (*Chromosome*) y la posición en la que sucede (*Position (bp)*) (Tabla 6).

GDB database		HapMap	
Table	Attribute	File	Field
hotspot	name_chromosome	genetic_map_GRCH37_X.txt	Chromosome
	position		Position (bp)

Tabla 6 Correspondencia entre HapMap y la vista estructural

Sobre la tabla *species*, cabe destacar que actualmente no existe ninguna base de datos formal que recopile toda la información acerca de las especies existentes hoy en día, por lo que debido a que la cantidad de especies es muy reducida se llega al acuerdo de insertarlas manualmente. Por otra parte los factores de regulación del splicing tampoco se obtienen de ninguna base de datos sino que se extraen en el laboratorio por los expertos biólogos.

4.3.2 Vista de transcripción

Para cargar la vista traducción se hace uso de dos bases de datos: Ensembl y Uniprot.

Ensembl contiene datos actualizados de todos los transcritos que tiene cada gen, así como los exones que forman cada uno de ellos y sus regiones codificantes. Para extraer estos datos, como en la vista estructural, para algunos elementos, se utiliza la herramienta BioMart que facilita la búsqueda de información gracias a su amigable e intuitiva interfaz y devuelve los resultados en un fichero tabulado o los muestra en la pantalla en formado de tabla. Para la extracción de estos datos se selecciona la base de datos *Ensembl Gene 62* y el conjunto de datos de la especie que se quiera consultar en cada momento, por ejemplo: *Homo sapiens genes, Gallus gallus genes, Mus musculus gene*, etc (Tabla 7).

Respecto a las proteínas, se utiliza la base de datos de Uniprot. Uniprot es el repositorio central de datos de tipo proteínas creado por la combinación de Swiss-Prot, TrEMBL y PIR. Su principal función es proporcionar a la comunidad científica un recurso completo y de alta calidad de secuencias de proteínas con su respectiva información funcional. Cabe destacar también que es de acceso libre a través de Internet y que su información está extraída de diferentes publicaciones científicas. El consorcio UniProt compende el European Bioinformatics Institute (EBI), el Swiss Institute of Bioinformatics (SIB), y el Protein Information Resource (PIR).

Dentro de UniProt existen cuatro bases de datos diferentes: UniProtKB, UniRef, UniParc y UniMES. Para suplir las necesidades de este trabajo se usará la base de datos UniProtKB consistente en dos secciones: UniProtKB/Swiss-Prot (base de datos constantemente revisada en la que se incluyen datos de manera manual) y UniProtKB/TrEMBL (base de datos sin revisar en la que se incluyen los datos de manera automática). Ambas secciones son útiles por lo que los datos de las dos bases de datos van a almacenarse. Dentro de su página web, Uniprot proporciona un ftp donde se encuentra la información disponible, si seleccionamos la versión de datos actual, del 31 de Mayo del 2011, y escogemos el directorio *knowlegebase* accedemos a los ficheros que nos interesan: *uniprot_sprot.fasta.gz* para los datos de Swiss-Prot y *uniprot_trembl.fasta.gz* para los datos de TrEMBL (Tabla 8).

GDB database		Ensembl	
Table	Attribute	Query	Attribute
chromosome element	name_chromosome	Query database	Chromosome Name
	start_position		Exon Chr Start (bp)
	end_position		Exon Chr End (bp)
	strand		Strand
	type		“exon”
transcript	biotype		Transcript Biotype
	gene_name		Associated Gene Name
	start_position_ORF		5’UTR End y 3’UTR Start dependiendo del strand
	end_position_ORF		3’UTR Start y 5’UTR End dependientdo del strand

Tabla 7 Correspondencia entre Ensembl y la vista de transcripción

GDB database		Uniprot	
Table	Attribute	Query	
protein	transcript_id	uniprot_sprot.fasta.gz y uniprot_trembl.fasta.gz	Automáticamente con la herramienta de mapeo ID Mapping de Uniprot a partir del nombre de la proteína.
	name		name
	sequence		sequence
	accesion		id

Tabla 8 Correspondencia entre Uniprot y la vista de transcripción

El mapeo de identificadores entre las fuentes de datos Ensembl y Uniprot se realiza mediante ficheros disponibles en UniProt que proporcionan los identificadores que proporcionan algunas de las bases de datos más relevantes para cada proteína, entre ellas Ensembl.

4.3.3 Vista de variaciones

Para cargar la vista de variaciones de la base de datos han sido seleccionadas tres fuentes de datos: HapMap, Ensembl y Cosmic.

HapMap como se ha comentado anteriormente, proporciona información acerca de las variaciones ocurridas en el genoma humano, entre ellos los SNPs. La investigación de HapMap acerca de los SNPs ha sido llevada a cabo en tres fases. La última fase (Fase III) apareció en 2009 con amplios estudios del genoma que examina completamente menos alelos comunes en poblaciones con un amplio rango de ancestros. A pesar de que el número de SNPs genotipados es reducido a 1.6 millones aproximadamente, la cantidad de individuos de referencia y las poblaciones aumentan a 1184 y 11 respectivamente por lo que se gana conocimiento. Los datos del proyecto están disponibles para uso público en el sitio web de HapMap. Esta web ofrece altas cantidades de descarga de datos organizados en archivos de texto tabulados.

El conjunto de datos de interés para esta vista es la información contenida en los directorios: *frequencies*, *cnv_data* y *ld_data*. En primer lugar, *frequencies* es un directorio que contiene los mencionados SNPs con sus frecuencias de aparición en cada población. Los datos usados son los de la última versión, Agosto 2010, y en ella se incluye información sobre las dos fases anteriores. Debido a la gran cantidad de datos disponibles, los SNPs se dividen en diversos archivos dentro del directorio divididos a su vez en dos tipos: *allele_freqs* y *genotype_freqs*. *Allele_freqs* hace referencia a la frecuencia de las variaciones tipo SNP teniendo en cuenta solo un alelo del cromosoma y por otro lado, *genotype_freqs* hace referencia a la frecuencia de las variaciones de tipo SNP teniendo en cuenta los dos alelos. Además, todos estos archivos están divididos por el cromosoma en el cual se encuentran y la población a la cual afectan, referidos por los sufijos X e Y respectivamente.

Los campos de los archivos de frecuencia alélica que se usan en este trabajo son: SNP id (*rs#*), cromosoma afectado (*chrom*), posición en la que se encuentra (pos), alelo de referencia (*refallele*), frecuencia de aparición del alelo de referencia (*refallele_freq*), alelo que varia (*otherallele*) y frecuencia de ocurrencia del alelo del cambio (*otherallele_freq other*). El centro de secuenciación, las técnicas usadas para hacerlo y el numero de muestras para cada SNP han sido omitidas porque se encuentran fuera del ámbito de nuestro estudio (Tabla 9).

GBD database		HapMap	
Table	Attribute	File	Field
variation	source_id	allele_freqs_X_Y	rs#
	start		pos
snp_allele	allele		Refallele or otherallele
snp_allele_pop	frequency		Refallele_freq or otherallele_freq other

Tabla 9 Correspondencia entre HapMap y la vista de variaciones (1)

Por otro lado, los campos de los archivos de frecuencia genotípica que son interesantes para este studio son: SNP id (*rs#*), cromosoma afectado (*chrom*), posición del SNP (*pos*), alelo de referencia en homocigosis (*refhom-gt*), frecuencia de aparición del alelo de referencia en homocigosis (*refhom-freq*), alelos en heterocigosis (*het-gt*), frecuencia de aparición de los alelos en heterocigosis (*het-freq*), alelo raro en homocigosis (*otherhom-gt*) y frecuencia de aparición del alelo raro en homocigosis (*otherhom-freq*). Como en el otro archivo, el centro de secuenciación y las técnicas usadas para secuenciar se escapan del ámbito de estudio por lo que se decide no tenerlas en cuenta (Tabla 10).

GBD database		HapMap	
Table	Attribute	File	Field
variation	source_id	genotype_freqs_X_Y	rs#
	start		pos
genotype_allele	allele1		refhom-gt, het-gt o otherhom-gt
	allele2		refhom-gt, het-gt o otherhom-gt
genotype_allele_pop	frequency		refhom-freq, het-freq o otherhom-freq

Tabla 10 Correspondencia entre HapMap y la vista de variaciones (2)

En segundo lugar, el directorio *ld_data* hace referencia a los archivos de datos con información acerca del "linkage disequilibrium". En este caso la cantidad de información es elevada también, por lo que se divide, como en los otros archivos, por cromosoma y población referidos por los sufijos X e Y respectivamente. Los campos de cada uno de estos ficheros son: posición en el cromosoma del primer marcador (*chromosomal position of marker 1*), posición en el cromosoma del segundo marcador (*chromosomal position marker 2*), identitificador del primer marcador (*rs# from marker 1*), identificador del segundo marcador (*rs# from marker 2*) y tres parámetros estadísticos muy biológicos en los que no vamos a entrar en detalle que se utilizan para su medida (*Dprime, R square y LOD*) (Tabla 11).

GDB database		HapMap	
Table	Attribute	File	Field
variation	source_id	Ld_X_Y	rs# from marker 1 o rs# from marker 2
	start		chromosomal position marker 1 o chromosomal position marker 2
LD	dprime		Dprime
	rsquare		R square
	lod		LOD

Tabla 11 Correspondencia entre HapMap y la vista de variaciones (3)

En tercer lugar, *cnv_data* hace referencia a las variaciones en el número de copias de una secuencia de nucleótidos en un individuo dentro de su ADN. Este tipo de variaciones se codifica con los enteros [0,1,2,3,4]. El valor 2 hace referencia al valor esperado de copias de un individuo, el valor 1 indica un borrado en heterocigosis, el valor 0 un borrado en homocigosis y valores superiores a 2 indican el numero de duplicaciones. La cantidad de información sobre el tipo de polimorfismo es menor la relacionada con la frecuencia y la relacionada con el linkage disequilibrium, por lo que todos los datos son almacenados en un único fichero llamado: hm_cnv_submision.txt. Los campos de este fichero son: CNP_id (*cnp_id*), cromosoma afectado (*chr*), posición de inicio de la secuencia (*start*), posición final de la secuencia (*end*) y una muestra de los individuos a los que se les ha sometido a la prueba identificados por un string codificado (Tabla 12).

GDB database		HapMap	
Table	Attribute	File	Field
variation	source_id	hm_cnv_submision	cnp_id
	start		start
	end		end
cnp	max_rep		max of all samples
	min_rep		min of all samples

Tabla 12 Correspondencia entre HapMap y la vista de variaciones (4)

Finalmente, la información sobre las poblaciones es extraída del fichero README de los datos LD (*00README*) y del fichero README de la última versión de HapMap (*00README.releasenotes_rel28*). El archivo readme de LD proporciona información sobre el nombre y la descripción de la población y el fichero readme de la última versión de HapMap incluye información acerca del tamaño de cada una de ellas (Tabla 13).

GDB database		HapMap	
Table	Attribute	File	Field
population	name	00README	name
	descriptión		description
	Size	00README.releasenotes_rel28	size

Tabla 13 Correspondencia entre HapMap y la vista de variaciones (5)

Ensembl es una base de datos que contiene información actualizada de todos los SNPs que tiene cada población, así como la frecuencia de aparición de cada uno de ellos. Uno de los principales motivos por lo que se selecciona esta fuente es gracias a que incluye datos de otras fuentes, como por ejemplo dbSNP y HGMD, con lo que extrayendo únicamente la información desde este repositorio se obtendrían más cantidad de SNPs. Para extraer estos datos, igual que en las vistas estructural y de traducción, se utiliza la herramienta BioMart que facilita la búsqueda de información gracias a su amigable e intuitiva interfaz y devuelve los resultados en un fichero tabulado o los muestra en la pantalla en formado de tabla.

Para esta vista se selecciona la base de datos *Ensembl Variation 62* y el conjunto de datos de la especie que se quiera consultar en cada momento, por ejemplo: *Homo sapiens genes, Gallus gallus genes, Mus musculus gene*, etc (Tabla 14).

GBD database		Ensembl	
Table	Attribute	Query	Attribute
variation	source_id	query dataset	variation_id
	start		Position on Chromosome
snp	map_weight		Map-weight
	allele_string		Allele
	ancestral_allele		Ancestral Allele

Tabla 14 Correspondencia entre Ensembl y la vista de variaciones

Cosmic, Catalogue of Somatic Mutation in Cancer, es una fuente de datos publica que contiene miles de mutaciones somáticas que están implicadas en el desarrollo de cáncer. Desde 2004, esta base de datos contiene detalles de 1.5 millones de experimentos llevados a cabo con 13426 genes en al menos 370000 tumores y describiendo alrededor de 90000 mutaciones de individuos. Todos estos datos de mutaciones y su información asociada se extraen principalmente de la literatura y se introducen en la base de datos COSMIC. Para identificar artículos a la hora de insertar nuevas mutaciones, el recurso utilizado es PubMed. En PubMed empieza una búsqueda exhaustiva de artículos que contienen información relevante sobre datos de mutaciones y se extraen a través de patrones como por ejemplo: “Gene and human and mutation”. Una vez obtenidos los artículos estos se examinan detenidamente y la muestra y los datos de la

mutación son extraídas. Para facilitar la descarga de sus datos, COSMIC proporciona un directorio ftp en el que se encuentran todos los archivos que contienen su información en formato tabulado. Data_export es un directorio el cual contiene el archivo llamado *CosmicMutantExport_v51_270111.tsv* que proporciona todas las mutaciones asociadas a todos los genes estudiados en COSMIC.

GBD database		COSMIC	
Table	Attribute	File	Field
variation	source_id	CosmicMutantExport_v51_270111	mutation ID
	description		Mutation Description
	start		Posición calculada a partir del campo NCBI36 genome position
mutation	cds_mutation		Mutation CDS
	aa_mutation		Mutation AA
Inversión	bases		Obtenidas a partir del campo Mutation CDS
deletion	bases		Obtenidas a partir del campo Mutation CDS
insertion	repetition		Obtenidas a partir del campo Mutation CDS
	sequence		Obtenidas a partir del campo Mutation CDS
indel	ins_repetition		Obtenidas a partir del campo Mutation CDS
	del_bases		Obtenidas a partir del campo Mutation CDS
	Ins_sequence		Obtenidas a partir del campo Mutation CDS

Tabla 15 Correspondencia entre Cosmic y la vista de variaciones

Los campos de interés de este archivo para este trabajo son: el identificador de la variación (*mutation ID*), su descripción (*Mutation Description*), nomenclatura estándar en la cual se nombran las mutaciones con formato HGVS con respecto a la posición del CDS (*Mutation CDS*), nomenclatura standard en la cual se nombran las mutaciones con formato HGVS con respecto a la posición proteínica (*Mutation AA*) y la posición en el genoma en la que aparece la variación (*NCBI36 genome position*). El resto de campos del archivo se quedan fuera de nuestro trabajo (Tabla 15).

4.3.4 Vista de rutas metabólicas

BioPax [14] es un formato de intercambio de datos aprobado por la comunidad científica para describir pathways. La mayoría de bases de datos de impacto que definen las asociaciones entre procesos dentro del genoma humano, así como Reactome [37], BioCyc [38], BIND [39], etc, han decidido proporcionar la información en este formato, por lo que como se describirá en capitulo 5 la carga de datos de rutas metabólicas no se realiza en base a este modelo conceptual que se ha descrito en el apartado 4.1, sino que se propone efectuarla en base al formato estándar de BioPax. Para ello se propone el diseño de un modelo conceptual extraído a partir de la descripción que proporciona BioPax en su sitio web y que modele dicho

vocabulario, de tal manera que a partir de dicho modelo se pueda extraer automáticamente la base de datos que permita contener toda esta información.

4.3.5 Vista de fuentes de datos y bibliografía

Con respecto a las fuentes de datos utilizadas que se almacenarán en la base de datos cabe destacar que son todas las nombradas anteriormente: Ensembl, HapMap, UCSC, Jaspar, Cosmic y Uniprot, y que para cada una de ellas se guardará el identificador que proporciona dicha base de datos para el elemento que de ella se ha extraído.

Con respecto a la bibliografía cabe destacar que no todas las bases de datos proporcionan artículos literarios que demuestren que esa información está revisada por un experto. Los elementos que la proporcionan normalmente devuelven el identificador de PubMed del artículo, por lo que el trabajo de inserción de autor, título, año de la publicación... es un trabajo de búsqueda, extracción e inserción una vez se obtiene la información. Por otra parte, para los elementos que no proporcionan ninguna referencia bibliográfica, se intenta realizar una criba, también en PubMed, seleccionando como palabras clave ciertos aspectos relevantes del concepto. En caso de encontrar un artículo que confirme la existencia demostrada del elemento en cuestión, dicho artículo se insertará en la base de datos. Este proceso de selección del artículo no es un trabajo trivial, cualquier persona no podría realizarlo ni mucho menos una máquina de manera automática con el 100% de aciertos, debido a que muchas veces el contenido del artículo aunque contenga esas palabras clave no coincide con lo que se está buscando, por este motivo, se convierte en una necesidad la ayuda de un experto en el campo que ayude a la elección.

4.4 Implementación de la base de datos

En este apartado describe el procedimiento que se ha seguido a la hora de cargar la información en la base de datos. Dadas las buenas propiedades de los sistemas ETL (extracción, transformación y carga) se decide utilizar esta aproximación para el ámbito de esta tesis con el objetivo de extraer información desde múltiples fuentes de datos, darle formato y limpiarla y finalmente insertarla en nuestra base de datos.

El primero de los niveles del proceso es la extracción que consiste en obtener los datos desde los diferentes repositorios. Como ya se ha comentado anteriormente, en bioinformática es muy frecuente que cada sistema por separado utilice una organización diferente de los datos y un formato distinto. De entre los formatos más usuales se encuentran las bases de datos relacionales y los ficheros planos tabulados, pero puede incluso darse el caso de encontrar alguna base de datos no relacional e incluso que utilicen otras estructuras diferentes.

Una parte intrínseca dentro del proceso de extracción de los datos, es la de analizar los datos extraídos, es decir, realizar un chequeo que verifique si los datos han sido correctamente extraídos de las fuentes y en caso contrario los datos serán rechazados para volver a realizar la operación desde el principio.

Un requisito importante que se debe exigir a la tarea de extracción es que ésta cause un impacto mínimo en los sistemas de los cuales se quiere obtener la información, es decir, si son muchos los datos que se van a extraer, situación muy común en este dominio, el repositorio origen podría ralentizar su funcionamiento o incluso colapsarse. Esta situación puede llegar al punto en el que mientras se realice la descarga el

sistema de datos no pueda utilizarse con normalidad para su uso cotidiano por el resto de usuarios, pudiendo incluso llegar a ocasionar el desagrado de los propios dueños del sistema de datos. Para solucionar este tipo de problemas, se propone que las operaciones de extracción de un gran humero de datos se programen en horas o días donde este impacto sea mínimo y que la extracción de información de gran volumen se divida en secciones más pequeñas.

En segundo lugar, en la fase de transformación se aplican una serie de manipulaciones sobre la información extraída para convertirla en datos que enlacen con la estructura deseada a la hora de ser cargados. Se deben definir una serie de estándares a la hora de cargar la información como por ejemplo: traducir posiciones para que todo elemento haga referencia a una misma secuencia de referencia p.e. la posición de cada elemento va referida a una secuencia distinta dependiendo del momento en el que se encontró o con respecto a que se ha definido, codificar valores p.e. a la hora de indicar si la variación es cromosómica (C) o génica (G), obtener nuevos valores calculados p.e. a la hora de extraer la secuencia consenso de un factor de transcripción ... y como este tipo de ejemplos muchos otros. De entre las situaciones comentadas arriba, cabe destacar la transformación de posiciones entre secuencias de referencia distintas, siendo esta una de las situaciones más comunes en el ámbito de la bioinformática y más importante en esta fase del proceso de carga. Pese a ello y aunque las desavenencias seguirán existiendo siempre, debido a la cantidad de datos que hay que tener en cuenta para realizar un experimento, para intentar lidiar de manera menos complicada con este problema, las bases de datos más actuales están empezando a cambiar sus posiciones para referirse todas a una misma referencia, la del cromosoma, la que hemos usado nosotros también en nuestro trabajo.

Por último, en la tercera fase del proceso, la fase de carga, es cuando los datos son almacenados en la base de datos, en nuestro caso GDB. Esta fase es la que interactúa directamente con la base de datos y en esta fase es donde se tendrán en cuenta las restricciones en lenguaje natural que se han definido a la hora de implementar la base de datos. Por otro lado, al realizar la operación de inserción, el resto de restricciones definidas sobre el esquema así como los triggers también se aplicarán, lo que contribuirá a que se garantice la calidad de los datos almacenados.

La figura 20 trata de resumir de modo visual lo explicado en este apartado. Se puede decir que en la primera capa del proceso de carga (1), toda la información necesaria de las fuentes de datos que se ha detallado en el capitulo anterior es extraída desde la web. Esta información está desestructurada y heterogénea por lo que estos datos “crudos” son enviados a una segunda capa (2) donde se les realizan diversas transformaciones con el objetivo de dar formato a los datos de acuerdo con la representación de la información en el esquema de base de datos. De esta manera, dichas transformaciones son enviadas a una tercera capa (3) que comunica directamente con la base de datos y se insertan sin necesidad de más operaciones.

Cada uno de los módulos de este proceso ETL es completamente independiente de los demás, lo que facilita el diseño del sistema y mejora su flexibilidad y escalabilidad. El desarrollo por capas seguido permite la incorporación de nuevas fuentes de datos de manera sencilla y hace que el módulo sea flexible a cambios tanto en las fuentes con en la propia base de datos.

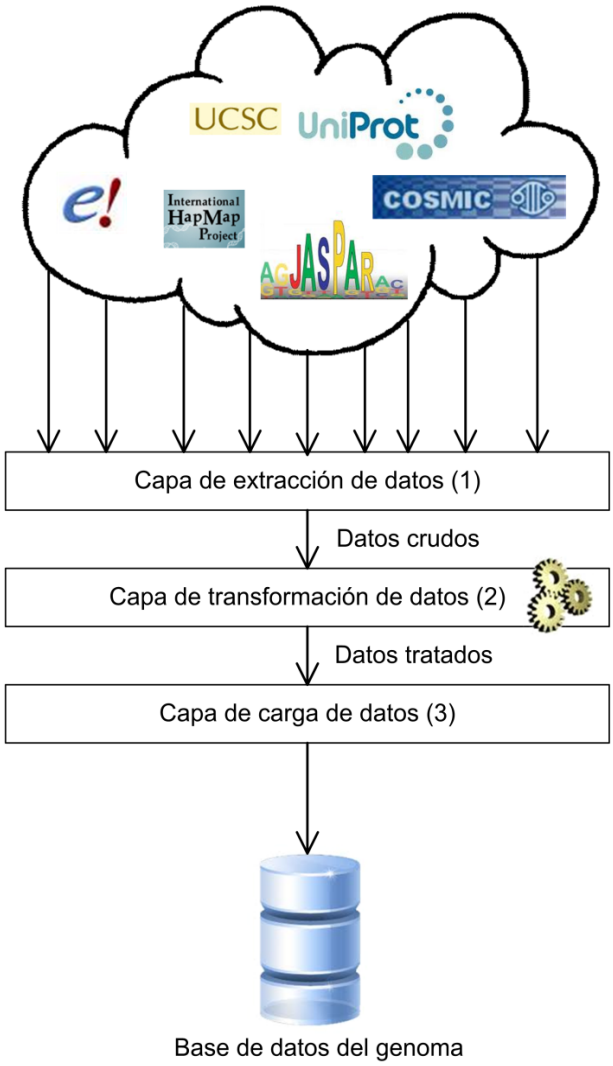


Fig. 20 Estructura ETL del modulo de carga

Con respecto a la implementación cabe destacar que a pesar de que muchos entornos de desarrollo permiten extraer automáticamente código a partir de un modelo conceptual, en este proyecto no se ha utilizado dicha metodología. Esto es debido a que tanto la base de datos como el módulo de carga se diseñaron de manera paralela al modelo conceptual, es decir, el modelo conceptual iba definiéndose por vistas, y por cada vista se iba implementando manualmente la base de datos completando la versión anterior, del mismo modo el código para la carga de datos se fue implementando paralelamente con el objetivo de que algunos biólogos pudiesen ir efectuando sus experimentos sin necesidad de alargar la espera de todos. Esto se ha debido a que el trabajo desarrollado ha sido una colaboración larga y extensa de más de un año de trabajo por lo que si no se trabajaba en paralelo, los resultados de algunas vistas se hubiesen obtenido con mucho mas retardo.

Finalmente, el lenguaje de programación utilizado ha sido Java debido a las ventajas que proporciona a la hora de programar ya que permite un alto nivel de funcionalidad y además está diseñado para funcionar en cualquier sistema operativo y la plataforma de desarrollo elegida ha sido Eclipse.

5. INTEGRACIÓN DE LA PROPUESTA BIOPAX

Como se ha comentado en la introducción de este trabajo, existen una gran cantidad de repositorios de información accesibles en Internet cada uno de ellos con diferente formato e información específica que resuelven problemas particulares. Para superar esta situación, una serie de expertos reconocidos en rutas metabólicas definen un vocabulario estándar para representar su información de manera que puedan intercambiar información en un mismo formato. Algunas de estas propuestas son comentadas y descritas en el capítulo 2 del estado del arte, por ejemplo SBML o PSI-MI, pero ninguna de ellas ha sido capaz de representar todos los principales tipos de rutas metabólicas existentes. Para lograr este reto, nace BioPax que cumple con los requisitos necesitados siendo capaz de representar todos los conceptos necesarios para definir dichas rutas, pero ¿qué sucede a la hora de intentar utilizar esta información? Ciertos problemas aparecen, por lo que se propone el uso de modelos conceptuales para solventarlos. Una vez representado el modelo conceptual, se implementa su base de datos asociada, así como un módulo de carga para almacenar toda la información en formato BioPax. Por otra parte, dicho modelo conceptual, debido a la relevancia que BioPax tiene en la comunidad científica, será integrado en nuestro modelo conceptual con el objetivo de cubrir bien las necesidades y requisitos de los expertos.

5.1 ¿Qué es BioPax?

BioPax [14] es un formato de intercambio de datos que trata de lograr la integración, intercambio y visualización de una amplia cantidad de rutas metabólicas (Fig. 21). El formato BioPax está actualmente definido en OWL e implementado en RDM/XML y está siendo usado por muchos de los principales repositorios de rutas metabólicas como por ejemplo Reactome o BioCyc para representar su información. De esta manera BioPax permite a los investigadores compartir e intercambiar información de rutas metabólicas como por ejemplo interacciones proteína-proteína o redes de regulación de genes bajo un mismo formato facilitando el trabajo. Esta representación proporciona un gran avance para la comunidad bioinformática y dado que nuestro trabajo depende del conocimiento de los expertos del dominio y de lo que ellos consideran importante es una necesidad para nosotros estudiarlo.

5.2 Desventajas del uso de BioPax

Con el nuevo formato BioPax, los biólogos tienen toda la información de interés sobre rutas metabólicas en el mismo formato, de esta manera, muchos problemas desaparecen, ahora los conceptos están definidos siguiendo la misma representación, pero, ¿qué sucede cuando un biólogo va a trabajar con un fichero OWL implementado en RDF/XML?

Cabe destacar que este estudio se realiza desde el punto de vista informático, no vamos a entrar en discusión de si la descripción de los términos es correcta o incorrecta, sino que vamos a centrarnos en si la representación a nivel de formato de los datos es la más adecuada y de qué manera podría mejorarse.

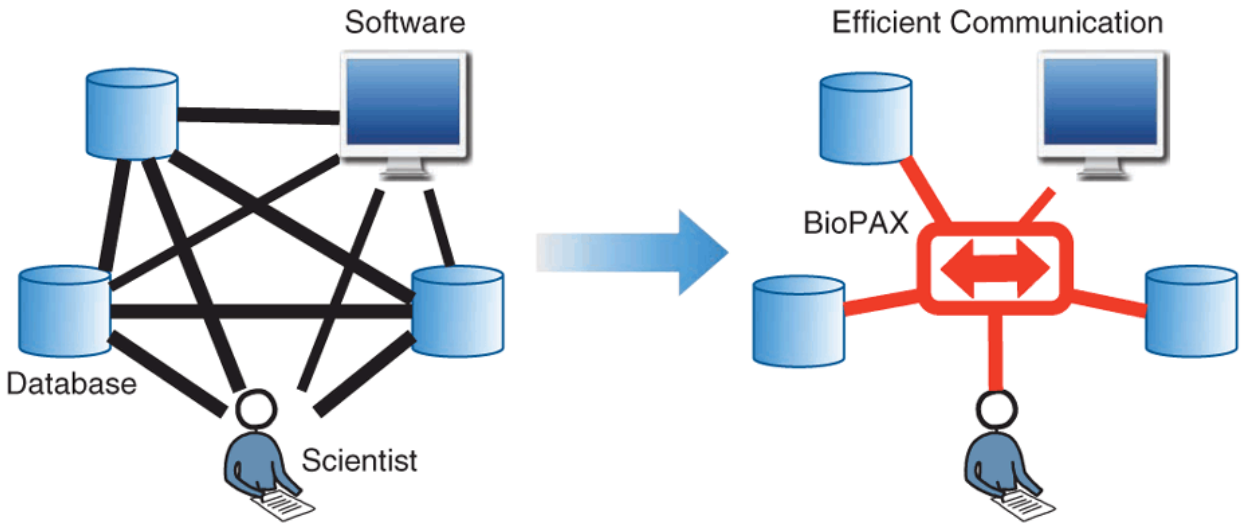


Fig. 21 BioPax es un lenguaje estándar de las rutas metabólicas compartido por la comunidad científica

Aunque BioPax sea un gran avance para la comunidad científica que ha conseguido solucionar el problema de la integración de tipos de datos biológicos con respecto a rutas metabólicas, su formato de presentación tiene ciertas desventajas:

- Los datos OWL están implementados en RDF/XML, lo que implica un problema a la hora de extraer datos y de analizar el fichero ya que es un formato muy poco intuitivo y difícil de interpretar.
- El experto debe realizar sus experimentos con todos los datos descargados en su propia máquina de trabajo.
- En caso de que el tamaño de los datos sea muy grande y se encuentren almacenados en un único fichero, el uso de la CPU aumentará y los requisitos de la memoria pueden llegar a ser altos.
- La integración de datos entre diferentes fuentes no es fácil y mucho menos intentar realizar consultas que relacionen los datos entre ellas.
- A partir de este tipo de ficheros, la generación de código de manera automática no es trivial.

Todas estas desventajas pueden llegar a ser más evidentes cuando nuevos datos sean obtenidos y estos ficheros se hagan más grandes, cosa que no es de extrañar debido a la gran cantidad de información que se genera a diario en este dominio.

Por estos motivos, para resolver las desventajas citadas, proponemos el desarrollo de un sistema de información basado en la descripción de tipología de datos propuesta en biopax, que integre la información de las rutas metabólicas mediante el uso de modelos conceptuales que implementen una base de datos relacional.

5.3 Modelado conceptual de la propuesta BioPax

Debido a la complejidad de comprensión del fichero OWL de BioPax, el equipo proporciona dentro de su página web, un documento en el que se describen cada una de las entidades que forman las rutas metabólicas y las relaciones existentes entre ellas. Para transformar la definición de términos en OWL de BioPax al lenguaje de modelado conceptual UML se han desarrollado una serie de correspondencias entre los dos lenguajes usando la especificación y descripción proporcionada por dicho documento BioPax. Dicha correspondencia se inicia con el estudio del documento a fondo en el que se definen los diversos formatos con los que se describen los conceptos, una vez realizado dicho estudio, se abstraen los elementos más importantes del lenguaje de modelado UML para tratar de asociarlos con su correspondiente representación en la descripción del documento BioPax y una vez realizada la correspondencia se define el modelo conceptual correspondiente.

5.3.1 Correspondencias entre el lenguaje UML y el lenguaje de BioPax

Para comprender la organización del documento BioPax se va a describir en primer lugar su estructura. A grandes rasgos, podemos decir que BioPax está formado por una serie de términos con sus respectivas características y relaciones. BioPax proporciona en su fichero de documentación la descripción de cada uno de sus elementos de dos maneras diferentes, textual y mediante diagramas de propiedades de objetos. Cada uno de los formatos se complementan el uno al otro, por lo que cada uno de los términos se encuentran descritos en ambas formas, no obstante cada formato proporciona información diferente al otro en algunos aspectos.

Para entender mejor este tipo de formatos se van a describir cada uno de ellos de manera que se visualice de una forma más sencilla lo que se está explicando. Además, para facilitar la comprensión al lector se van a abstraer un conjunto de elementos existentes en BioPax con el objetivo de mostrar cada una de sus descripciones y definir cuál es la información que se extrae de cada uno de ellos. De esta manera si luego extrapolamos el problema a la descripción entera de BioPax obtendremos su completo conocimiento. Los elementos seleccionados para realizar el ejemplo son *Entity* y *Gene* debido a que, partiendo de la base de que casi todos tienen más o menos la misma complejidad a la hora de ser descritos, se han querido seleccionar unos términos que pudiesen abarcar todos los conceptos en cuanto a posibles características y que no fuesen demasiado tediosos de explicar.

La representación textual describe a modo escrito (Fig. 22-23) como su propio nombre indica, cada uno de los elementos que forman BioPax o en otras palabras los tipos de datos existentes en el dominio. Su definición viene determinada por:

- ***Definition:*** una definición que describe qué es el elemento.
- ***Comment:*** un comentario que sirve como aclaración de la definición arriba proporcionada.
- ***Example:*** un ejemplo de elemento real para que el experto tenga más claros los conceptos y pueda asociarlos a algún término real.
- ***Parent class:*** una clase padre o, en otras palabras, término que indica que el elemento es una especialización de otra clase superior. Como se puede observar en las figuras 8 y 9, el elemento Entity es la raíz del esquema por lo que en su definición no contiene la propiedad Parent class lo que indica que no es especialidad de ninguna clase; por contraposición, el elemento Gene que es una especialización de Entity si que contiene esa propiedad.

- ***Properties:*** una serie de propiedades del elemento. Por cada una de las propiedades, la definición textual de BioPax proporciona:

    - ***Cardinalidad:*** manera en la que cada elemento participa en la relación (mínima, máxima). Normalmente la cardinalidad se indica dependiendo de la propiedad, es decir si es el tipo del objeto al que hace referencia es un tipo primitivo (Integer, String…) la definición de la cardinalidad viene proporcionada textualmente mientras que si es una instancia de un objeto de otro elemento se indica la cardinalidad siguiendo el formato (min or max). Este mecanismo de descripción de la cardinalidad no es muy riguroso, ya que existen excepciones. En caso de no encontrar la cardinalidad en esta descripción, se consulta el diagrama de propiedades de objetos y en caso fallido se consulta el archivo con la descripción OWL.

    - ***Tipo del objeto al que hace referencia:*** al igual que en el caso anterior, el tipo de objeto al que hace referencia viene determinado dependiendo de si el tipo es primitivo (Integer, String,..), en este caso la descripción del tipo tiene que extraerse por la descripción textual del documento o si es una instancia de un objeto de otro elemento de BioPax, en este caso el tipo es representado siguiendo el siguiente formato “object: TipoObjeto”. Igual que ocurría con la cardinalidad, la descripción no es rigurosa para definir el tipo de cada elemento y en este caso el tipo no está proporcionado por la descripción textual por lo que se debe acudir al documento OWL de BioPax para buscarlo.

    - ***Descripción:*** representa la descripción de cada una de las propiedades del elemento.

Gene

Definition: An entity that encodes information that can be inherited through replication. This is a generalization of the prokaryotic and eukaryotic notion of a gene. N.B. this is used only for genetic interactions (class **GeneticInteraction**), gene expression regulation makes use of **DNA**, **RNA**, **DnaRegion** and **RnaRegion** physical entities.
Comment: A gene is not a physical entity, but both genes and physical entities are continuants, as defined by most top level ontologies. **Gene** and **PhysicalEntity** classes are conceptually similar, though there is no continuant class in BioPAX to group them.
Examples: The BRCA1 gene
Parent class: Entity
Properties: *organism*, availability, comment, *dataSource*, *evidence*, name, *xref*

organism - (0 or 1 object:BioSource) An organism, e.g. 'Homo sapiens'. This is the organism that the gene is found in.

Fig. 22 Representación textual del elemento Gene

Entity (ontology root class)

Definition: A discrete biological unit used when describing pathways.
Comment: This is the root class for all biological classes in the ontology, which include pathways, interactions, physical entities and genes.
Synonyms: thing, object, bioentity.
Properties: availability, comment, *dataSource*, *evidence*, name, *xref*

availability - Describes the availability of this data (e.g. a copyright statement). The availability statement applies to the instance it is attached to and all children instances. For example, the availability statement on a **Pathway** instance applies to the pathway and all elements of the pathway (proteins, evidence, xrefs).

comment - Comment on the data in the container class. This property should be used instead of the OWL documentation elements (rdfs:comment), as OWL metadata properties are not required to be recognized by BioPAX tools.

dataSource - (0 or 1 object:Provenance) A description of the source of this data, e.g. a database or person name. This property should be used to describe the source of the data by database groups that export their data to the BioPAX format or by systems that are integrating data from multiple sources. Similar to the availability property, the data source applies to the instance it is attached to and all children instances. This property reports the last data source, not all data sources that the data has passed through from creation. Further described in the **Provenance** class documentation.

evidence - (0 or 1 object:Evidence) Scientific evidence supporting the existence of the entity. Further described in the **Evidence** class documentation.

name - One or more names of this entity. This will automatically include values of the *displayName* and *standardName* properties, as they are child properties of the name property. *displayName* values are short names suitable for display in a graphic. Standard names are names that follow a standard nomenclature, like systematic yeast ORF names (e.g. YJL034W).

xref - Values of this property define external cross-references from this entity to entities in external databases. Further described in the Xref class documentation.

Fig. 23 Representación textual del elemento Entity

Por otro lado el diagrama de propiedades de objetos es el formato que ha creado el equipo BioPax para representar de manera visual las relaciones entre objetos y sus cardinalidades, ayudando a complementar la información de la descripción textual:

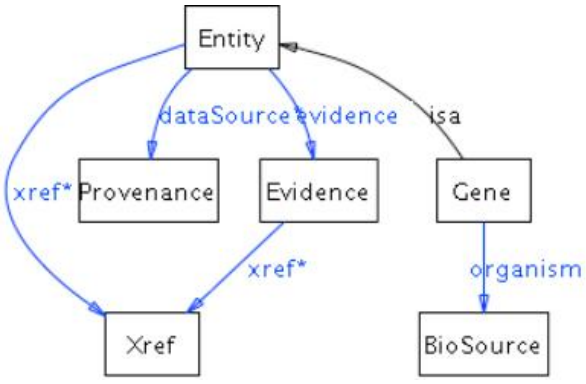


Fig. 24 Diagrama de propiedades de objetos de los elementos Entity y Gene

Cada rectángulo del diagrama representa un elemento de BioPax, las relaciones negras con el nombre isa representan una relación de especialización mientras que las flechas azules representan las relaciones entre términos del lenguaje. Además en las flechas azules la cardinalidad máxima viene representada por el símbolo * que significa que la cantidad de objetos de ese tipo que puede contener el elemento es ilimitado. Este tipo de diagramas a veces presenta inconsistencias con la representación textual, por lo que en caso de duda, siempre se debe consultar el documento OWL proporcionado. La figura 25 representa la relación entre ambos formatos de representación.

A pesar de que los expertos de BioPax hayan proporcionado una documentación para una mejor comprensión del lenguaje, no es demasiado rigurosa, ya que en algunas ocasiones como se ha comentado los tipos y las cardinalidades o se desconocen o se contradicen en ambas representaciones por lo que la dificultad del trabajo de interpretación se incrementa. Por otra parte cabe destacar que las relaciones entre clases solo se proporcionan en un sentido, por lo que se ha considerado que si no se proporciona esta información es porque la cardinalidad en sentido contrario puede tomar todos los valores posibles, o en otras palabras, toma la cardinalidad mínima, de 0…* que es la menos restrictiva.

Siguiendo con la descripción del capítulo, se van a indicar cuáles son los principales elementos del UML y una vez extraídos, se va a realizar la correspondencia con las definiciones arriba proporcionadas. Los elementos de la representación UML para el modelado conceptual de BioPax son: las clases, sus atributos, sus relaciones con otras clases, las cardinalidades tanto de propiedades como de relaciones y la herencia.

- ***Clases:*** encontramos una clase UML en la definición de BioPax cuando el concepto el cual representa es un término con un significado en el dominio y necesita ser descrito. Dichos términos están bien localizados ya que el documento los describe uno a uno. En nuestro ejemplo hemos seleccionado dos términos del documento BioPax, *Entity* y *Gene*, que pasarán a ser clases en el modelo UML.

- ***Atributos:*** dentro de las propiedades de cada uno de los términos representados, se distingue un grupo cuyo tipo de objeto es primitivo o en otras palabras es de tipo String, Integer… Estas propiedades son las que en el modelo conceptual se representarán como atributos dentro de las clases. En este ejemplo, los atributos que encontramos son: *availability*, *comment* y *name*, los tres de la clase *Entity*, por lo que dicha clase contendrá esos tres atributos. Respecto a la cardinalidad de los dos primeros atributos, *availability* y *comment*, cabe destacar que tienen una descripción en lenguaje natural y que en ningún momento se especifica con claridad ninguna de las dos propiedades del atributo. A pesar de ello de la descripción del documento se extrae que ambas son de tipo String y del fichero OWL se extrae que la cardinalidad de ambas es de 0…1. Respecto a la cardinalidad del atributo *name*, cabe destacar que de manera textual especifica que este atributo puede tomar valores de 1…* y como en el resto de atributos se extrae que el tipo es String.

- ***Relación de asociación:*** dentro de las propiedades de cada elemento encontramos también las relaciones entre clases. Este tipo de propiedades se distinguen de los atributos en que el tipo de objeto que tienen asociado es una instancia de otra clase existente en la descripción. Estas propiedades se representaran en el modelo conceptual como relaciones con otras clases. En este ejemplo las relaciones son las descritas por las propiedades: *dataSource*, *evidence*, *xref* y *organism*. Las tres primeras relaciones pertenecen a la clase *Entity* y cada una de ellas va asociada con otro término de la definición de BioPax que pasará a ser una clase del modelo conceptual, en este caso son *Provenance*, *Evidence* y *Xref* respectivamente. La última de las relaciones pertenece a la clase *Gene* y va asociada al concepto *BioSource* de BioPax que pasará a ser otra clase del modelo conceptual. Respecto a la cardinalidad de las relaciones, cabe destacar que la mayoría de estas están especificadas, *dataSource* de 0…1, *evidence* de 0…1 y *organism* de 0…1 también, pero por el contrario *xref* no tiene una cardinalidad especificada, por lo que habrá que consultarlo en el fichero

OWL de BioPax que indica que es de 0…*. Como se ha comentado anteriormente, las relaciones solo van especificadas en un sentido, por lo que el sentido contrario de la relación no especificado, tomará el valor de 0…* o valor mínimo de restricción, en todas las ocasiones.

- ***Relación de herencia:*** la relación de herencia se encuentra entre objetos que comparten características comunes pero que además tienen sus propias particularidades. Este es el caso del término Gene que comparte todas las propiedades con *Entity* y que además contiene su propia propiedad distintiva que es *organism*. Se interpreta de la documentación de BioPax por la propiedad Parent Class de la descripción textual y por las flechas negras de nombre *isa* en el diagrama de propiedades de objetos.

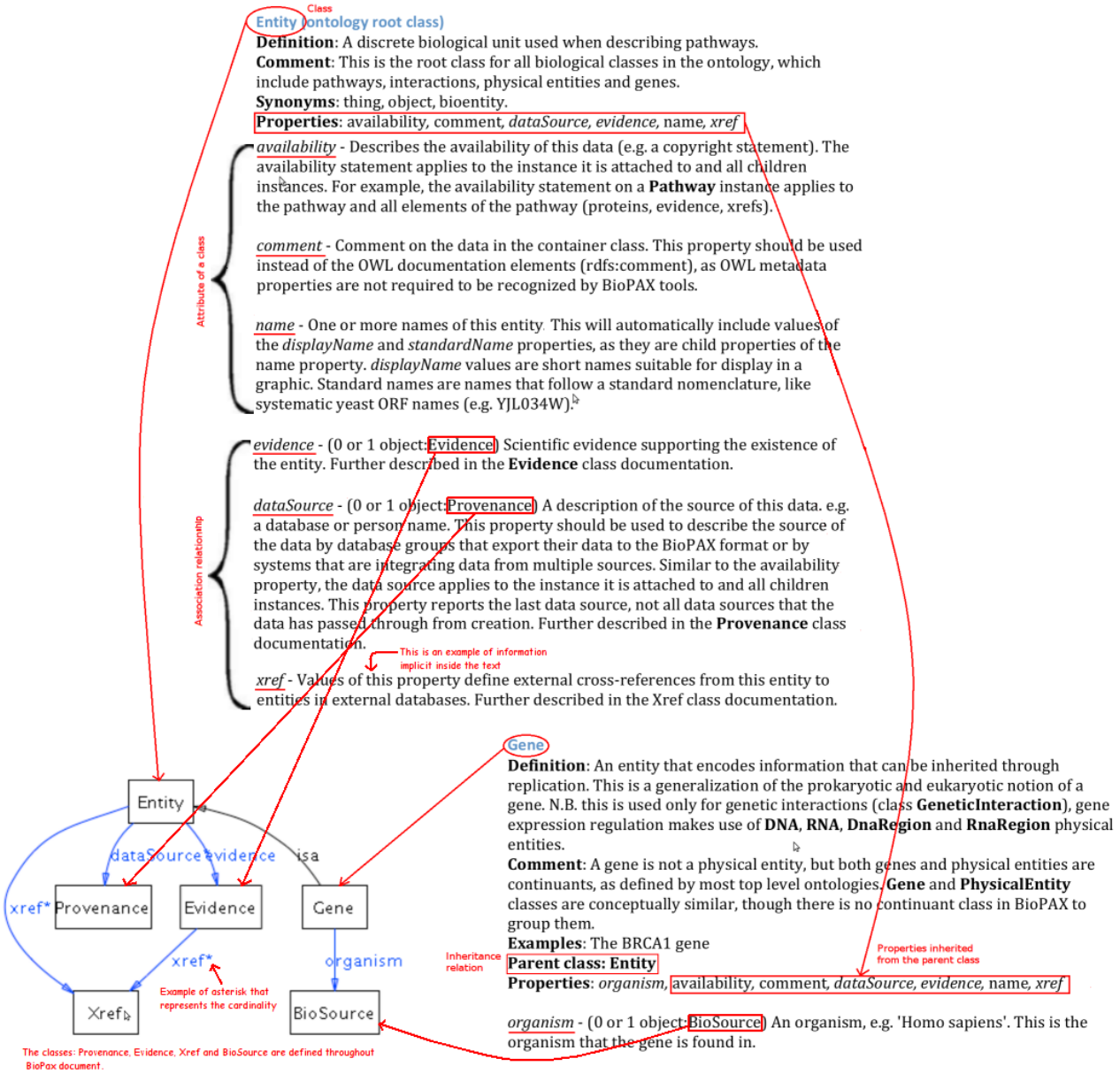


Fig. 25 Relación entre la representación textual y el diagrama de propiedades de objetos

Una vez extraída la correspondencia entre la propuesta BioPax y el lenguaje de representación de modelado conceptual UML se obtiene el modelo de la figura 26.

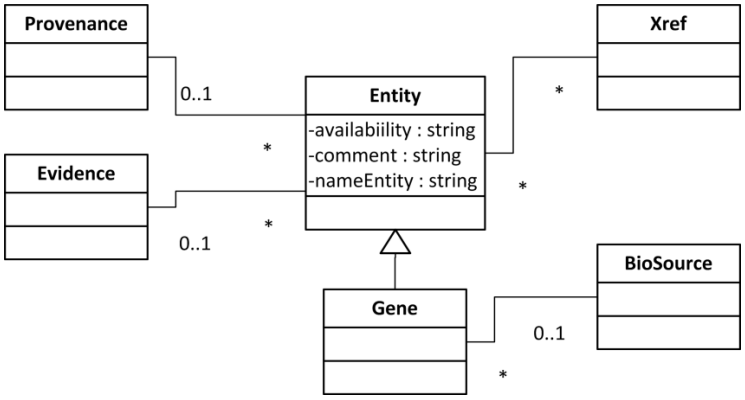


Fig. 26 Modelo conceptual de BioPax de los conceptos Entity y Gene

Como se ha comentado anteriormente si extrapolamos la solución presentada para traducir la propuesta BioPax a UML de manera que abarque toda la información mostrada el documento, se obtiene un amplio modelo conceptual que se presenta en la figura 30. Como se puede observar, debido a su gran cantidad de conceptos biológicos, su diseño y documentación se realizará por vistas a continuación para facilitar su comprensión.

Como se ha comentado arriba, para describir mejor el modelo conceptual, diseñado a partir de la propuesta BioPax, se va a dividir en tres vistas: la vista principal que define los aspectos primordiales de BioPax, la vista de interacción que define los tipos de interacciones existentes en las rutas metabólicas y la vista de entidades físicas que determina los tipos de estructuras que participan en cada proceso de interacción.

5.3.2 Vista central de BioPax

BioPax se define esencialmente utilizando cinco clases básicas, la clase *Entity* que se define como el objeto que envuelve a todo el resto de clases del modelo y las cuatro subclases que se especializan de ella y que son: *Pathway*, *Interaction*, *PhysicalEntity* y *Gene*. Además cada objeto *Entity* incluye información acerca de la disponibilidad de los datos si son públicos o privados, atributo *availability*, una breve descripción, atributo *comment*, la fuente externa de la que proceden dichos datos, relación clase *Provenance*, identificador externo que proporciona la fuente del dato en cuestión, relación clase *Xref* y publicaciones asociadas al dato que permitan demostrar su existencia, relación clase *Evidence*.

La figura 27 muestra estas clases junto a sus atributos y clases de utilidad asociadas que proporcionan características de cada una de ellas. Estas clases de utilidad son estructuras de datos complejas que constan de diversos atributos, por lo que se representan como clases aparte en lugar de atributos. Se representan sombreadas en el dibujo.

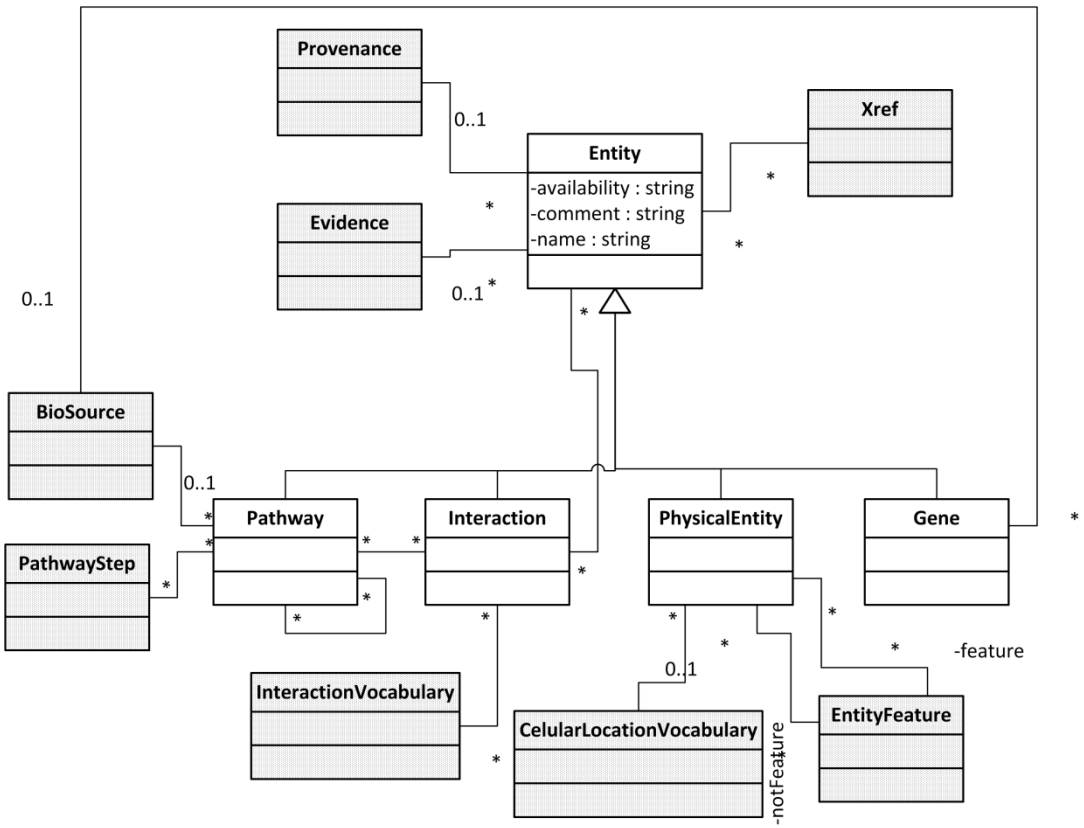


Fig. 27 Vista central de BioPax

A continuación, se ofrece una descripción sencilla de cada una de las clases principales de la vista para comprender de una manera más fácil la descripción que BioPax proporciona:

- **Pathway**: conjunto de interacciones, a menudo formando una red, que los biólogos han considerado interesante estudiar. Los pathways pueden estar formados por interacciones u otros pathways y pueden ser descompuestos en sub-pathways por los que está formado. Un ejemplo de este tipo de procesos es el de la glicolisis. Además, dado que los pathways suceden dependiendo de la especie y está formado por muchos procesos es interesante conocer el orden en el que se suceden, relación clase *PathwayStep* y el organismo en el que se produce, relación clase *BioSource*.

- **Interaction**: relación atómica entre una o más entidades, es decir proceso que ocurre entre dos entidades y que no puede ser descompuesto en sub-interacciones. Por ejemplo una interacción proteína-proteína.

- **PhysicalEntity**: conjunto de entidades que tiene una estructura física. El grupo de entidades puede estar constituido por una única entidad o ser un conjunto de entidades complejas agrupadas entre sí. Estas entidades físicas poseen, o por el contrario carecen, de ciertas características que son interesantes conocer a la hora de que tenga lugar las interacciones, estas características se representan mediante la clase *EntityFeature*.

- **Gene**: representa lo que hasta ahora se ha entendido también como gen durante esta tesis, una entidad que codifica información que puede ser heredada a través de la replicación. Como los genes varían entre especies es interesante conocer a cual pertenece cada uno, relación objeto *BioSource*.

Dentro de BioPax existen una serie de vocabularios que se utilizan para nombrar de manera estándar ciertos aspectos de las rutas como por ejemplo la localización celular o el fenotipo. En esta versión del

modelo no se han tenido en cuenta estos vocabularios, por lo que a partir de ahora no se entrará al detalle en dichas clases.

Una vez modelada la vista central de BioPax, se ha decidido dividir el resto del modelo conceptual en vistas de acuerdo a la clasificación de dos de los tipos de entidades existentes que se han definido arriba: *Interaction* y *PhysicalEntity* que son las clases que tienen gran volumen de datos asociados.

5.3.3 Vista de interacción de BioPax

La vista de interacción de BioPax (Fig. 28) describe la relación existente entre una o más entidades cuya función se lleva a cabo de manera atómica, es decir no se puede dividir en sub-procesos. Su clase principal es *Interaction* y es abstracta, ya que en todos los casos es más apropiado utilizar una de las subclases de ella misma para definir un tipo de interacción.

Las subclases en las que se especializa la clase *Interaction* son:

- **Control**: interacción en la cual una entidad regula, modifica o influencia de alguna manera a otra. Se conocen dos tipos de interacciones, las que activan un proceso o las que lo inhiben. Las interacciones de tipo control a su vez se dividen en:
    - **Catalysis**: interacción de tipo *Control* en la cual una entidad física incrementa la velocidad de una interacción de tipo conversión mediante la reducción de su energía.
    - **Modulation**: interacción de tipo *Control* en la cual una entidad física modula una interacción de catálisis.
    - **TemplateReactionRegulation**: interacción de tipo *Control* que regula una estructura reacción con estructura conocida a través de una entidad física.
- **Conversion**: interacción en la cual una o más entidades físicas son físicamente transformadas en una o más diferentes. Las interacciones de tipo *Conversion* a su vez se dividen en:
    - **BiochemicalReaction**: interacción de tipo *Conversion* en la cual una o más entidades sufre cambios covalentes físicos para convertirse en una o más entidades.
    - **ComplexAssembly**: interacción de tipo *Conversion* en la cual un conjunto de entidades físicas, al menos una de ellas macromolécula, forman una entidad física de tipo complejo.
    - **Degradation**: interacción de tipo *Conversion* que muestra el proceso de degradación de una entidad física convirtiéndose ésta en una serie de componentes degradados no especificados.
    - **Transport**: interacción de tipo *Conversion* en la cual una entidad física cambia de localización dentro de la célula o con respecto a ella.
    - **TransportWithBiochemicalReaction**: interacción de tipo *Transport* y *BiochemicalReaction* en la que una o más componentes cambian su localización así como su estructura física.
- **GeneticInteraction**: interacciones genéticas ocurridas entre genes que se producen cuando dos perturbaciones genéticas tienen un efecto fenotípico combinado que no se hubiese sido causado por una de ellas sola.
- **MolecularInteraction**: interacción que tiene lugar cuando se produce en contacto molecular entre entidades físicas. El mecanismo exacto de este tipo de interacciones no suele ser conocido.

- TemplateReaction: interacción donde una macromolécula es polimerizada por una macromolécula con estructura conocida.

Las clases sombreadas representan por una parte características propias de cada una de las clases arriba definidas que sirven, de la misma manera que los atributos, para describirlas y por otra parte unión con la vista central de BioPax representada con la clase *Entity*.

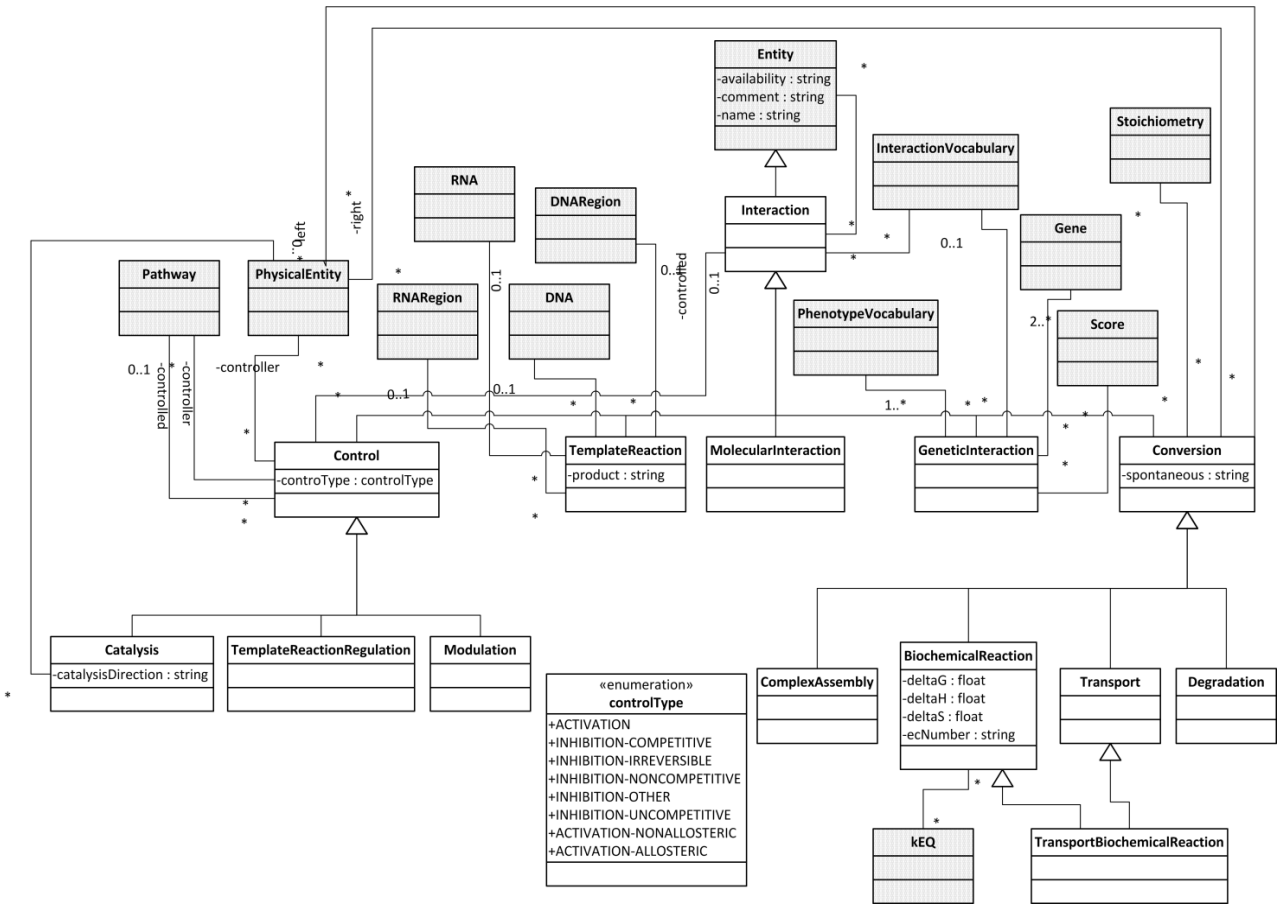


Fig. 28 Vista de interacción de BioPax

5.3.4 Vista de entidades físicas de BioPax

La vista de entidades físicas (Fig. 29) representa un grupo de entidades, cada una de ellas con su propia estructura física diferente, que cubren algunos aspectos estructurales del dominio del genoma. Existen varios tipos de entidades físicas que son:

- **Complex**: entidad física que está formada por otras entidades físicas más simples.
- **DNA**: entidad física que consiste en una secuencia de ácido desoxirribonucleico (ADN).
- **Protein**: entidad física que consiste en un secuencia de aminoácidos.
- **RNA**: entidad física que consiste en una secuencia de ácido ribonucleico (ARN).
- **SmallMolecule**: pequeñas moléculas bioactivas.

- **DNARegion**: región específica de la secuencia de ADN.
- **RNARegion**: region específica de la secuencia de ARN.

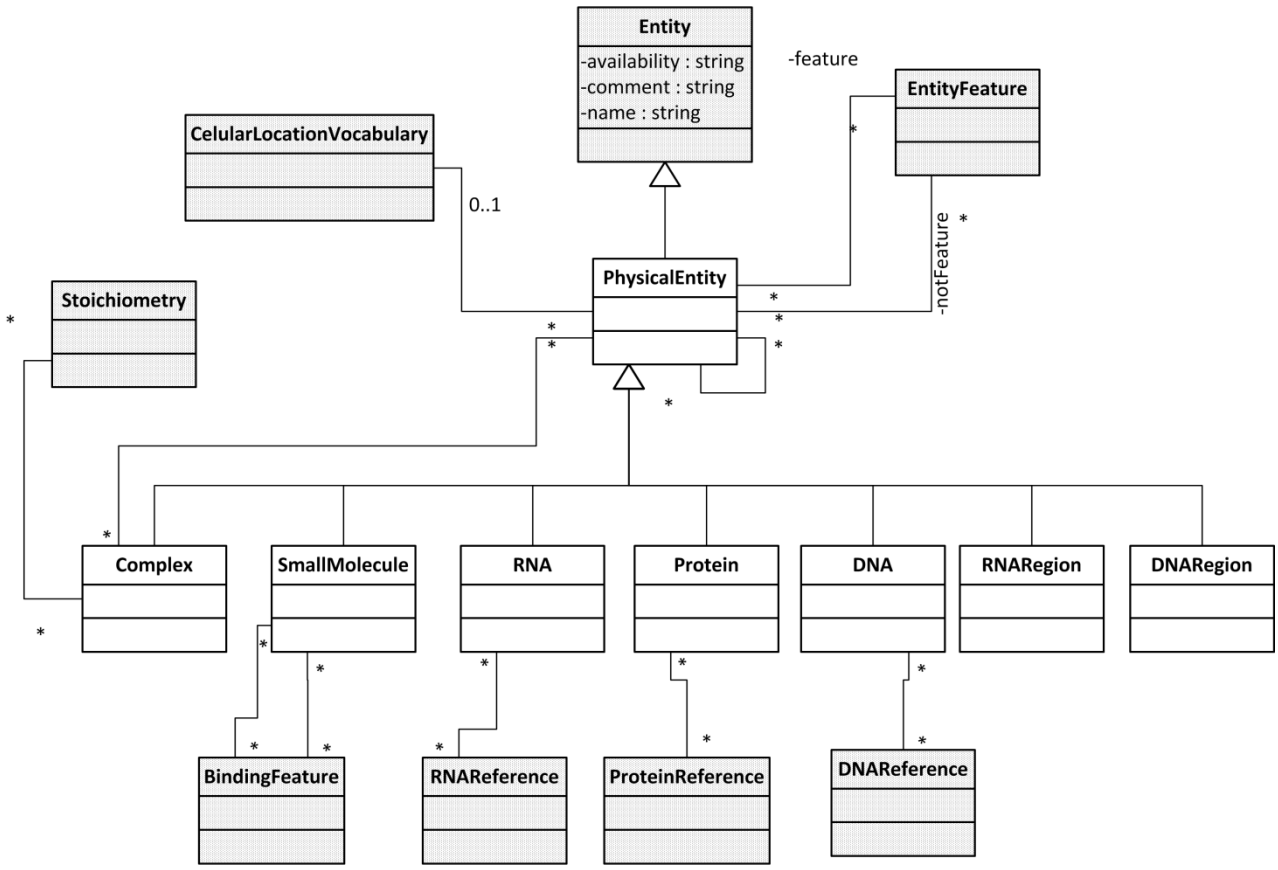


Fig. 29 Vista de entidades físicas de BioPax

Cabe destacar que las clases sombreadas son, igual que en la vista anterior, por una parte, clases de utilidad que sirven definir propiedades de las propias clases que sí forman la estructura de BioPax y por otra, clases de unión entre la vista principal de BioPax, *Entity*.

5.3.5 Modelo conceptual de BioPax

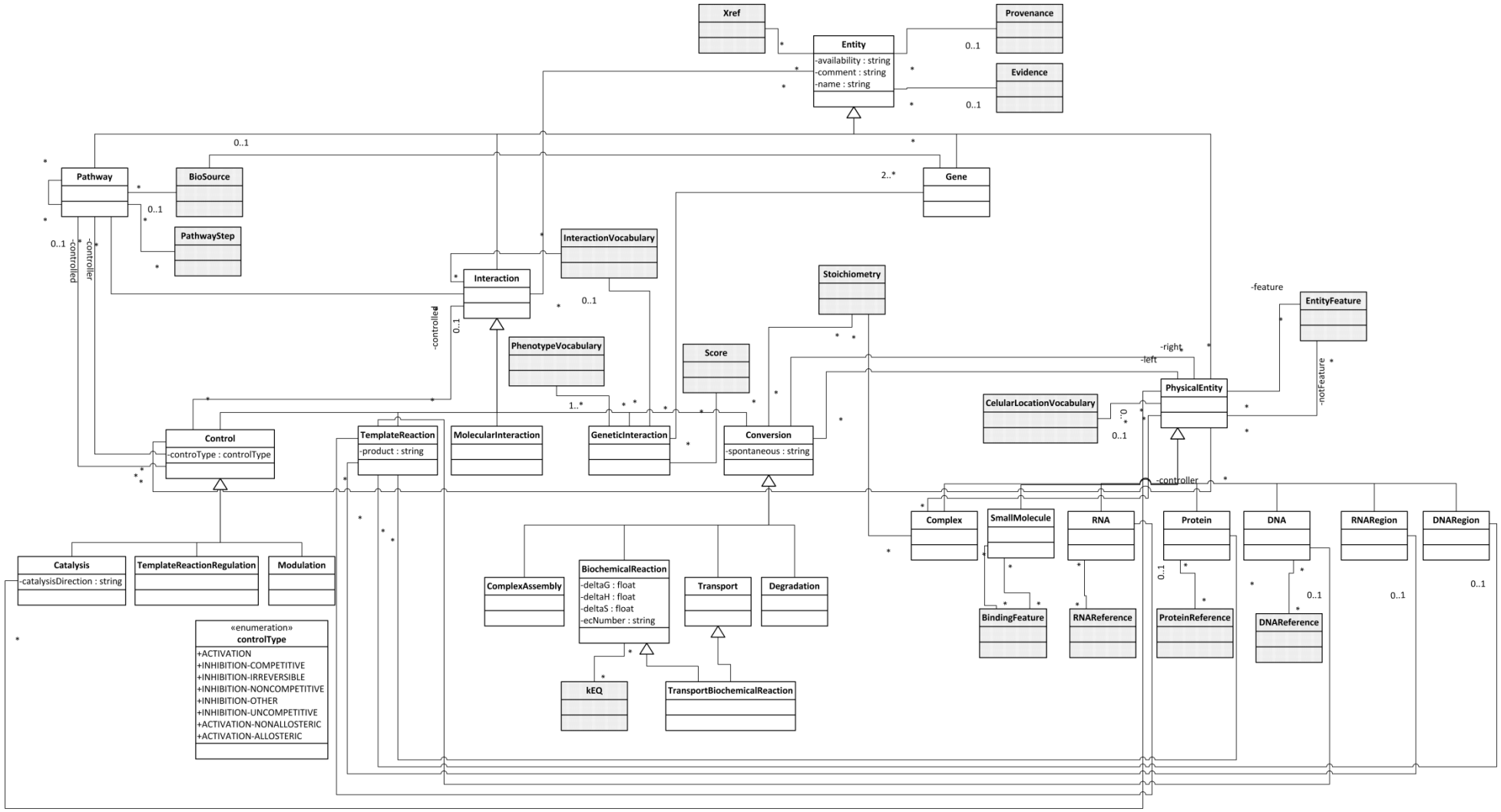


Fig. 30 Vista completa del modelo conceptual de BioPax

5.4 Integración de la propuesta BioPax en el modelo conceptual del genoma

La integración de la propuesta BioPax en el modelo conceptual del genoma se va a realizar por vistas de acuerdo a la descripción propuesta en las secciones del apartado 5.3. En primer lugar se realiza la integración de la parte central de BioPax que incluye las cuatro clases principales y luego se integrarán la vista de las interacciones y la vista de las entidades físicas respectivamente. Finalmente se proporcionará una vista completa del modelo conceptual del genoma con esta nueva vista adaptada.

Cabe destacar que la información proporcionada por BioPax llega a niveles más profundos que la información proporcionada por el modelo conceptual del genoma. De esta manera, aunque el conocimiento que se modela es el mismo, BioPax proporciona información mucho más detallada de cada uno de los elementos.

5.4.4 Integración de la vista central de BioPax

Para realizar la integración de la parte central del modelo BioPax en el modelo conceptual del genoma se muestra una imagen de los dos modelos (fig. 31), cuya parte izquierda es el modelo conceptual del genoma y cuya parte derecha es el modelo conceptual de la propuesta BioPax. La vista central del modelo BioPax se compone principalmente de una clase llamada *Entity,* que es una clase genérica que se especializa en cuatro tipos de elementos que expresan de manera completa los procesos de las rutas metabólicas, y de sus cuatro tipos de elementos especializados: *PhysicalEntity*, *Interaction*, *Pathway* y *Gene*. Por otra parte, el modelo conceptual del genoma tiene este conocimiento representado utilizando una modelización un tanto distinta, por un lado se encuentra la clase *Event*, que podría corresponderse con la clase *Entity* del modelo BioPax, aunque no en su totalidad, que se especializa en dos clases, *Process* y *Pathway* y por otro lado la clase llamada *Entity*. Ambas clases *Entity* y *Event* del modelo conceptual del genoma contienen el atributo *name* al igual que la clase *Entity* del modelo BioPax. Cabe destacar que la clase *Provenance* del modelo conceptual BioPax hace referencia a la clase *Data Bank Entity identification* del modelo conceptual del genoma, proporcionando información acerca de la fuente de datos de la cual se ha extraído información, pero en el modelo de BioPax esta información se extrapola a todas las instancias del tipo *Entity*, no solo a las estructuras físicas. Por otra parte, también merece una mención la clase *blibliography reference* del modelo conceptual del genoma que hace referencia a la clase *Evidence* del modelo BioPax indicando las publicaciones que demuestran la evidencia de que esos hechos han sido constatados. Además, el atributo *comment* de la clase *Entity* del modelo BioPax se ve representado únicamente en la clase *take_part*, con el nombre de *notes*, que relaciona los procesos con sus respectivas entidades físicas. Los atributos y clases objeto asociadas, *availability* y *Xref* no se han tenido en cuenta a la hora de diseñar el modelo conceptual del genoma.

Por otra parte, la clase *Pathway* del modelo conceptual del genoma se corresponde directamente con la clase *Pathway* del modelo conceptual BioPax, ambas clases representan un conjunto de interacciones que tienen lugar conjuntamente dando lugar a un objetivo. Esta definición está representada en ambos modelos mediante relaciones que permiten a un elemento de tipo pathway estar formado por muchos elementos de tipo pathway o proceso/interacción. Además la correspondencia entre la clase *BioSorce*, que ofrece información acerca del tipo de organismo en el que tiene lugar el proceso, se extrae en el modelo conceptual del genoma a partir de las entidades físicas que ayudan a que el proceso o pathway tenga lugar.

Cada entidad física se encuentra modelada en la vista estructural de modelo conceptual del genoma, y se encuentra asociada una especie, por lo que dicha información queda extraída a partir de dichas relaciones.

Además, la clase *Process* en el modelo conceptual del genoma, se corresponde con la clase *Interaction* del modelo conceptual BioPax, representando ambas un proceso atómico en el que forman parte dos o más estructuras físicas. Esta definición queda representada en ambos modelos pudiendo estar una interacción o proceso en ambos modelos formado por una serie de entidades físicas. En el modelo conceptual del genoma esta relación se modela mediante la clase intermedia *takes_part* entre los conceptos *Entity* y *Process*, mientras que en el modelo de BioPax se relaciona mediante una relación muchos a muchos entre las dos clases *Entity* e *Interaction*.

La relación muchos a muchos que existe en el modelo conceptual entre la clase *Event* y ella misma que sirve para especificar en qué posición ocurre cada proceso en la ruta metabólica, queda representada en el modelo de BioPax mediante la clase de utilidad *PathwayStep* que ordena los procesos existentes en cada ruta.

Por otra parte, la clase *Entity* del modelo conceptual del genoma se corresponde con la clase *PhysicalEntity* del modelo de BioPax, haciendo referencia a las estructuras físicas nombradas en el capítulo 4. Cabe destacar que en BioPax se modela la clase *Gene* fuera de la clase *PhysicalEntity* porque se considera que un gen no es una estructura física y además es utilizado simplemente para interacciones genéticas haciendo uso de entidades físicas. En el modelo conceptual del genoma la clase *gene* está representada en la vista estructural, vista que define la composición del genoma. Las características relevantes que posee o de las que carece una entidad física para que la interacción tenga lugar no han sido tenidas en cuenta al nivel en el que se ha llevado a cabo el trabajo.

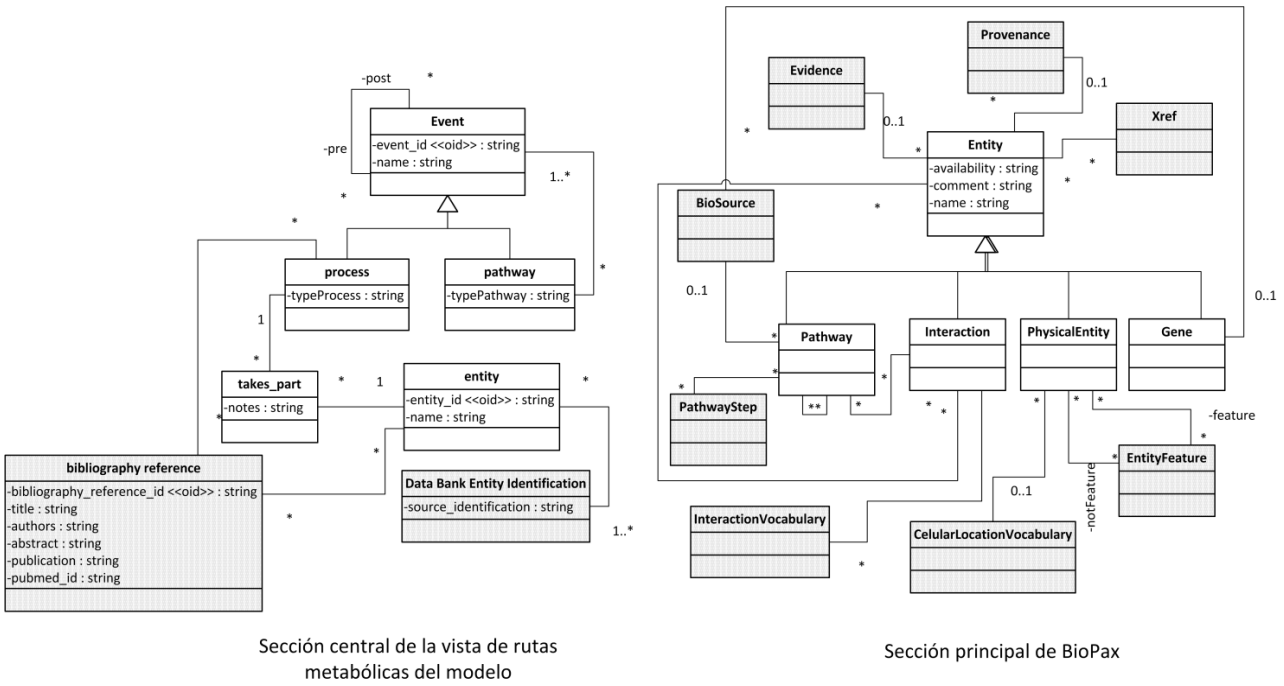


Fig. 31 Vista central: modelo conceptual del genoma vs Modelo conceptual BioPax

5.4.5 Integración de la vista de interacción de BioPax

Como se ha comentado en el apartado anterior de la integración de la vista central de BioPax, la clase *process* en el modelo conceptual del genoma es la equivalente a la clase *Interaction* del modelo BioPax (Fig. 32). A primera vista, puede parecer que el número de clases entre un modelo y otro tiene una diferencia abismal, esto es debido a que BioPax proporciona una especialización de clases de tipo interacción que es mucho más profunda que la del modelo conceptual de genoma. Estas clases son *GeneticInteraction*, *MolecularInteraction*, *TemplateReaction*, *Control* y *Conversion*, las dos últimas divididas también en *Catalysis*, *TemplateReactionRegulation*, *Modulation*, *ComplexAssembly*, *BiochemicalReaction*, *Degradation* y *Transport*. Por el contrario, en el modelo conceptual del genoma este conocimiento queda representado mediante el atributo *typeProcess* y solo queda modelada la clase *Catalysis*.

Respecto a las relaciones entre las interacciones y las estructuras físicas cabe destacar que tanto en un modelo como en otro, existen tipos de interacciones que se relacionan únicamente con un tipo de entidad física concreta que se representan en sus respectivos modelos añadiendo una relación de asociación entre ellas. Debido a que el modelo BioPax va más allá en cuanto a profundidad se refiere y existen estudios que demuestran que hay casos en los que en una reacción de tipo catálisis pueda estar involucrada otro tipo de entidad que no sea de tipo enzima, la relación entre *catalysis* y *enzime* del modelo conceptual del genoma se sustituye por una relación directa entre *Catalysis* y *PhysicalEntity* que abarca todas las estructuras. Además el atributo *EC number,* nomenclatura de enzimas, asignado a la clase catalysis por su relación directa con la clase *enzyme* queda representado en el modelo conceptual de BioPax en la clase BiochemicalReaction debido a que todas las relaciones enzimáticas deben ser reacciones bioquímicas.

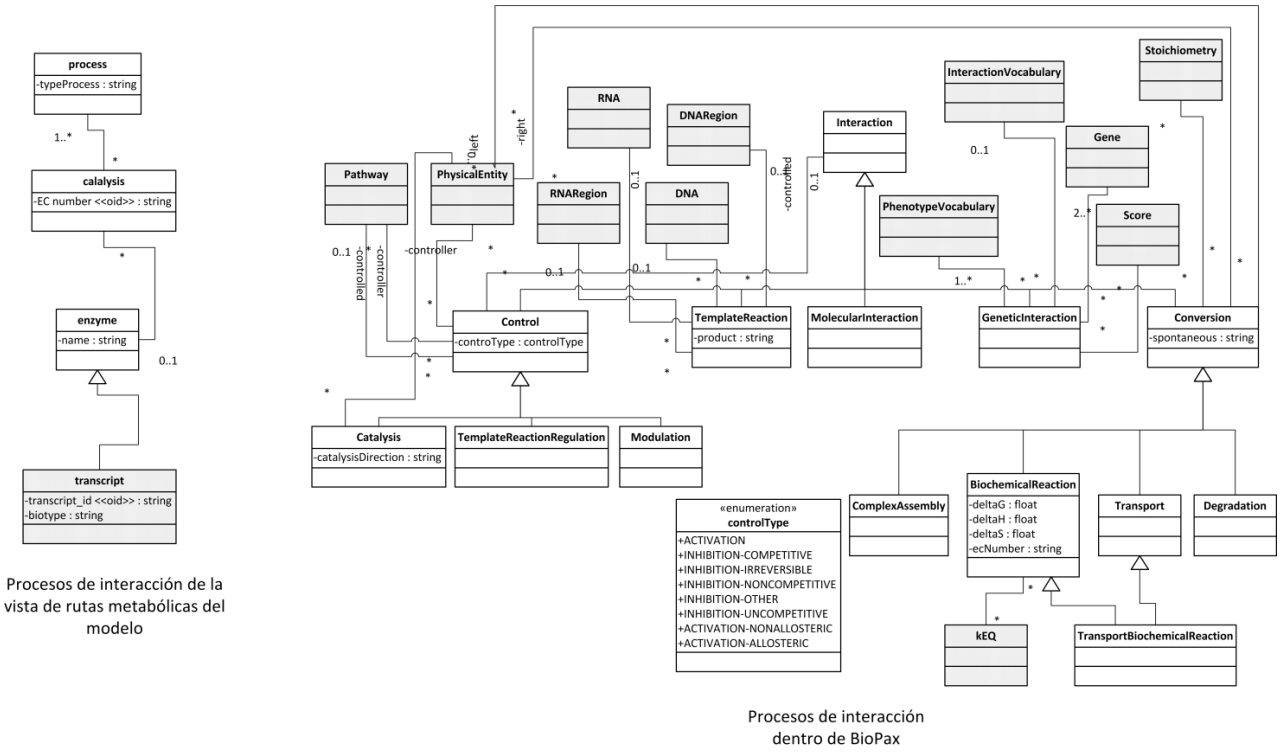


Fig. 32 Vista de interacción: modelo conceptual del genoma vs modelo conceptual BioPax

5.4.6 Integración de la vista de entidades fisicas de BioPax

Como se ha comentado en el apartado en el que se describía la integración de la parte central de los modelos de rutas metabólicas, la clase *entity* del modelo conceptual del genoma es la representada por la clase *PhysicalEntity* del modelo de BioPax (Fig. 33). Respecto a sus especializaciones, la clase *complex* tiene el mismo significado en ambos modelos y, mientras que sus características en el modelo conceptual del genoma están representados como atributos (*stoichiometry* y *interaction*_type), en el modelo conceptual de BioPax están representados con la clase externa *Stoichiometry* y una relación directa a la clase *PhysicalEntity* que representa que una entidad compleja puede ser de cualquier tipo especializado a partir de ella. Por otra parte, la relación *entitySet* del modelo conceptual del genoma queda representada en el modelo conceptual de BioPax mediante la relación que mantiene la clase *PhysicalEntity* consigo misma. Respecto a la clase *simple* y sus especializaciones cabe destacar que en el modelo de BioPax la clase simple desaparece y las clases que hacen referencia a dichas especializaciones se especializan directamente de la clase *PhysicalEntity*. De esta manera las clases *chromosome element* y *nucleotide_e* hacen referencia a las clases *DNA* y *DNARegion*, la clase *transcript* hace referencia a las clases *RNA* y *RNARegion*, las clases *protein* y *aminoacid_e* hacen referencia a la clase *Protein* y *basic_e* hace referencia a la clase *SmallMolecule*. Por otra parte, la clase *polymer* queda representada mediante las características proporcionadas por la clase *EntityFeature*.

Finalmente, las clases *RNAReference*, *ProteinReference*, *DNAReference* y *BindingFeature* se corresponden en el modelo conceptual del genoma con la vista estructural presentada en el capítulo 4.1 que define los conceptos que forman el genoma y sus principales características.

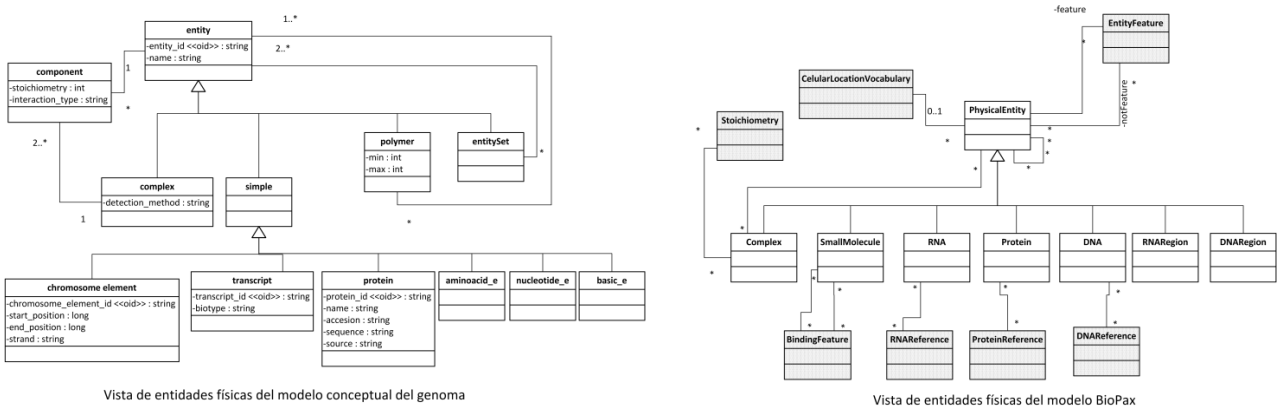


Fig. 33 Vista de entidades físicas: modelo conceptual del genoma vs modelo conceptual BioPax

5.4.7 Integración completa de BioPax en el modelo conceptual

La integración del modelo BioPax dentro del modelo conceptual del genoma (Fig. 34) ha sido relativamente sencilla debido a que el conocimiento que se expresa en ambos modelos con respecto a rutas metabólicas es el mismo, pero utilizando diversas representaciones de la realidad y diferentes niveles de abstracción. Para una mejor comprensión de la integración de ambos modelos se han tenido en cuenta la jerarquía que proporciona BioPax con sus clases y atributos asociados, pero las clases unión entre ambos modelos han sido representadas con respecto a la modelización llevada a cabo en el modelo conceptual del genoma. Estas clases unión son: chromosome element, transcript, protein y basic_e que han sido sustituidas por sus semblantes en el modelo conceptual de BioPax. De esta manera la integración con respecto al resto del modelo se hace mucho más sencilla a la hora de entender los conceptos cuando se habla de todo el modelo

conceptual, pero a la hora de cargar los ficheros proporcionados por las bases de datos externas con formato BioPax aparecen dificultades que deben que solventarse. A pesar de que el conocimiento representado sea el mismo, plasmar la realidad es un trabajo muy diferente dependiendo de los puntos de vista de la persona o personas que trabajen en él. Esto es debido a que existen muchas interpretaciones para una misma realidad. Por este motivo, la futura base de datos extraída a partir de este modelo, no se corresponderá 100% con los ficheros BioPax que ofrecen las fuentes externas. La solución a este problema ha sido contemplada y tenida en cuenta durante este trabajo, pero debido a que la demanda de un repositorio que integrase la máxima cantidad de rutas metabólicas era elevada, la base de datos se generó en base a la vista completa del modelo BioPax, dejando como trabajo futuro la manipulación de los ficheros en formato BioPax antes de su inserción en la base de datos totalmente integrada.

5.5 Base de datos de BioPax

Debido a la demanda de una base de datos BioPax que integrase la información de las principales bases de datos existentes en la actualidad sobre rutas metabólicas, la base de datos presentada y cargada en este capítulo se basa en el modelo conceptual de BioPax sin haberse realizado la integración en el modelo conceptual del genoma o en otras palabras sin haberse realizado las correspondientes transformaciones en los ficheros formato BioPax para poder cargarse sin problemas en el esquema totalmente integrado.

A partir del modelo conceptual, se puede extraer automáticamente la base de datos mediante herramientas de desarrollo dirigido por modelos (DDM). El DDM se ha convertido en un nuevo paradigma de desarrollo que promete una mejora de la productividad y de la calidad del software a través de un proceso guiado por modelos y soportado por potentes herramientas que generan código a partir de modelos [40]. Este paradigma nace con el objetivo de adaptarse rápidamente a las necesidades cambiantes de los sistemas. Dentro de este paradigma, las tendencias actuales en generación de bases de datos proponen su generación automática a partir de un modelo, disminuyendo así el tiempo de trabajo del desarrollador y reutilizando el conocimiento previamente adquirido.

Así como en la creación de la base de datos presentada en el capítulo 4 no utilizamos ninguna herramienta automática, en este pequeño sistema de información si se ha usado. Esto es debido a que el modelo conceptual se definió completamente antes de la implementación de la base de datos y de la carga de información, fue un proceso lineal 100%. La herramienta de desarrollo seleccionada fue Moskitt [41] ya que es una plataforma libre basada en Eclipse para el DDM que utiliza el lenguaje de modelado UML. De entre sus principios, cabe destacar su predisposición al uso de estándares siempre que sea posible, facilitando la interoperabilidad con otras herramientas y su diseño de arquitectura modular para ser fácilmente extendida y adaptada a cambios futuros. Su funcionalidad es bastante amplia soportando la edición gráfica de modelos así como las transformaciones entre ellos, por lo que se adapta a las necesidades del ámbito de trabajo por el diseño y transformación entre modelos que ofrece. Además cabe destacar que su interoperabilidad es muy sencilla para el usuario lo que facilita el tiempo que el usuario tarda en aprender a usar la herramienta. Otro punto a tener en cuenta es que es una herramienta del mismo grupo de trabajo, ProS, hecho que convierte el trabajo en una labor mucho más satisfactoria, ya que además de esta manera se puede ayudar a solventar posibles errores de uso que con las pruebas no se hayan detectado.

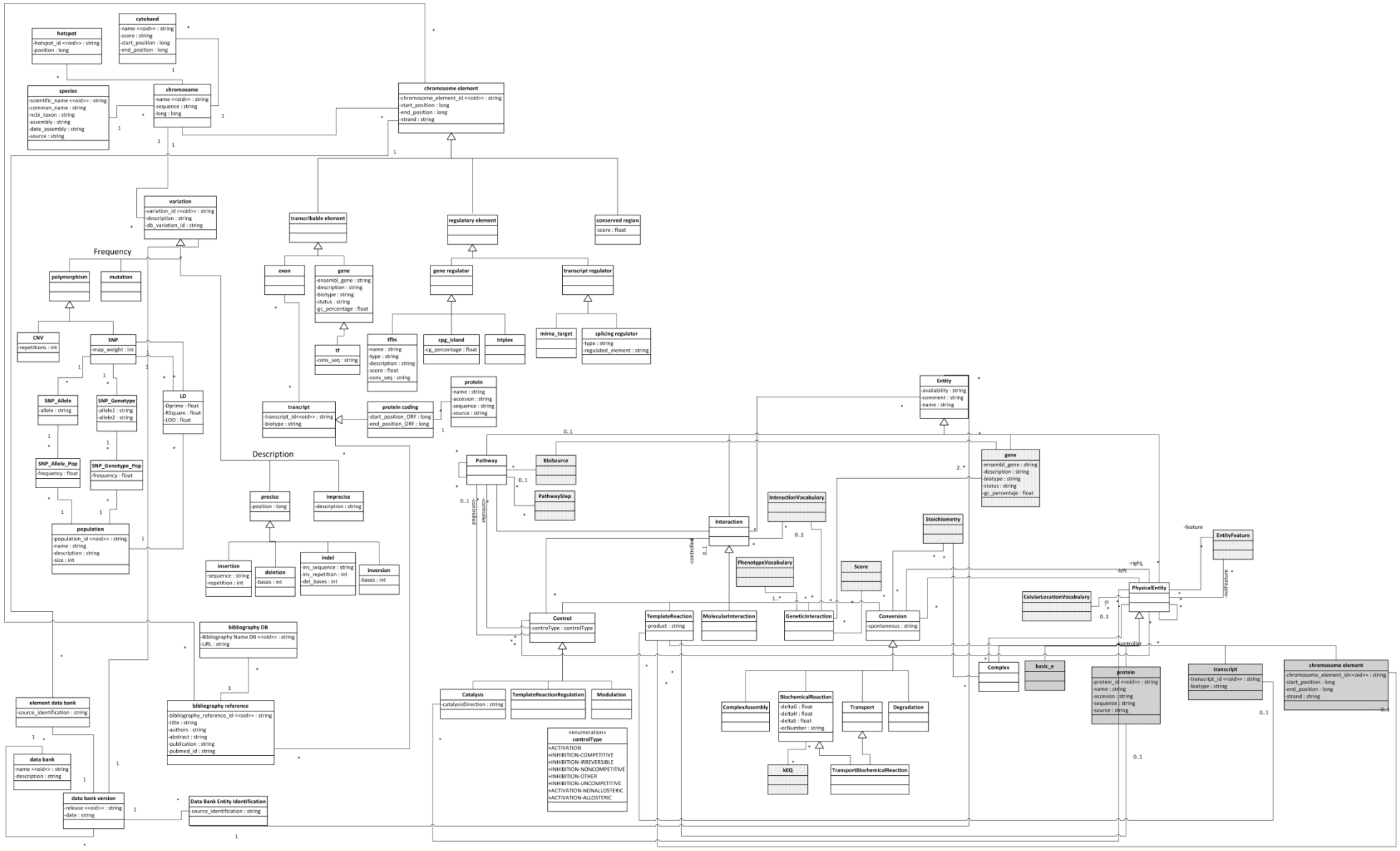


Fig. 34 Integración modelo BioPax en el modelo conceptual del genoma

La base de datos utilizada es tal y como se obtiene de la salida del programa Moskitt, pero debido a su gran cantidad de tablas, relaciones y atributos, se decide dividirla en vistas tal y como se ha hecho para definir su modelo conceptual (Figs. 35-37).

5.5.1 Vista central de BioPax

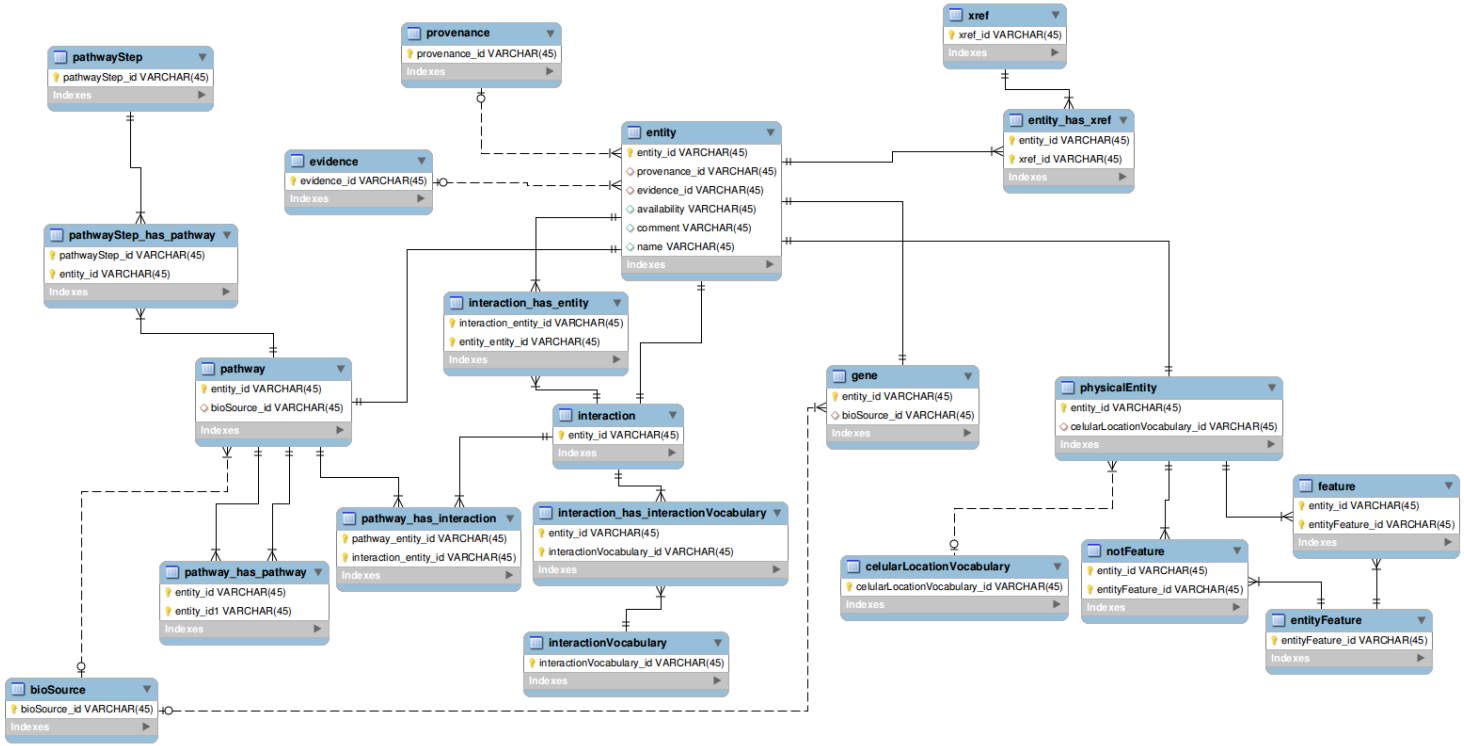


Fig. 35 Vista central de la base de datos de BioPax

5.5.2 Vista de interacción de BioPax

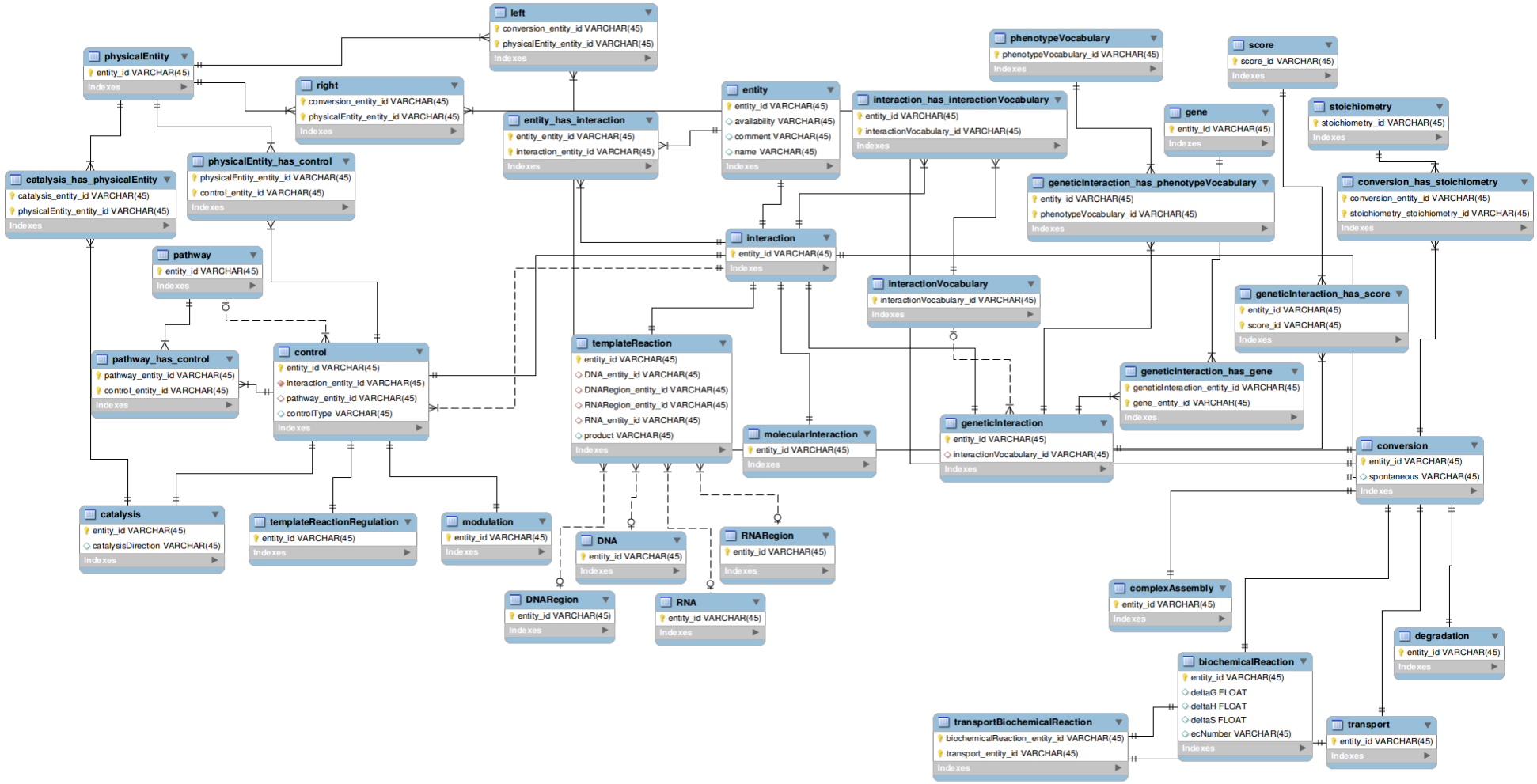


Fig. 36 Vista de interacción de la base de datos de BioPax

5.5.3 Vista de entidades físicas de BioPax

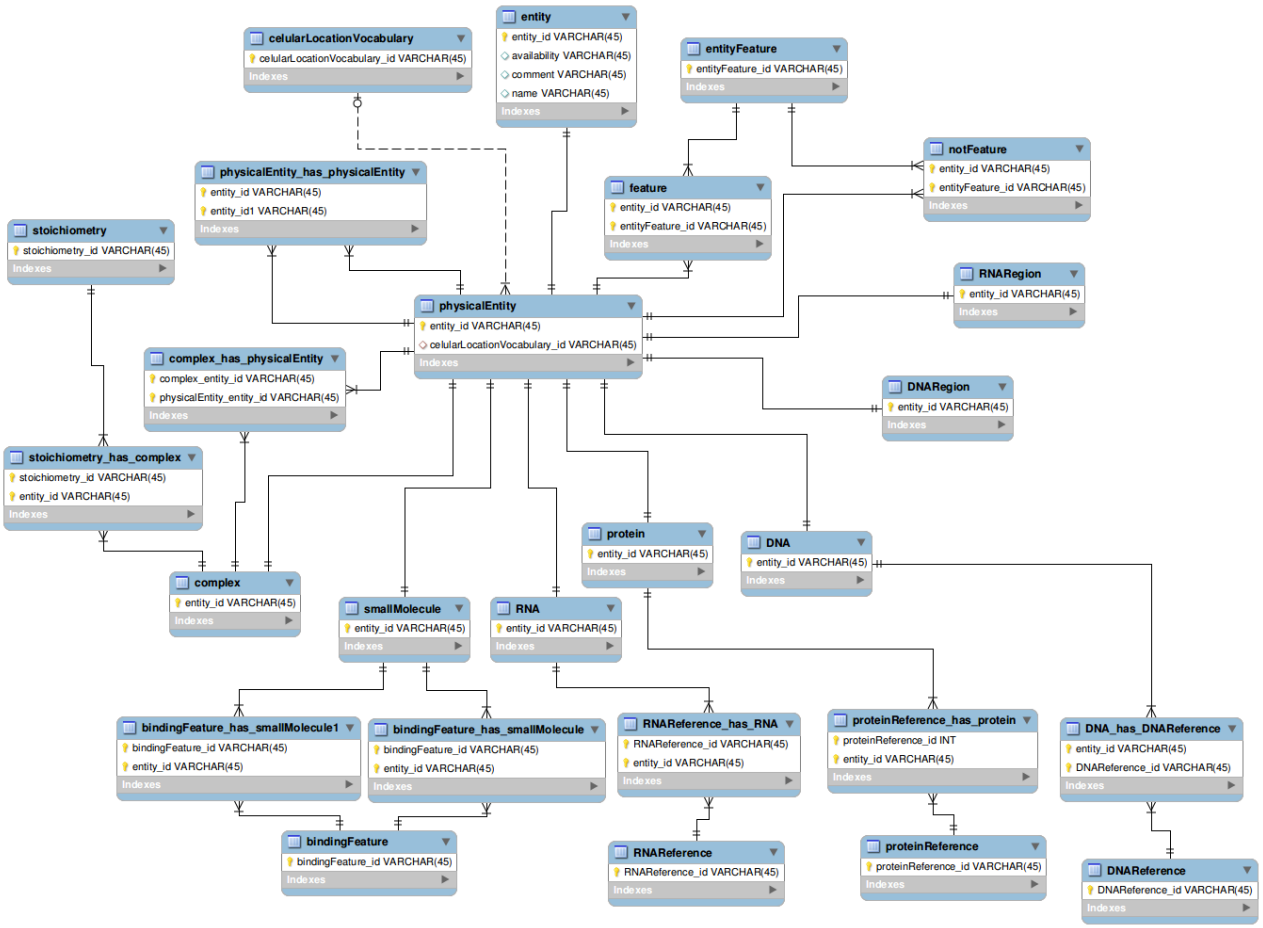


Fig. 37 Vista de entidades físicas de la base de datos de BioPax

5.6 Implementación de la base de datos de BioPax

Para comprobar si el modelo conceptual, y por tanto la base de datos, ha sido bien diseñado y cumple con los requisitos del formato de los ficheros BioPax se selecciona un conjunto de archivos de repositorios, como por ejemplo Reactome, que ofrecen sus datos en este formato con el objetivo de almacenarlos dentro. Por el mismo motivo que se ha comentado anteriormente del desarrollo 100% lineal del sistema de información de BioPax, se plantea la posibilidad de seleccionar una herramienta DDM para almacenar y extraer de manera sencilla la información de la base de datos. La herramienta seleccionada es Hibernate [42] pudiendo generar a partir del modelo los objetos java (POJOS) con los que se trabajará durante la implementación facilitando el trabajo de inserción y recuperación de datos.

El motivo por el que se han utilizado dos entornos de desarrollo, Moskitt e Hibernate es debido a que los dos se complementan para obtener el resultado deseado. En estos momentos, Moskitt ha sacado una nueva versión del producto que genera los objetos Java comentados arriba a partir del modelo conceptual. Ha día de hoy dicha herramienta hubiese sido utilizada, pero la versión inicial en la que se realizó el modelo no proporcionaba dichas características. Además, dado que el modelo conceptual fue fácilmente

exportable a eclipse, no fue un inconveniente ni una pérdida de tiempo el uso de dos entornos de desarrollo diferentes.

Debido a que los servicios web proporcionan una mejor integración e interoperabilidad entre los biólogos y sus datos, el Centro de Investigación Príncipe Felipe proporciona un servicio web que permite a los expertos acceder a la información de nuestra base de datos como si la tuviesen instalada en sus propias de trabajo. Este servicio Web también es de ayuda para los programadores ya que les permite construir complejas aplicaciones como por ejemplo herramientas de análisis sin la necesidad de instalar y mantener la base de datos. El servicio web implementado se basa en el estilo de arquitectura REST [43] y es capaz de proporcionar fácilmente resultados sobre pathways, proteínas, interacciones…

6. Conclusiones

En esta tesis se ha presentado el diseño y desarrollo de un sistema de información genómica que integra datos procedentes de las fuentes que almacenan la información más relevante en el área. En el proceso, se ha demostrado las ventajas que proporciona una aproximación metodológica basada en técnicas de modelado conceptual, al poder abordar los problemas de heterogeneidad, dispersión y desestructuración de dichas fuentes.

Las aportaciones concretas de este proyecto han sido:

- Se ha completado el diseño de un modelo conceptual que representa y unifica el conocimiento que los científicos tienen actualmente sobre el genoma. El modelado se ha abordado siguiendo una técnica de vistas que representan parcelas diferenciadas del dominio: estructura del genoma, transcripción, variaciones, rutas metabólicas y fuentes de datos y referencias bibliográficas.

- Se ha diseñado e implementado una base de datos, núcleo del sistema de información, en correspondencia con el modelo conceptual que permite la integración de las fuentes de datos genómicas más relevantes.

- Se ha desarrollado un módulo de carga, siguiendo una estrategia ETL (extracción, transformación y carga). En este sentido ha sido importante y laboriosa la definición de las correspondencias entre las fuentes y el esquema de la base de datos. El desarrollo por capas seguido permite aislar los problemas y hace que el módulo sea flexible a cambios tanto en las fuentes como en la propia base de datos.

- Se ha estudiado la propuesta de BioPax para rutas metabólicas que surge con voluntad de estandarización, para el intercambio de datos entre fuentes, y se ha integrado en el modelo conceptual diseñado en el proyecto. De esta forma se demuestra la validez de nuestro modelo al admitir la incorporación de nuevas extensiones.

Como trabajos futuros se contemplan:

- La extensión del modelo para nuevas áreas del dominio, por ejemplo información relativa a la asociación fenotipo-genotipo, información relativa a genomas reales de individuos particulares, información relativa a tratamientos conocidos, …

- La posibilidad de implementar la base de datos actual con tecnologías No SQL [44], ya que éstas están concebidas para soportar un volumen de datos mucho mayor que en el caso de las bases de datos relacionales.

El valor del trabajo realizado queda respaldado con la aceptación del artículo “Integrating human genome variation data: an information system approach” en un workshop asociado a la conferencia internacional DEXA, en su edición de 2011, y está pendiente de aceptación la release note en la revista Bioinformatics sobre la integración de la propuesta BioPax en el modelo conceptual diseñado.

Referencias bibliográficas

[1] A. Olivé, *Conceptual modeling of information systems*: Springer-Verlag New York Inc, 2007.

[2] O. Pastor and J. C. Molina, *Model-driven architecture in practice: a software production environment based on conceptual modeling*: Springer Verlag, 2007.

[3] E. D. Falkenberg, W. Hesse, P. Lindgreen, B. E. Nilsson, J. L. H. Oei, C. Rolland, R. K. Stamper, F. J. M. Van Assche, A. A. Verrijn-Stuart, and K. Voss, "A framework of information systems concepts," 1996.

[4] E. Bornberg-Bauer and N. W. Paton, "Conceptual data modelling for bioinformatics," *Briefings in bioinformatics,* vol. 3, p. 166, 2002.

[5] K. Garwood, C. Garwood, C. Hedeler, T. Griffiths, N. Swainston, S. G. Oliver, and N. W. Paton, "Model-driven user interfaces for bioinformatics data resources: regenerating the wheel as an alternative to reinventing it," *BMC bioinformatics,* vol. 7, p. 532, 2006.

[6] N. W. Paton, S. A. Khan, A. Hayes, F. Moussouni, A. Brass, K. Eilbeck, C. A. Goble, S. J. Hubbard, and S. G. Oliver, "Conceptual modelling of genomic information," *Bioinformatics,* vol. 16, p. 548, 2000.

[7] S. Ram and W. Wei, "Modeling the semantics of 3D protein structures," *Conceptual Modeling–ER 2004,* pp. 696-708, 2004.

[8] e-fungi Project, " http://www.cs.man.ac.uk/cornell/eFungi/index.html."

[9] C. Hedeler, H. M. Wong, M. J. Cornell, I. Alam, D. M. Soanes, M. Rattray, S. J. Hubbard, N. J. Talbot, S. G. Oliver, and N. W. Paton, "e-Fungi: a data resource for comparative analysis of fungal genomes," *BMC genomics,* vol. 8, p. 426, 2007.

[10] O. Pastor, "Conceptual Modeling Meets the Human Genome," *Conceptual Modeling-ER 2008,* pp. 1-11, 2008.

[11] O. Pastor, van der Kroon,M., Levin,A., Casamayor,J.C., Celma,M., "A Conceptual Modeling Approach to Improve Human Genome Understanding. ," *Handbook of Conceptual Modeling: Theory, Practice and Research Challenges. In Embley,D., Thalheim,B. (Eds.), Springer,* pp. 517-541, 2011.

[12] O. Pastor, A. Levin, M. Celma, J. Casamayor, A. Virrueta, and L. Eraso, "Model-Based Engineering Applied to the Interpretation of the Human Genome," *The Evolution of Conceptual Modeling,* pp. 306-330.

[13] M. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, and C. Mungall, "The Gene Ontology (GO) database and informatics resource," *Nucleic acids research,* vol. 32, p. D258, 2004.

[14] E. Demir, M. P. Cary, S. Paley, K. Fukuda, C. Lemer, I. Vastrik, G. Wu, P. D'Eustachio, C. Schaefer, and J. Luciano, "The BioPAX community standard for pathway data sharing," *Nature biotechnology,* vol. 28, pp. 935-942.

[15] H. Hermjakob, L. Montecchi-Palazzi, G. Bader, J. Wojcik, L. Salwinski, A. Ceol, S. Moore, S. Orchard, U. Sarkans, and C. Von Mering, "The HUPO PSI's molecular interaction format—a community standard for the representation of protein interaction data," *Nature biotechnology,* vol. 22, pp. 177-183, 2004.

[16] A. Cornish-Bowden, P. Hunter, A. Cuellar, E. Mjolsness, N. Juty, S. Dronov, K. Takahashi, Y. Nakayama, E. Gilles, and J. Kasberger, "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics,* vol. 19, pp. 524-531, 2003.

[17] A. A. Cuellar, C. M. Lloyd, P. F. Nielsen, D. P. Bullivant, D. P. Nickerson, and P. J. Hunter, "An overview of CellML 1.1, a biological model description language," *Simulation,* vol. 79, p. 740, 2003.

[18] N. Le Novère, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M. I. Aladjem, and S. M. Wimalaratne, "The systems biology graphical notation," *Nature biotechnology,* vol. 27, pp. 735-741, 2009.

[19] C. Tao and D. Embley, "Seed-based generation of personalized bio-ontologies for information extraction," *Advances in Conceptual Modeling–Foundations and Applications,* pp. 74-84, 2007.

[20] P. Flicek, M. R. Amode, D. Barrell, K. Beal, S. Brent, Y. Chen, P. Clapham, G. Coates, S. Fairley, and S. Fitzgerald, "Ensembl 2011," *Nucleic acids research,* vol. 39, p. D800.
[21] R. A. Gibbs, J. W. Belmont, P. Hardenbol, T. D. Willis, F. Yu, H. Yang, L. Y. Ch'ang, W. Huang, B. Liu, and Y. Shen, "The international HapMap project," *Nature,* vol. 426, pp. 789-796, 2003.
[22] G. A. Thorisson, A. V. Smith, L. Krishnan, and L. D. Stein, "The international HapMap project web site," *Genome research,* vol. 15, p. 1592, 2005.
[23] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, and M. Magrane, "The universal protein resource (UniProt)," *Nucleic acids research,* vol. 33, p. D154, 2005.
[24] E. Jain, A. Bairoch, S. Duvaud, I. Phan, N. Redaschi, B. E. Suzek, M. J. Martin, P. McGarvey, and E. Gasteiger, "Infrastructure for the life sciences: design and implementation of the UniProt website," *BMC bioinformatics,* vol. 10, p. 136, 2009.
[25] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, and W. FitzHugh, "Initial sequencing and analysis of the human genome," *Nature,* vol. 409, pp. 860-921, 2001.
[26] M. B. Gerstein, C. Bruce, J. S. Rozowsky, D. Zheng, J. Du, J. O. Korbel, O. Emanuelsson, Z. D. Zhang, S. Weissman, and M. Snyder, "What is a gene, post-ENCODE? History and updated definition," *Genome research,* vol. 17, p. 669, 2007.
[27] B. Modrek and C. Lee, "A genomic view of alternative splicing," *Nature genetics,* vol. 30, pp. 13-19, 2002.
[28] J. Den Dunnen and S. Antonarakis, "Nomenclature for the description of human sequence variations," *Human genetics,* vol. 109, pp. 121-124, 2001.
[29] R. Richesson and J. P. Turley, "Conceptual models: Definitions, construction, and applications in public health surveillance," *Journal of Urban Health,* vol. 80, pp. 128-128, 2003.
[30] O. Pastor, A. M. Levin, J. C. Casamayor, M. Celma, L. E. Eraso, M. J. Villanueva, and M. Perez-Alonso, "Enforcing conceptual modeling to improve the understanding of human genome," pp. 85-92.
[31] K. Paigen and P. Petkov, "Mammalian recombination hot spots: properties, control and evolution," *Nature Reviews Genetics,* vol. 11, pp. 221-233.
[32] J. T. Den Dunnen and S. E. Antonarakis, "Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion," *Human Mutation,* vol. 15, pp. 7-12, 2000.
[33] S. Bamford, E. Dawson, S. Forbes, J. Clements, R. Pettett, A. Dogan, A. Flanagan, J. Teague, P. Futreal, and M. Stratton, "The COSMIC (Catalogue of Somatic Mutations in Cancer) database and website," *British journal of cancer,* vol. 91, pp. 355-358, 2004.
[34] A. Sandelin, W. Alkema, P. Engström, W. W. Wasserman, and B. Lenhard, "JASPAR: an open access database for eukaryotic transcription factor binding profiles," *Nucleic acids research,* vol. 32, p. D91, 2004.
[35] P. A. Fujita, B. Rhead, A. S. Zweig, A. S. Hinrichs, D. Karolchik, M. S. Cline, M. Goldman, G. P. Barber, H. Clawson, and A. Coelho, "The UCSC Genome Browser database: update 2011," *Nucleic acids research,* vol. 39, p. D876.
[36] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, and A. M. Zahler, "The human genome browser at UCSC," *Genome research,* vol. 12, p. 996, 2002.
[37] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, and L. Matthews, "Reactome: a knowledgebase of biological pathways," *Nucleic acids research,* vol. 33, p. D428, 2005.
[38] R. Caspi, H. Foerster, C. A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Y. Rhee, A. G. Shearer, and C. Tissier, "The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases," *Nucleic acids research,* vol. 36, p. D623, 2008.
[39] G. D. Bader, D. Betel, and C. W. V. Hogue, "BIND: the biomolecular interaction network database," *Nucleic acids research,* vol. 31, p. 248, 2003.
[40] B. Selic, "The pragmatics of model-driven development," *Software, IEEE,* vol. 20, pp. 19-25, 2003.
[41] Moskkitt, "www.moskitt.org."
[42] C. Bauer and G. King, *Hibernate in action*: Manning, 2005.

[43] R. L. Costello, "Building web services the rest way," *URL: http://www.xfront.com/REST-Web-Services. html. V*ol. 11, p. 2007, 2007.

[44] M. Stonebraker, "SQL databases v. NoSQL databases," *Communications of the ACM,* vol. 53, pp. 10-11.

Anexo I: Lista de figuras

Anexo II: Lista de tablas

Anexo III: Lista de términos