Esquema Conceptual del Genoma Humano: una herramienta para la integración y gestión de su información

Aremy Olaya Virrueta Gordillo

Supervisores:

Oscar Pastor López
Matilde Celma Giménez

Centro de Investigación en Métodos de Producción de Software

UNIVERSIDAD POLITECNICA DE VALENCIA

UNIVERSIDAD POLITECNICA DE VALENCIA

Master en Ingeniería de Software, Métodos Formales y Sistemas de Información.

Esquema Conceptual del Genoma Humano: una herramienta para la integración y gestión de su información.

Tesina presentada por:
Aremy Olaya Virrueta Gordillo.

Supervisores:
Oscar Pastor López.
Matilde Celma Giménez.

Valencia, España. Septiembre 2009.

A Dios por bendecir mi vida.

A mis padres porque todo lo que soy es gracias a ellos.

A mis hermanos por ser mis compañeros incondicionales.

Resumen

La secuenciación del genoma humano ha sido una aportación muy importante para la ciencia y es considerado el punto de partida para muchos otros estudios científicos referentes a la biología humana. Los resultados de estos estudios constituyen información muy valiosa, por lo que actualmente se publican en la web con la finalidad de que estos datos puedan ser buscados y recuperados.

Sin embargo, la publicación de datos biológicos no está estandarizada y realizar procesos de búsqueda y recuperación de información en la profundidad de la web son actividades que implican una cantidad significativa de tiempo. Este escenario hace evidente la necesidad de diseñar y desarrollar herramientas informáticas que sirvan de soporte para la integración y gestión de información biológica.

En el presente trabajo de investigación se aborda al Genoma Humano como un dominio de un Sistema de Información. Para formalizar la propuesta se constituye una base de conocimientos con todos los elementos y procesos relevantes en este dominio.

Posteriormente se diseña un esquema conceptual siguiendo la perspectiva MDE (Model Driven Engineering). Esto permite aplicar técnicas de Modelado Conceptual para especificar el Genoma Humado y posibilita el intercambio de conocimientos entre biólogos e informáticos a un nivel más alto de abstracción; lo cual permite la adquisición, identificación y evaluación de conceptos importantes presentes en el dominio.

Especificar el Genoma Humano en un esquema conceptual es una tarea que involucra desafíos conceptuales interesantes. Por lo que en esta tesina se cuenta la experiencia de diseñar un esquema conceptual para éste dominio poco convencional considerando la evolución que ha experimentado el esquema para poseer las propiedades de corrección y completitud que distinguen a los esquemas conceptuales de Sistemas de Información de dominios comunes.

Finalmente se describe el proceso de creación de una base de datos a partir del esquema conceptual del Genoma Humano, considerándola una contribución importante para la demanda del diseño y desarrollo de aplicaciones software que permitan utilizar los datos biológicos publicados.

Agradecimientos

En primer lugar expreso mi gratitud al Sistema Nacional de Educación Superior Tecnológica (SNEST) de México, por su apoyo en la realización de mis estudios de posgrado. De igual forma agradezco a mis supervisores Dra. Matilde Celma Giménez y Dr. Oscar Pastor López por sus valiosas aportaciones y el tiempo dedicado al desarrollo de este trabajo de investigación.

También hago un agradecimiento especial al Centro de Investigaciones en Producción de Software (ProS), en especial a todos mis compañeros del grupo “genoma” por las enriquecedoras charlas y por compartir su conocimiento.

Finalmente agradezco a todas las demás personas que han compartido, de una u otra forma, el proceso de elaboración de esta tesina. Especialmente a Caro y a Leo por su cariño y apoyo incondicional. A Vlad por estar siempre conmigo y acompañarme en todo.

Índice de Contenidos

Índice de Figuras

1 Introducción

La evolución de los paradigmas de diseño y desarrollo de sistemas de información en los que se fundamenta la Ingeniería de Software han permitido que hoy en día el proceso de obtención de sistemas de calidad implique la aplicación de técnicas de Modelado Conceptual, de tal forma que la obtención de modelos a diferentes niveles de abstracción determine el producto software final.

El Modelado Conceptual marca el inicio de la Ingeniería de Software moderna: el entendimiento del dominio del problema y la conceptualización del conocimiento que se tiene sobre él, a un nivel abstracto antes de implementar una solución, permite que los ingenieros de software y los clientes de los sistemas trabajen al mismo nivel y que, además, exista un entendimiento sobre lo que se tiene que hacer y el producto que se obtendrá. Es importante mencionar que en este contexto, aplicar técnicas de gestión de modelos conceptuales permite la automatización de varios procesos involucrados en la producción del sistema. Ejemplos de ellos son: la generación de código, la especificación de requisitos, estimaciones de costos, especificación de recursos, etc.

El diseño de modelos conceptuales, con diferentes niveles de abstracción, es abordado a través de la Ingeniería Dirigida por Modelos MDE (Model-Driven Engineering). MDE es una metodología de desarrollo de software que se enfoca en la creación de modelos o abstracciones de conceptos en un dominio particular. Estos conceptos están definidos con una aproximación mayor hacia el dominio en lugar de estar más enfocados hacia los conceptos (o algoritmos) computacionales. Esto significa un incremento de la productividad maximizando la compatibilidad entre sistemas, simplificando el proceso de diseño y permitiendo la comunicación entre los elementos del sistema.

Por lo tanto, la idea principal de MDE consiste en dirigir el desarrollo de software a partir de la definición de modelos del dominio del problema y dar soporte al procesamiento de los diferentes modelos definidos. Este escenario permite observar que los artefactos software desarrollados concuerdan con lo que el cliente desea. Actualmente una de las principales propuestas en la Ingeniería Dirigida por Modelos es el estándar MDA (Model-Driven Architecture) [11], que se define como una arquitectura de MDE sustentada por el consorcio OMG (Object Management Group)

[12]. Usando la metodología MDA, la funcionalidad del sistema se define con un modelo independiente de la plataforma, en la que el artefacto software se implementará, a través de un lenguaje específico para el dominio del que se trate [5].

Los éxitos obtenidos en la producción de sistemas de información, diseñados y desarrollados usando la metodología MDA, conduce a pensar en el beneficio que la aplicación del Modelado Conceptual puede aportar a la situación actual del desarrollo de aplicaciones software utilizadas en la gestión de datos biológicos heterogéneos.

1.1 *Antecedentes*

La realización de estudios científicos en Biología Molecular y Genética involucra la aplicación de procedimientos y protocolos determinados para la extracción de ADN a partir de muestras de sangre de sujetos de estudio y el tratamiento del ADN extraído para detectar la expresión genética correspondiente.

El volumen de los datos adquiridos de la extracción de ADN y los datos generados a partir de los experimentos conlleva a los biólogos al uso de herramientas informáticas para el tratamiento de esta información.

Anteriormente los datos eran almacenados en hojas de cálculo y analizados a partir de paquetes estadísticos. Por ello los complejos protocolos de biología molecular no tenían el soporte informático adecuado para la gestión de los datos biomédicos.

Con el objetivo de suplir la necesidad de tratamiento eficiente de información que demandaban los procedimientos y protocolos en Biología Molecular, se empezaron a desarrollar herramientas informáticas, principalmente alrededor del Proyecto del Genoma Humano [1] que debían manejar y analizar la enorme cantidad de datos que se generaban diariamente. Sin embargo, las herramientas informáticas desarrolladas hasta ese momento no abordaban de manera eficiente la automatización de procesos más formales de la Genética que contribuyen a la secuenciación completa del genoma humano.

La necesidad de incluir herramientas informáticas para apoyar el trabajo de los biólogos da lugar a la Biología Computacional, como una disciplina que incorpora métodos, técnicas y herramientas de la Ciencia de la Computación para mejorar la capacidad de la investigación tradicional; también comienza a hablarse de Genética Computacional como una especialización de los estudios en Biología. Posteriormente estas aproximaciones han podido ser abordadas a partir de lo que hoy en día se conoce como *Biología Sistémica*, el estudio del comportamiento de la estructura y de los procesos biológicos complejos en términos de constituyentes moleculares [2]; *Bioinformática*, una disciplina académica y un área de investigación que se define como la aplicación de técnicas computacionales a la gestión de la información biológica y el uso de métodos derivados de disciplinas tales como la Matemática Aplicada, la Algorítmica, la Estadística, Ciencias de la Salud, etc. [3]; o *Informática Biomédica*, una disciplina que pretende establecer los fundamentos teóricos, las metodologías y las técnicas, y diseñar procedimientos e instrumentos, que permitan integrar la información gestionada en los distintos niveles de información sobre salud en un sistema conceptualmente homogéneo [4].

La definición de aproximaciones dentro de la Genética Computacional conlleva a la idea de que los principios y procedimientos utilizados en las Ciencias de la Computación pueden ser explotados para apoyar la gestión de información genética. Para lograr que los beneficios otorgados por las ciencias de la computación a otras áreas puedan trasladarse a la Biología, es necesario realizar trabajos bioinformáticos de calidad que generen documentos de investigación que marquen los principios bajo los que se desarrollarán múltiples aplicaciones computacionales. El trabajo de investigación presentado en esta tesina es un trabajo de Bioinformática y es el producto de un proceso de estudio muy detallado sobre el genoma humano y la aplicación de principios de las ciencias de la computación, específicamente del Modelado Conceptual.

Originalmente, los resultados de los análisis genéticos eran almacenados en papel y estaban disponibles solo para el laboratorio que había realizado el experimento de secuenciación. Con la aparición de los ordenadores modernos y la posibilidad de almacenamiento digital, los biólogos comienzan a almacenar los resultados de sus análisis en hojas de cálculo y posteriormente en bases de datos. No obstante, estas secuencias de ADN continuaban siendo útiles únicamente para el laboratorio que las

obtenía. Con la introducción y extensión de Internet, los biólogos empiezan a publicar los resultados de sus experimentos en la Web y es cuando estos datos comienzan a utilizarse como referencias para investigaciones realizadas en otros laboratorios.

De esta forma, la proliferación de repositorios de datos biológicos ha provocado el desarrollo de sistemas software que permitan a los científicos navegar, buscar y analizar los datos que estos contienen.

Generalmente los datos biológicos de estos repositorios son accedidos por medio de páginas Web, pero también suelen ser accedidos por medio de aplicaciones de escritorio y herramientas de líneas de comandos.

Actualmente los análisis genéticos han incluido un nuevo paso en su protocolo: la búsqueda de referencias bibliográficas. Así los pasos a seguir en cualquier análisis genético actual son: secuenciación – análisis – búsqueda de bibliografía – conclusión/diagnóstico [4]. Esta nueva fase, en la que se realiza una búsqueda de referencias bibliográficas sobre las secuencias de nucleótidos que otros biólogos han determinado previamente, ha sido incluida debido a que los datos ya no provienen solamente de los resultados que se obtienen en un laboratorio, sino que pueden provenir de la Web.

No obstante, la búsqueda de bibliografía digital, asociada a secuenciaciones de genomas de seres humanos, es una tarea que implica mucho tiempo. Esto se debe a que no existe una forma estandarizada para publicar las secuencias en los diferentes sitios Web y la búsqueda junto con la recuperación de información se convierten en tareas difíciles. Los datos publicados tienen estructuras diversas y el formato en el que se presentan son diferentes para cada sitio. Por ejemplo, para algunos casos las secuencias se representan como texto o bien como resultado de una consulta a una base de datos. Sin embargo, toda la información es valiosa y debe ser buscada, recuperada y procesada de manera eficiente para ser utilizada como referencia bibliográfica.

Hoy en día se invierte mucho tiempo en la realización de un análisis genético, a pesar de que los procesos para secuenciar el ADN y para el manejo de las secuencias han experimentado un avance tecnológico significativo. Sin embargo, es la parte de

búsqueda bibliográfica para referenciar el análisis genético, en la que se consume demasiado tiempo.

Aunque existen muchos artefactos software para acceder a las secuencias de nucleótidos publicadas, la tarea de los biólogos para buscar las referencias bibliográficas que respalden sus determinaciones genéticas continúa siendo complicada, debido sobre todo a que los artefactos software son generalmente hechos a la medida y los volúmenes de datos biológicos son muy grandes, además de que son mayormente heterogéneos.

En principio, para abordar la complejidad de esta tarea, biólogos e informáticos unen esfuerzos y generan soluciones software que han sido bastante útiles. Sin duda la más significativa ha sido la fundación del NCBI (*National Center for Biotechnology Information*) [6]**.** Este centro, a través de un sitio web, agrupa la información biomédica que está disponible en Internet. Aún así, esa web tiene una capacidad limitada para encontrar toda la información requerida, por lo que también se utilizan otros portales webs con su correspondiente banco de datos, como son HGMD (*Human Gene Mutation Database*) [7], HUGO (*Gene Nomenclature Committee*) [8] y muchas otras más. Actualmente hay más de 1000 bases de datos disponibles en la web, relacionadas con el dominio biológico y la cantidad aumenta día a día considerablemente [9].

Sin embargo, el objetivo común, con el que se crean estos artefactos software, es asegurar que existan herramientas informáticas apropiadas, sistemas de información y bases de datos disponibles para manejar y explotar información biológica [10].

1.2 *Motivación*

La Ingeniería Dirigida por Modelos se ha aplicado a numerosos dominios, la mayoría orientados a aplicaciones convencionales de gestión o a sistemas organizacionales. Sin embargo, existe un dominio específico que no se ha beneficiado de las ventajas que proporciona el Modelado Conceptual; este dominio es el Genoma Humano. Los trabajos realizados en este ámbito se centran en el Espacio de la Solución y se orientan a la búsqueda de patrones sobre cantidades ingentes de información genómica. Por lo tanto, la integración y gestión de datos biológicos son tópicos importantes que demandan ser investigados de manera rigurosa.

La Biología Molecular, y en consecuencia el genoma humano, es un dominio complejo para los ingenieros de software. No obstante, para los biólogos los temas de ingeniería de software también representan complejidad. Sin embargo, biólogos e ingenieros de software trabajan en conjunto para desarrollar aplicaciones software que gestionen información genómica y que den soporte a la automatización de procesos involucrados en temas de Biología Molecular.

La gestión de datos biológicos ha sido abordada desde un escenario específico, es decir, para cada proceso es desarrollada una implementación software a la medida. Al mismo tiempo, esta aplicación gestiona únicamente los datos que necesita el proceso determinado; este escenario ilustra el hecho de que los principios del Modelado Conceptual no han sido explotados totalmente.

Por lo tanto, aplicar técnicas de Modelado Conceptual permite una aproximación eficiente en un escenario general para la automatización de varios procesos. Esta aproximación es tratada para una misma implementación software. Además, la gestión e integración de datos biológicos es abordada a través de la tecnología de modelos lo que permite tomar ventaja de los beneficios que esta tecnología ha proporcionado a otros Sistemas de Información.

Sin embargo, la información biológica que ha de gestionarse en los Sistemas de Información debe ser delimitada debido al gran volumen de información disponible. Esta delimitación implica la especificación de un dominio. Para este trabajo, se considera información acerca del genoma humano. Por lo tanto, es necesario agrupar conocimientos sobre la composición y funcionamiento del genoma humano y utilizar todas las ventajas que proporciona el Modelado Conceptual, para lograr que los excelentes resultados obtenidos en los dominios en los que se ha aplicado la ingeniería dirigida por modelos, se obtengan también para los estudios científicos y aplicaciones médicas basadas en la interpretación del genoma humano.

El problema de recuperar y procesar la información sobre datos biológicos para que éstos sean útiles para cualquier proceso científico en el que se vean implicados temas sobre el procesamiento de información del genoma humano, debe ser abordado a un nivel más abstracto de tal forma que la solución pueda ser comprendida tanto por

biólogos como por ingenieros de software. Esta solución deberá funcionar como base para diseñar y desarrollar sistemas de información que apoyen la investigación científica sobre el genoma humano.

Considerando las ventajas que el Modelado Conceptual aporta a la calidad en los Sistemas de Información, y la necesidad que tiene el estudio del genoma humano de disponer de Sistemas de Información capaces de resolver los grandes avances que actualmente se producen en Biología Molecular. Se propone diseñar un esquema conceptual para la especificación del genoma humano. Este esquema puede ser visto como la base para generar diferentes Sistemas de Información que sean capaces de dar soporte a la proyección y a la calidad que actualmente demanda el procesamiento de información referente a la secuenciación del genoma de los seres humanos.

1.3 *Objetivos del trabajo*

Objetivo General

- Diseñar un esquema conceptual del genoma humano que sea lo suficientemente completo para representar su estructura y su comportamiento.

Objetivos Específicos

- Realizar una revisión de la ingeniería dirigida por modelos MDD y del estándar MDA.
- Adquirir los conocimientos biológicos necesarios que permitan el análisis del dominio del problema.
- Identificar los conceptos del dominio que se consideran relevantes para que el esquema conceptual describa con precisión la estructura del genoma y su funcionamiento.
- Diseñar el esquema conceptual para la interpretación del genoma humano siguiendo una aproximación por vistas con la finalidad de diferenciar la estructura y el funcionamiento.
- Implementar la base de datos del esquema conceptual diseñado.

- Realizar una evaluación de la información pública disponible actualmente. Con la finalidad de identificar los datos publicados y la forma en que pueden almacenarse en la base de datos.

1.4 *Estructura de la tesina*

La estructura del trabajo de investigación presentado en esta tesina es la siguiente: En principio se hace una introducción sobre el Modelado Conceptual en la evolución de los paradigmas de diseño y desarrollo de sistemas y la motivación que lleva a aplicar técnicas de Modelado Conceptual a temas de Biología Molecular. Se describe la problemática, producto del trabajo de biólogos e informáticos, en el desarrollo y uso de artefactos software; y se enuncia una solución que muestra la utilidad de fusionar modelado conceptual y la especificación del genoma humano para que este pueda ser interpretado tanto por biólogos como por informáticos. También se dan a conocer los objetivos del trabajo de investigación.

En el capitulo 2 se incluyen los fundamentos teóricos sobre Modelado Conceptual y Biología Molecular sobre los que se sustenta la solución propuesta. Se presenta una descripción detallada sobre el dominio abordado a partir de la idea de modelar conceptualmente el genoma humano. Posteriormente, se realiza un estudio del estado del arte sobre la aplicación del modelado conceptual en temas de biología molecular, particularmente el genoma humano.

En el capítulo 3 se describe con detalle el trabajo realizado para aplicar Modelado Conceptual en la especificación del genoma humano. Se propone un esquema conceptual y posteriormente se explica como este ha evolucionado a partir de la obtención del nuevo conocimiento sobre el dominio. Al mismo tiempo se cuenta como este esquema conceptual debe ser tratado desde dos perspectivas: 1) Un esquema conceptual ideal (ECI), que se corresponde con la especificación del genoma humano a partir de los conceptos involucrados en su constitución y funcionamiento; y 2) Un esquema conceptual real (ECR), que es una adecuación del ECI a partir de la información que está disponible actualmente. Esta adecuación se realiza con la intención de crear un SI operativo que interactúe con la información disponible actualmente.

En el capitulo 4 se muestra una aplicación de este modelo conceptual para la creación de una base de datos. Finalmente, se detallan las conclusiones, aportaciones personales, publicaciones realizadas y líneas de trabajo futuras.

2 Fundamentos teóricos: Modelado Conceptual y Biología Molecular

Resumen: En este capitulo se enuncian fundamentos teóricos sobre el modelado conceptual. Su intención es proporcionar un panorama general sobre elementos y técnicas que son utilizados en el diseño de un esquema conceptual que representa un dominio. Principalmente se describe la técnica de diagramas de clases, ya que el esquema conceptual presentado en esta memoria ha sido diseñado con esta técnica. Así mismo, en la segunda sección del capítulo se incluyen fundamentos teóricos sobre Biología Molecular haciendo énfasis en una descripción global que abarca desde las células y los genomas hasta la especificación de elementos y mecanismos celulares propios del genoma humano. Este escenario tiene una correspondencia directa con la descripción del dominio del problema para el que se ha diseñado el esquema conceptual presentado. Finalmente, en la tercera sección del capitulo se presenta un estado del arte sobre técnicas de Modelado Conceptual y temas de Biología Molecular.

2.1 *Modelado Conceptual*

El Modelado Conceptual, es la actividad que tiene como objetivo obtener y definir conocimiento sobre un sistema. En este contexto, debe entenderse por sistema un conjunto de elementos relacionados entre sí y con su entorno. Los sistemas pueden ser naturales, filosóficos, económicos, de información, etc.

Modelar conceptualmente un sistema implica la especificación de un dominio, es decir, la identificación de las propiedades relevantes del sistema, en un instante determinado, así como su comportamiento. Estas propiedades relevantes son una abstracción de la realidad según el punto de vista del observador.

Cada sistema tiene una serie de objetivos, que deben ser tomados en cuenta cuando se realiza la descripción de sus características relevantes. Los sistemas de información (SI) son sistemas que contribuyen a que otros sistemas (más amplios) cumplan sus objetivos [13].

Esta perspectiva es esencial ya que los SI se describen en un entorno de ingeniería de software que posteriormente se convierten en implementaciones que ayudan a sistemas de cualquier tipo a cumplir sus objetivos a través de artefactos software.

Puede decirse que un SI es el sistema que recoge, almacena, procesa y distribuye información sobre las propiedades relevantes de un sistema y su entorno (dominio), con la intención de que el sistema en cuestión cumpla sus objetivos [15].

Por lo tanto, el Modelado Conceptual en la constitución de un SI es la actividad que tiene como objetivo obtener y definir el conocimiento que el SI necesita sobre su dominio.

A continuación se detallarán fundamentos sobre el modelado conceptual con la finalidad de mostrar porque el genoma humano puede ser visto como un SI y toda la información asociada a su dominio puede ser descrita en un modelo conceptual.

La intención de este capítulo es que las personas relacionadas con el conocimiento del dominio: genoma humano, conozcan las bases sobre las que se construyó el esquema conceptual del genoma humano. Esto logra que biólogos e informáticos hablen un lenguaje común.

2.1.1 Fundamentos de modelado conceptual

En el campo de los SI se supone que un dominio consta de un conjunto de objetos y de relaciones entre objetos que se clasifican en conceptos. Esto se denomina compromiso ontológico [14].

La denominación de compromiso ontológico, se hace a partir de que la Ontología es la rama de la Filosofía que estudia la naturaleza y organización de la realidad. Por lo tanto, un compromiso ontológico es una manera concreta de observar la realidad.

Un modelo conceptual es un compromiso ontológico, ya que es una forma de observar un dominio determinado. Que puede definirse como un conjunto de conceptos y de

reglas destinados a representar de forma global los aspectos lógicos de la información existente en la realidad que está siendo analizada [16].

Para representar la información del dominio, el modelo conceptual, emplea un lenguaje de modelado. Dentro de un modelo pueden presentarse varios lenguajes diferentes de modelado asociados pero todos ellos serán equivalentes.

La conceptualización del dominio se realiza a partir de identificar los conceptos dentro de la información del dominio del sistema. La definición de conceptos es una de las actividades más importantes en el modelado conceptual. Posteriormente esta conceptualización se especifica en un esquema conceptual al describir con un lenguaje de modelado los conceptos del dominio [18].

2.1.2 Elementos conceptuales

Un modelo conceptual explica cuales son y como se relacionan los conceptos que son relevantes en la descripción de la información del dominio de un sistema.

Los modelos conceptuales se basan en la relación concepto- instancia. Un concepto es una idea formada generalizando las propiedades relevantes que tienen en común un conjunto de instancias; una instancia es la una ocurrencia de un concepto determinado con valores específicos para las propiedades relevantes.

La relación concepto--instancia, también puede ser vista como clase conceptual--objeto conceptual, sin confundir esta relación con su significado en programación orientada a objetos.

Una clase conceptual denota un conjunto de objetos conceptuales que comparten características comunes. Estas características pueden ser atributos o relaciones. Un objeto conceptual denota una entidad o concepto del dominio.

Un atributo en una clase conceptual es una característica intrínseca de un objeto, es independiente de otros objetos. Tiene un nombre y un rango posible de valores que pueden ser mutables o inmutables.

Una relación es una característica que vincula conceptualmente a varias clases conceptuales distintas. Cada clase conceptual u objeto conceptual juega un rol conceptual en ese vínculo.

Los dominios de los SI, que se describen en modelos conceptuales, tienden a varias su base de información con el tiempo. Por lo que la conceptualización que se haga del dominio en un modelo conceptual debe ser flexible para insertar o borrar clases de conceptos.

Dos de las propiedades más importantes que debe poseer un modelo conceptual son la completitud y la corrección. Esto con la finalidad de garantizar que la descripción que se hace en el modelo conceptual del dominio, es una representación válida en cualquier momento.

La completitud se refiere a que la base de información del SI debe contener todos los conceptos relevantes del dominio. La corrección se refiere a que todos los conceptos almacenados en la base de información del SI deben ser correctos.

Los modelos conceptuales incluyen restricciones de integridad en sus descripciones del dominio para garantizar su completitud y su corrección. Una restricción de integridad es una condición que la base de información debe satisfacer para garantizar que es una representación válida del dominio. La base de información es consistente si satisface todas sus restricciones de integridad.

2.1.3 Técnicas de modelado conceptual

Existen muchas técnicas de modelado, con distintos grados de sofisticación, para describir el modelo conceptual.

- Diccionario/Glosario: Es una lista de clases de con sus atributos y relaciones. Tiene poca estructura y es difícil de analizar.

- Diagrama de Entidad Relación: Es un lenguaje gráfico que introduce estructura para las clases de conceptos. Es utilizado para el diseño de bases de datos.

- Diagrama de Clases: Es una extensión del diagrama Entidad Relación que añade características expresivas como herencia, modificadores, etc.

En la realización de este trabajo de investigación para construir un esquema conceptual del genoma humano, se ha utilizado la técnica de diagramas de clases. Por lo que se describe a continuación.

2.1.4 Diagramas de clases

Un diagrama de clases es un tipo de diagrama que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro [17].

En un diagrama de clases se identifican los siguientes elementos.

- *Clases*: una clase es la agrupación de un conjunto de propiedades y características comunes que presentan elementos del dominio.
- *Atributos*: también llamados propiedades o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación.
- *Asociaciones*: son las diferentes relaciones que existen entre las instancias de las clases que definen como los elementos del dominio se interrelacionan.
- *Herencia*: se define como la asociación de una clase padre (superclase) ya definida para poder extender la funcionalidad en una clase hija (subclase). Las clases hijas heredan todas las propiedades de una clase madre.

Al diseñar una clase (Fig.1) se debe pensar en cómo se puede identificar un objeto real, como una persona, un transporte, un documento o un paquete. Estos ejemplos de clases de objetos reales, es sobre lo que un sistema se diseña. Durante el proceso del diseño de

las clases se toman las propiedades que identifican como único al objeto y otras propiedades adicionales como datos que corresponden al objeto.

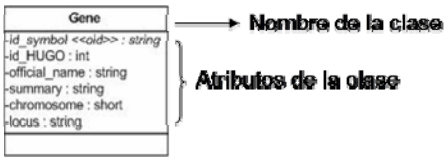


Fig. 1 Ejemplo de una clase

Asociaciones entre clases

La asociación expresa una conexión bidireccional entre las clases. Puede definirse como una abstracción de la relación existente en los enlaces entre los objetos.

Las asociaciones poseen una multiplicidad, también denominada cardinalidad, para describir el número de instancias de una clase determinada que están asociados con el número de instancias de otra clase determinada [17].

La multiplicidad de las asociaciones que se utiliza en los diagramas de clases es la siguiente:

1 Un elemento relacionado.
0..1 Uno o ningún elemento relacionado.
0..* Varios elementos relacionados o ninguno.
1..* Varios elementos relacionados pero al menos uno.
* Varios elementos relacionados.
M..N Entre M y N elementos relacionados.

Las asociaciones son definidas además por un rol que es un nombre al final de la asociación que describe la semántica de la relación en el sentido indicado. Por lo tanto cada asociación tiene dos roles; cada rol es una dirección en la asociación.

En la Fig. 2 se describe la asociación *"Corresponds"* entre la clase *Gene* y la clase *Gene-Segment*. La cardinalidad presentada es (1..1: 0..*) lo que especifica que un objeto

de la clase *Gene* corresponde con varios objetos de la clase *Gene-Segment* o ninguno. Mientras que un objeto de la clase *Gene-Segment* está correspondido específicamente con un objeto de la clase *Gene*. Debe observarse que los roles son *"Corresponds"* y *"is correspond"*. En ocasiones no se escriben los dos roles porque uno es inferido a partir de un rol definido, como el caso de esta asociación.

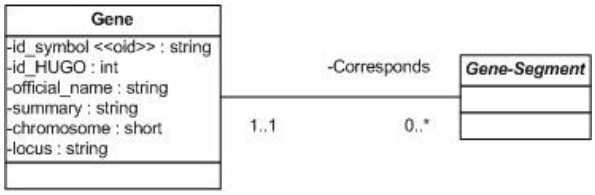


Fig. 2 Asociación entre clases.

Además de las asociaciones básicas, existen diferentes tipos de asociaciones que hacen que el modelo conceptual sea más descriptivo. Los tipos especiales de asociación utilizados en la construcción del esquema conceptual del genoma humano son los siguientes:

Agregación

Es una asociación especial, una relación del tipo "todo/parte" dentro de la cual una o más clases son partes de un conjunto.

En la Fig. 3 se describe una asociación de agregación desde la clase *ResearchCentre* hacia la clase *Genome* para describir que un centro de investigaciones puede agrupar estudios sobre varios genomas, que son parte de las investigaciones que realizan. De tal forma que una instancia de la clase *ResearchCentre* determinada será un conjunto de instancias de la clase *Genome* agregados o incluso puede no ser ese conjunto de objetos agregados.

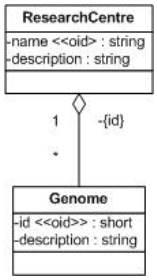


Fig. 3 Agregación entre clases.

Composición

La composición es una forma 'fuerte' de agregación. Es diferente a la agregación por el hecho de que tanto el todo como las partes tienen el mismo ciclo de vida y que una instancia puede pertenecer solamente a una composición.

En la Fig. 4 se presenta una asociación de composición desde la clase *Genome* hacia la clase *Chromosome* para describir el hecho de que un genoma siempre estará compuesto por cromosomas. De tal forma que un objeto de la clase *Genome* determinado será una composición de los objetos de la clase *Chromosome* agregados.

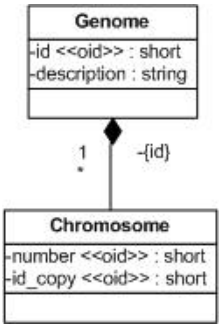


Fig. 4 Composición entre clases.

Generalización / Especificación

Una generalización se refiere a una relación entre una clase general (superclase o padre) y una versión más específica de dicha clase (subclase o hija).

Las subclases heredan las características de sus superclases es decir, atributos y relaciones. Además, las subclases pueden incorporar nuevos atributos o relaciones que las superclases no tienen.

Debe observarse que, al ocurrir una generalización/especialización en un par de clases, la clase que hereda a la subclase es más general mientras que la clase que ha heredado de la superclase es mas especifica. La identificación de una especificación o una generalización depende de la clase a partir de la cual se observe la asociación.

En la Fig. 5 Se presenta una generalización de la clase *ChromosomeSegment* hacia las clases *Gene-Segment* y *NonGeneSegment* para describir que generalmente los objetos de la clase *Gene-Segment* y *NonGeneSegment* son considerados objetos de *ChromosomeSegment* pero las propiedades diferentes entre *Gene-Segment* y *NonGeneSegment*, que permiten identificar la parte codificante de la parte no codificante dentro de los cromosomas, permiten especializar a la clase *ChromosomeSegment* en *Gene-Segment* y *NonGeneSegment*.

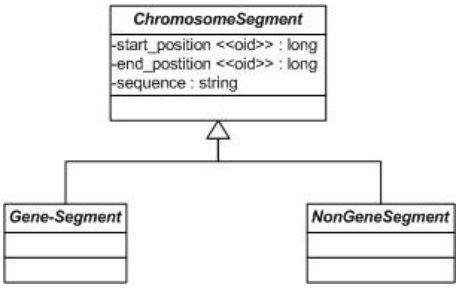


Fig. 5 Generalización de clases.

Es importante mencionar que al usar discriminadores, también llamados denominadores específicos, se pueden tener varias especializaciones de una misma superclase.

En la Fig. 6 se describe una especialización jerárquica de la clase *Variation*, utilizando tres discriminadores *Effect, Location, Description*. Esto hace más expresivo el esquema conceptual.

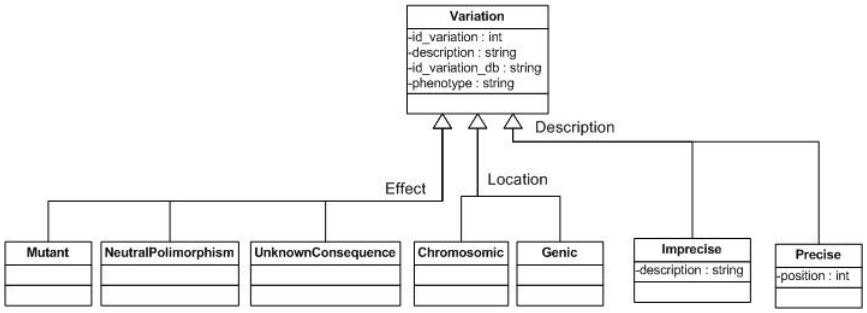


Fig. 6 Especialización con el uso de discriminadores.

El diagrama de clases define la estructura del mundo mientra que un diagrama de objetos define el mundo en un instante dado.

La relación entre los objetos se corresponde con la de sus clases de tal forma que toda instancia de una clase X tiene enlaces con instancias de Y de acuerdo al tipo de relación, atributos y modificadores que X tiene con Y en el modelo de clases [17].

En resumen puede decirse que el diagrama de clases y de objetos se usa para modelar el dominio de un sistema de información. Un Modelo Conceptual es un conjunto de clases que denotan "tipos" de clases conceptuales del mundo real. Mientras que los objetos denotan clases conceptuales del mundo real.

La forma de abstraer información del dominio para identificar clases de conceptos y los niveles de descripción, que deben abordarse, para que el modelo conceptual final describa el dominio del SI que corresponda con la realidad, se han establecido en un estándar denominado MDA que se menciona a continuación.

2.1.5 MDA

La arquitectura dirigida por modelos (Model-Driven Architecture o MDA) [11] es una aproximación al diseño de software, respaldado por el *Object Management Group* [12]. MDA ha sido propuesta para incluir los principios de la ingeniería dirigida a modelos de los sistemas de información. Es una arquitectura que proporciona un conjunto de principios para estructurar especificaciones expresadas como modelos.

Usando la metodología MDA, la funcionalidad del sistema será definida en primer lugar como un modelo independiente de la plataforma (Platform-Independent Model o PIM) a través de un lenguaje específico para el dominio del que se trate. Dado un modelo de definición de la plataforma (Platform Definition Model o PDM), el modelo PIM puede traducirse a uno o más modelos específicos de la plataforma (Platform-specific models o PSMs) para la implementación correspondiente, usando diferentes lenguajes específicos del dominio, o lenguajes de propósito general [5].

Los principios de MDA pueden aplicarse a la mayoría de las áreas, ya que el PIM, independiente de la tecnología y de la arquitectura es adaptado a los sistemas de las diferentes áreas.

El modelo MDA está relacionado con múltiples normas, incluyendo el lenguaje de modelado unificado (Unified Modeling Language o UML), Meta-Object Facility (MOF), XML Metadata Interchange (XMI), Enterprise Distributed Object Computing (EDOC), el Software Process Engineering Metamodel (SPEM) y el Common Warehouse Metamodel (CWM). El término "arquitectura" en los metamodelos no se refiere a la arquitectura del sistema modelizado sino más bien a la arquitectura de los distintos estándares y formas del modelo que sirven de base tecnológica al MDA.

Uno de los principales objetivos de MDA es separar el diseño de la arquitectura de las tecnologías de construcción de los SI. Logrando que el diseño y la arquitectura de los SI puedan ser alterados independientemente de sus implementaciones finales.

MDA se asegura de que el modelo independiente de la plataforma (PIM), el cual representa un diseño conceptual que concreta los requerimientos funcionales, sobrevive a los cambios que se produzcan en las tecnologías de fabricación y en las arquitecturas software.

Los niveles de descripción involucrados en la arquitectura MDA se ilustran en Fig. 7:

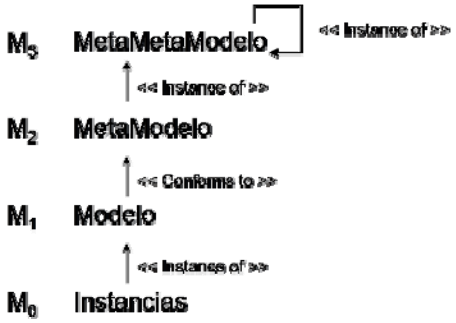


Fig. 7 Model-Driven Architecture.

El nivel M_3 MetaMetaModelo, es el nivel más abstracto. Aquí se encuentra un lenguaje abstracto para definir metamodelos: el Meta-Object Facility (MOF).

El nivel M_2 MetaModelo, define la estructura semántica y las restricciones para una familia de modelos. Un ejemplo de esto es el MetaModelo de UML que especifica las primitivas para cada uno de los diagramas implicados: diagramas de clases, de casos de uso, de secuencia, etc.

El nivel M_1 Modelo, cada modelo esta definido en un lenguaje de su metamodelo. En este nivel se incluyen los modelos específicos definidos en los metamodelos del nivel M_2. Un ejemplo de este es el diagrama de clases de UML.

El nivel M_0 Instancias, este nivel representa a las instancias de los modelos. Un ejemplo de esto son los esquemas conceptuales.

Puede decirse que el esquema conceptual del genoma humano, se ha construido considerando los principios que marca la arquitectura MDA con la finalidad de trabajar con niveles mas altos de abstracción de forma que los biólogos y los ingenieros de software puedan hablar un lenguaje común y que además la especificación del dominio, a través de un esquema conceptual, pueda utilizar las ventajas que proporcionan a los sistemas de información las técnicas de modelado conceptual.

2.2 *Biología Molecular*

Para entender la constitución y el funcionamiento del genoma humano y poder modelarlo conceptualmente, es fundamental tener un panorama general de lo que se considera Biología Molecular. Por lo que a continuación se enuncian conceptos importantes relacionados a esta rama de estudio de la Biología y posteriormente se describe el dominio del problema, en este caso el genoma humano, para el que se diseña un esquema conceptual.

La Biología Molecular es el estudio de la vida a un nivel molecular. Esta área esta relacionada con otros campos de la Biología y la Química, particularmente Genética y Bioquímica. La Biología Molecular concierne principalmente al entendimiento de las interacciones de los diferentes sistemas de la célula, lo que incluye muchísimas relaciones, entre ellas las del ADN con el ARN, la síntesis de proteínas, el metabolismo, y el cómo todas esas interacciones son reguladas para conseguir un afinado funcionamiento de la célula.

Al estudiar el comportamiento biológico de las moléculas que componen las células vivas, la Biología Molecular roza otras ciencias que abordan temas similares. Para el caso concreto de la elaboración de este trabajo de investigación, se habla de la Genética que se interesa por la estructura y funcionamiento de los genes y por la regulación de la síntesis intracelular de enzimas y de otras proteínas.

2.2.1 El genoma humano

Al describir el genoma humano como un dominio para un Sistema de Información es necesario considerar el conjunto de conocimiento científico derivado de los principios que marcan la Genética y la Biología Molecular. Bajo estos principios deberá describirse el dominio de este Sistema de Información para integrar todos los conceptos relevantes en un esquema conceptual.

Para entender los procesos biológicos relacionados con el Genoma Humano, se deben tener algunas bases en Genética, además de tener conocimiento de los elementos que interactúan, sus funciones y características. Por esta razón es importante iniciar la

descripción del dominio con información acerca de los genomas independientemente del organismo al que pertenecen.

Un genoma (Fig. 8, fuente [33]) es todo el material genético contenido en las células de un organismo en particular. En los seres eucarióticos, aquellos cuyas células contienen la información genética encerrada dentro de una doble membrana que delimita el núcleo celular, el genoma se refiere al ADN contenido en el núcleo organizado en cromosomas.

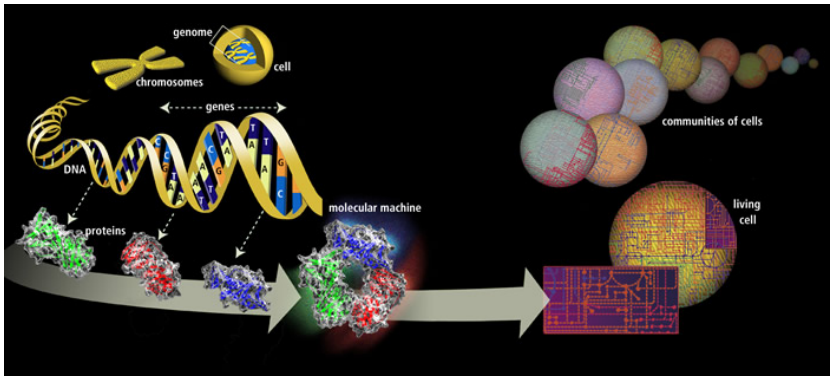


Fig. 8 Genoma.

Cada organismo posee un número concreto de cromosomas, en el caso del hombre es de 23 pares ya que sus células somáticas son diploides (2n). Sólo los gametos (óvulos y espermatozoides) pueden catalogarse como haploides (n). El término diploide indica que un organismo tiene dos copias del genoma en sus células debido a la presencia de pares de cromosomas homólogos que provienen del padre y de la madre; el término haploide indica que el organismo tiene una sola copia del genoma en sus células.

El Genoma Humano (GH) es la secuencia completa de ADN de un ser humano. Esta secuencia se describe en los 46 cromosomas que posee un ser humano. De estos hay 44 autosomas, 22 heredados de la madre y 22 del padre, y dos cromosomas sexuales que determinan el sexo del individuo: un cromosoma X, heredado de la madre, y un X (en las mujeres) o un Y (en los varones), heredado del padre.

El GH está compuesto por aproximadamente entre 25000 y 30000 genes distintos, unos son genes estructurales, otros reguladores, otros codifican proteínas; sin embargo,

generalmente cada uno de estos genes contiene codificada la información necesaria para la creación de una o varias proteínas.

2.2.2 El ADN

El ácido desoxirribonucleico, frecuentemente abreviado como ADN es una macromolécula que forma parte de todas las células. Contiene la información genética usada en el desarrollo y el funcionamiento de los organismos vivos conocidos y de algunos virus, siendo el responsable de su transmisión hereditaria.

Por lo tanto el principal actor en proceso biológico del genoma humano es el ADN que se define desde un enfoque químico como una macromolécula que resulta de la unión covalente de un grupo fosfato y una base heterocíclica con una pentosa. Esta macromolécula está formada por moléculas monoméricas orgánicas denominadas nucleótidos. A su vez cada nucleótido esta formado por un azúcar (la desoxirribosa), una base nitrogenada (adenina (A), timina (T), citosina (C) o guanina (G)) y un grupo fosfato. Su estructura detallada será explicada más adelante. Lo que distingue a un nucleótido de otro es la base nitrogenada, por ello una secuencia de ADN se especifica nombrando solamente la secuencia de sus bases. Además esta macromolécula contiene toda la información genética usada en el desarrollo y funcionamiento de los seres vivos y se hereda de padres a hijos.

Estructura del ADN

James Watson y Francis Crick [31] descubrieron que la molécula de ADN está formada por dos largos filamentos, cada uno de ellos es una cadena de nucleótidos, que se enrollan entre sí para dar lugar a una doble hélice, además también descubrieron que el ADN se podía "desenrollar" para que fuera posible su lectura o copia.

La doble hélice que forman los dos filamentos (cadena de nucleótidos) es parecida a una escalera de caracol. La parte lateral o "barandilla" de la escalera está formada por azúcares (desoxirribosa) y fosfatos y los peldaños son pares de bases. (Fig. 9).

La adhesión de las dos hebras de ácido nucleico se debe a un tipo especial de unión química conocido como puente de hidrógeno. Los puentes de hidrógeno son uniones más débiles que los típicos enlaces químicos, esto significa que las dos hebras de la hélice pueden separarse con facilidad, quedando intactas.

Una larga hebra de ácido nucleico está enrollada alrededor de otra hebra y forma un par entrelazado. Dicha hélice mide 3,4 nm de paso de rosca y 2,37 nm de diámetro, y está formada, en cada vuelta, por 10,4 pares de nucleótidos enfrentados por sus bases nitrogenadas.

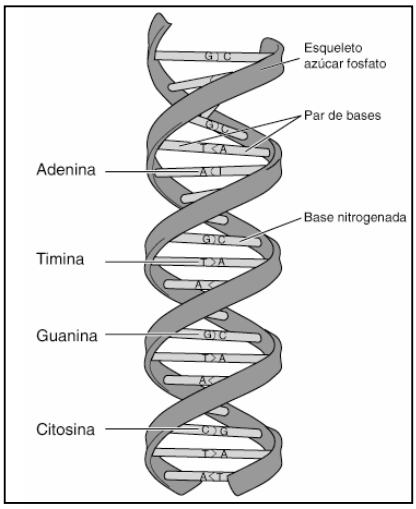


Fig. 9 La doble hélice del ADN.

El tipo de nucleótidos y el orden -denominado secuencia- en que se disponen estas moléculas es la manera cómo se escribe la información genética, mediante una especie de alfabeto de 4 letras.

Si todo el ADN contenido en el núcleo de las células humanas se estirara llegaría a medir hasta 1,8 metros, es decir, unas 300.000 veces más que el diámetro del núcleo. Para evitar este problema, el ADN está altamente plegado formando unas estructuras denominadas cromosomas.

Por lo tanto las tres características que permiten a la molécula de ADN ser la depositaria de la información genética de un organismo son: que la molécula de ADN contiene información basada en el orden y composición de los nucleótidos que la forman; que es

capaz de pasar esta información de generación en generación gracias a que cada cadena puede servir como molde para fabricar su complementaria; y que es flexible, lo que permite que pueda almacenarse toda la información que requiere un ser vivo para ser como es y realizar sus funciones en un espacio tan pequeño como el interior de las células.

Componentes

La estructura de soporte de una hebra de ADN está formada por unidades alternas de grupos fosfato y azúcar. El azúcar en el ADN es una pentosa, concretamente, la desoxirribosa (Fig. 10, fuente [34]).

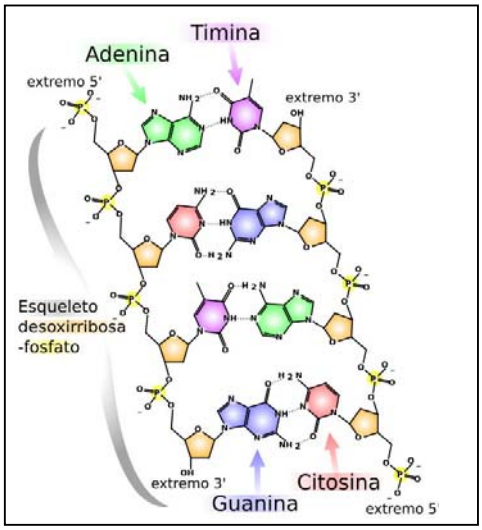


Fig. 10 Componentes del ADN.

- *Acido fosfórico*: Cada nucleótido puede contener uno (monofosfato: AMP), dos (difosfato: ADP) o tres (trifosfato: ATP) grupos de ácido fosfórico.

- *Desoxirribosa*: Es un monosacárido de 5 átomos de carbono (una pentosa) derivado de la ribosa. Las moléculas de azúcar se unen entre sí a través de grupos fosfato, que forman enlaces entre los átomos de carbono tercero (3′, «tres prima») y quinto (5′, «cinco prima») de dos anillos adyacentes de azúcar. La formación de enlaces asimétricos implica que cada hebra de ADN tiene una

dirección. En una doble hélice, la dirección de los nucleótidos en una hebra (3′ → 5′) es opuesta a la dirección en la otra hebra (5′ → 3′). Esta organización de las hebras de ADN se denomina antiparalela; son cadenas paralelas, pero con direcciones opuestas. De la misma manera, los extremos asimétricos de las hebras de ADN se denominan extremo 5′ («cinco prima») y extremo 3′ («tres prima») respectivamente.

- *Bases nitrogenadas*: Las cuatro bases nitrogenadas mayoritarias que se encuentran en el ADN son la adenina (abreviado A), citosina (C), guanina (G) y timina (T). Cada una de estas cuatro bases está unida al armazón de azúcar-fosfato a través del azúcar para formar el nucleótido completo (base-azúcar-fosfato).

Emparejamiento de los nucleótidos

El rasgo fundamental es que las bases de nucleótidos de una hebra de ADN corresponden con la especie de nucleótidos de la otra, en el sentido de que la Adenina siempre corresponde con la Timina (lo que se denomina A....T) y la Guanina siempre corresponde con la Citosina (G...C) ver Fig.11.

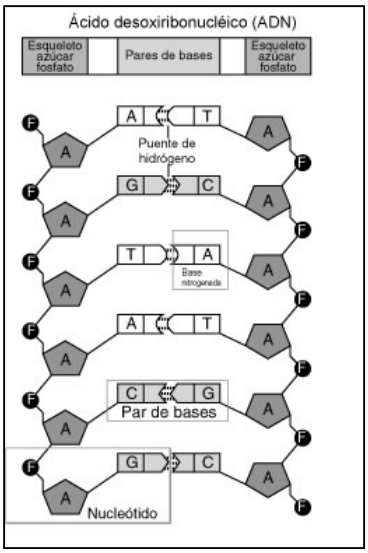


Fig. 11 Emparejamiento de nucleótidos.

Este emparejamiento corresponde a la observación realizada por Erwin Chargaff [32], de que en todas las muestras, la cantidad de Adenina es siempre la misma que la de Timina, como ocurre con la Guanina y la Citosina, así se aseguran cantidades iguales. Así, una purina (Adenina y Guanina) siempre se empareja con una pirimidina (Timina y Citosina), así se hace uniforme la doble hélice.

La cantidad de purina (A+G) es siempre igual a la cantidad de pirimidina (T+C). Se estima que el genoma humano tiene alrededor de 3.000 millones de pares de bases. Dos unidades de medida muy utilizadas son la kilobase (kb) que equivale a 1.000 pares de bases, y la megabase (Mb) que equivale a un millón de pares de bases.

2.2.3 El ARN

Para que la información que contiene el ADN pueda ser utilizada por la maquinaria celular, debe copiarse en primer lugar en secuencias de nucleótidos, más cortos y con unas unidades diferentes, llamados ARN.

Las moléculas de ARN se copian exactamente del ADN mediante un proceso denominado transcripción. Una vez procesadas en el núcleo celular, las moléculas de ARN pueden salir al citoplasma para su utilización posterior. La información contenida en el ARN se experimenta el proceso de traducción usando el código genético, que especifica la secuencia de los aminoácidos de las proteínas.

Por lo tanto el ARN es la molécula que dirige las etapas intermedias de la síntesis proteica; el ADN no puede actuar solo, y se vale del ARN para transferir esta información vital durante la síntesis de proteínas (producción de las proteínas que necesita la célula para sus actividades y su desarrollo). Varios tipos de ARN regulan la expresión génica, mientras que otros tienen actividad catalítica. Es evidente que el ARN es mucho más versátil que el ADN.

Estructura del ARN

El ARN (Acido Ribonucleico), al igual que el ADN, está formado por una cadena larga de nucleótidos. Cada uno de estos nucleótidos está formado por una molécula de un azúcar llamado Ribosa, un grupo Fosfato y una base nitrogenada, que son las mismas que en el ADN pero la Timina se sustituye por el Uracilo, ver Fig.12.

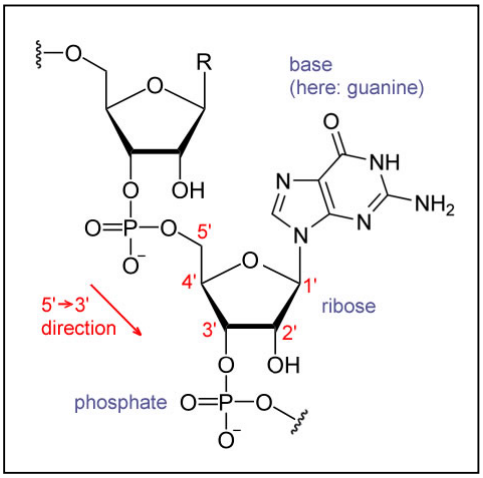


Fig. 12 Estructura del ARN.

Los carbonos de la ribosa se numeran de 1' a 5' en sentido horario. La base nitrogenada se une al carbono 1'; el grupo fosfato se une al carbono 5' y al carbono 3' de la ribosa del siguiente nucleótido. El fosfato tiene una carga negativa lo que confiere al ARN carácter polianiónico. Las bases púricas (Adenina y Guanina) pueden formar puentes de hidrógeno con las pirimidínicas (Uracilo y Citosina) según el esquema C=G y A=U.

El ARN se diferencia del ADN químicamente, ya que la molécula de azúcar del ARN contiene un átomo de oxígeno que falta en el ADN, además contiene la base Uracilo (U) en lugar de la Timina (T) del ADN. Otra diferencia importante es que el ADN es una hélice doble, sin embargo el ARN casi siempre está formado por una única cadena. Estas características hacen la molécula de ARN más frágil que la molécula de ADN. Ver Fig. 13.

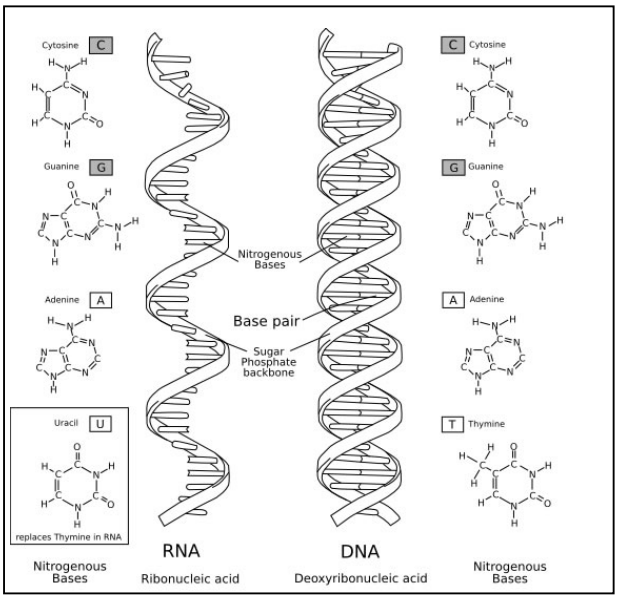


Fig. 13 Diferencias entre el ADN y el ARN.

Tipos de ARN

Existen varios tipos diferentes de ARN pero pueden ser clasificados en dos grandes grupos: el ARN codificante, que interviene el proceso de síntesis de proteínas; y el ARN no codificante.

Todos los tipos de ARN son importantes, sin embargo, para los fines de la realización de este trabajo de investigación, el tipo de ARN que demanda principal atención es el ARN mensajero (ARNm).

El ARN mensajero (ARNm) es el tipo de ARN que lleva la información del ADN a los ribosomas, para realizar el proceso de la síntesis de proteínas. La secuencia de nucleótidos del ARNm determina la secuencia de aminoácidos de la proteína. Por lo tanto el ARNm es denominado ARN codificante.

Sin embargo, los otros tipos de ARN, denominados ARNs no codificantes, apoyan a los procesos de traducción, como el caso de ARN de transferencia (ARNt) y el ARN ribosómico (ARNr); o bien son diversos tipos de ARN que apoyan la regulación de la expresión génica.

2.2.4 Cromosomas

Se denomina cromosoma a cada uno de los pequeños cuerpos en forma de bastoncillos en que se organiza la cromatina del núcleo celular durante las divisiones celulares (mitosis y meiosis) (ver Fig.14.) La cromatina es un material microscópico que lleva la información genética de los organismos eucariotas y está constituida por ADN asociado a proteínas.

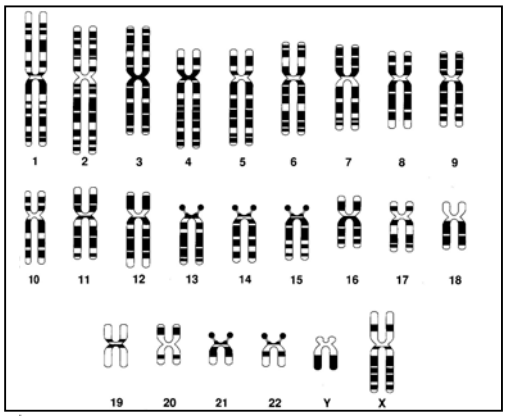


Fig. 14 Cromosomas del genoma humano.

El Genoma Humano, esta compuesto por 23 pares de cromosomas, 22 pares son autosomas (cromosomas no sexuales) y un par de cromosomas sexuales: X, Y. Con un tamaño total aproximado de 3200 millones de pares de bases de ADN.

En un cromosoma, además de la información genética, se pueden distinguir una serie de elementos diferentes como los centrómeros, Telómeros, y ORI's, cada uno de estos caracterizados por contener secuencias de nucleótidos específicas.

Cuando se examina con detalle la longitud de los cromosomas durante la mitosis, se observa que los cromosomas presentan una forma y un tamaño característicos. Cada cromosoma tiene una región condensada, llamada centrómero, que confiere la apariencia general de cada cromosoma y que permite clasificarlos según su posición en metacéntrico (centrómero en el centro), submetacéntrico (un poco más abajo o arriba del centro), acrocéntrico (un brazo corto y el otro largo) y telocéntrico (el centrómero está en un extremo).

Los telómeros son los extremos de los cromosomas. Son regiones de ADN no codificante, altamente repetitivas, cuya función principal es la estabilidad estructural de los cromosomas en las células eucariotas, la división celular y el tiempo de vida de las estirpes celulares.

El ORI conocido como el Origen de la Replicación, es el lugar del cromosoma donde se inicia la replicación de la cadena de ADN. Se trata, por lo tanto, de una determinada secuencia de nucleótidos a partir de la cual se desarrolla una horquilla de replicación que dará lugar a las dos cadenas idénticas de ADN resultantes.

En resumen puede decirse que cada cromosoma es una única molécula de ADN que, a su vez, está formado por millares de nucleótidos. Si se escribiera en el alfabeto de 4 letras toda la información genética que contiene una célula humana, se llenaría un libro con más de 500 000 páginas.

2.2.5 Genes

Las secuencias de ADN que constituyen la unidad fundamental, física y funcional de la herencia se denominan genes (Fig. 15). Cada gen contiene una parte que se transcribe a ARN y otra que se encarga de definir cuándo y dónde deben expresarse. La información contenida en los genes se emplea para generar ARN y proteínas, que son los componentes básicos de las células.

En un único cromosoma, y por lo tanto en una única molécula de ADN, se encuentran alineados muchísimos genes, sin embargo, se denomina gen al segmento de ADN que lleva la información para sintetizar una proteína.

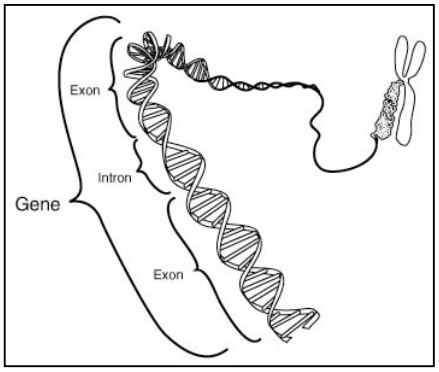


Fig. 15 Gen.

La secuencia de nucleótidos, que describe la expresión de un gen, puede llegar a formar proteínas o no, dependiendo de la funcionalidad asignada para la célula que aporten los cromosomas. En general, los genes son secuencias que serán transcritas en proteínas. Los genes poseen diferentes números de nucleótidos en sus secuencias por lo tanto, cada gen debe ser localizable dentro del cromosoma para poder utilizar la información genética que posee.

Este mecanismo de localización del gen dentro del cromosoma, lleva a hablar sobre el Promotor, secuencia de ADN que indica a las enzimas de transcripción donde empezar a traducir el gen. Cada gen tiene su propio Promotor. En Células Eucariotas la secuencia del Promotor siempre es la misma para todos los genes.

Al igual que se debe de localizar el inicio de la secuencia de nucleótidos considerada información genética (llamada secuencia transcribible o expresión del gen), se debe saber donde finaliza. Esto hace necesario hablar del Terminador, que es la secuencia de ADN que indica donde termina la secuencia transcribible y por lo tanto debe parar el proceso de transcripción. El gen junto con el promotor y el terminador se denominan Unidad de Transcripción (Fig. 16).

Fig. 16 Unidad de transcripción en los genes.

Una unidad de transcripción es una secuencia de ADN que se transcribe en una molécula de ARN mediante una reacción enzimática por la ARN-polimerasa.

La secuencia de nucleótidos de un gen define la proteína que un organismo es capaz de sintetizar en uno o varios momentos de su vida, a partir de la información de dicha secuencia. La relación entre la secuencia de nucleótidos y la secuencia de aminoácidos de la proteína es determinada por un mecanismo celular de traslación, conocido de forma general, como código genético.

En el ADN los genes son considerados como tal a partir de que se expresan, es decir, si no ocurre la expresión de un gen determinado, no puede justificarse que una secuencia de nucleótidos pertenece a ese gen específicamente. La expresión de los genes es controlada por otra secuencia de nucleótidos también presente en el ADN. Esta secuencia es denominada secuencia reguladora (también se denomina región reguladora o "elemento regulador"). Es un segmento de ADN en donde los factores de transcripción se ligan preferentemente. Estas regiones o secuencias reguladoras, que corresponden a tramos normalmente cortos del ADN, se encuentran posicionadas adecuadamente en el genoma, usualmente a una corta distancia del gen que regulan.

2.2.6 Alelos

Un alelo es cada una de las formas alternativas que puede tener un gen que se diferencian en su secuencia y que se puede manifestar en modificaciones concretas de la función de ese gen. Los seres humanos poseen dos alelos de cada gen, uno de ellos procedente del padre y el otro de la madre. Cada par de alelos se ubica en el mismo lugar dentro del cromosoma.

El concepto de alelo se entiende a partir de la palabra alelomorfo (en formas alelas) es decir, algo que se presenta de diversas formas dentro de una población de individuos. Los alelos, como formas alternas en las que se expresa un gen, difieren en secuencia o función.

Toda característica genéticamente determinada depende de la acción de cuando menos un alelo.

- Los alelos que varían en secuencia tienen diferencias en el ADN, como deleciones, inserciones o sustituciones.

- Los alelos que difieren en función pueden tener o no diferencias conocidas en las secuencias, pero se evalúan por la forma en que afectan al organismo.

En función de su expresión en el fenotipo, los alelos se pueden dividir en:

- *Alelos dominantes:* aquellos que aparecen en el fenotipo de los individuos heterocigotos (una copia diferente para cada alelo del gen homologo) para una determinada característica, además de en el homocigoto (dos copias iguales para el gen homologo).

- *Alelos recesivos:* los que están presentes en el fenotipo de un individuo heterocigoto y sólo aparecen en un individuo homocigoto cuando se trata de un gen recesivo.

Los alelos de un gen pueden ser muchos pero solo uno de ellos se considera el alelo silvestre o natural y los demás se clasifican como variantes alélicas. Independientemente de que la forma alternativa sea el alelo natural o alguna variante alélica, se mantiene la estructura génica genérica: unidades de trascripción (promotor, secuencia transcribible y terminador) y secuencias reguladoras.

En resumen puede decirse que un alelo es una de las varias formas en las que un gen se expresa. Estos se transmiten en la procreación de un ser humano. Un alelo puede ser dominante (se expresará sólo con una copia y por lo tanto, si el padre o la madre lo poseen, el hijo lo expresará) o recesivo (se necesitan dos copias del mismo gen, dos alelos, para que se exprese).

2.2.7 Proteínas

Las proteínas se forman en el ribosoma a partir de la información suministrada por los genes. Se describen como la combinación de 20 tipos de aminoácidos. Por lo tanto una proteína es la secuencia en la que se disponen dichos aminoácidos de acuerdo con las instrucciones indicadas por el ADN.

Los aminoácidos (Fig. 17) que componen las proteínas son 20: alanina, arginina, asparagina, aspartato, cisteína, fenilalanina, glicina, glutamato, glutamina, histidina, isoleucina, leucina, lisina, metionina, prolina, serina, tirosina, treonina, triptófano y valina.

Ala (A)	GCU, GCC, GCA, GCG	Lys (K)	AAA, AAG
Arg (R)	CGU, CGC, CGA, CGG, AGA, AGG	Met (M)	AUG
Asn (N)	AAU, AAC	Phe (F)	UUU, UUC
Asp (D)	GAU, GAC	Pro (P)	CCU, CCC, CCA, CCG
Cys (C)	UGU, UGC	Sec (U)	UGA
Gln (Q)	CAA, CAG	Ser (S)	UCU, UCC, UCA, UCG, AGU, AGC
Glu (E)	GAA, GAG	Thr (T)	ACU, ACC, ACA, ACG
Gly (G)	GGU, GGC, GGA, GGG	Trp (W)	UGG
His (H)	CAU, CAC	Tyr (Y)	UAU, UAC
Ile (I)	AUU, AUC, AUA	Val (V)	GUU, GUC, GUA, GUG
Leu (L)	UUA, UUG, CUU, CUC, CUA, CUG		
Comienzo	AUG	Parada	UAG, UGA, UAA

Fig. 17 Aminoácidos.

Una proteína es diferente a otra debido a su secuencia específica de aminoácidos, que es la que le confiere la forma que la proteína adopta en el espacio y que es absolutamente básica para su función. Cuando una proteína pierde su forma nativa se dice que se desnaturaliza y esta desnaturalización suele conllevar una pérdida de funcionalidad.

2.2.8 Genotipo y Fenotipo

Un aspecto fundamental al estudiar el genoma humano y su constitución y funcionamiento, es poner especial atención en lo que se ha denominado Herencia Genética.

La Herencia Genética es la transmisión, a través de información genética contenida en el núcleo celular, de las características anatómicas y fisiológicas de un ser vivo a sus descendientes. El ser vivo resultante tendrá características de uno o los dos padres.

La función principal de la herencia es la especificación de las proteínas, el ADN es una especie de plano o receta para las proteínas. Unas veces la modificación del ADN que provoca disfunción proteica es llamada enfermedad, otras veces, en sentido beneficioso, dará lugar a lo que se conoce como evolución.

Cuando los genes se expresan se especifican las características que son hereditarias, este conjunto de características se conoce como Genotipo, conjunto de genes de un organismo, mientras que su manifestación exterior en ciertas características del individuo se conoce como Fenotipo. Puede decirse que el Fenotipo es la expresión del genotipo en un determinado ambiente. Los rasgos fenotípicos incluyen rasgos tanto físicos como conductuales. Dado que los fenotipos son mucho más fáciles de observar que los genotipos, la genética clásica usa los fenotipos para determinar las funciones de los genes.

2.2.9 Funcionamiento del genoma humano

El ADN puede ser visto como un almacén de información (mensaje) que se trasmite de generación en generación, con toda la información necesaria para construir y sostener el organismo en el que habita. Puede decirse que el ADN es el fichero genético en el que están impresas las instrucciones que necesita un ser vivo para nacer y desarrollarse a partir de la primera célula.

Se puede considerar que las obreras de ejecutar los mensajes del ADN, para construir y sostener un organismo, son las proteínas. Estas pueden ser estructurales como las proteínas de los músculos, cartílagos, pelo, etc., o bien funcionales como las de la hemoglobina o las de innumerables enzimas del organismo humano.

El ser humano tiene alrededor de treinta mil proteínas diferentes que están hechas de veinte aminoácidos diferentes. Sin embargo, el código genético de las células se

encuentra en forma de ADN, es decir, en las moléculas de ADN existe información para sintetizar las proteínas que utiliza el organismo; pero el ADN no se traduce directamente en proteínas por lo que este proceso no es lineal e implica varios procesos intermedios para que la información genética presente en el ADN pueda llegar a las proteínas y estas construyan y sostengan a un organismo.

Es importante mencionar que todas las células de un organismo disponen de la información necesaria para realizar todas las funciones corporales, sin embargo, cada tipo de célula se especializa en realizar una función determinada. Esta diferencia reside en el tipo de proteínas presentes y necesarias en cada célula. Por lo tanto, cada tipo de célula se caracteriza por expresar, sólo algunos de los genes de los que dispone en su genoma.

El funcionamiento del genoma humano puede resumirse de forma muy general en los siguientes pasos:

1. Obtener la información genética contenida en el ADN a través de la expresión de los genes.
2. Transcribir la información de ADN, resultante de la expresión de los genes, a ARN para que pueda ser traducida.
3. Traducir el ARN, resultado de la transcripción, con el código genético para formar la secuencia de aminoácidos que integran la proteína. Esto también es denominado síntesis de proteínas.
4. Ejecutar las instrucciones del ADN a través de la proteína construida que realizará una función celular determinada.

Proceso de Transcripción

La transcripción de la información genética, que el ADN proporciona a partir de la expresión de un determinado gen, es el proceso de crear una copia temporal del ADN, esta copia se denomina ARN.

La transcripción es necesaria ya que la síntesis de proteínas se realiza en el citoplasma, es decir, fuera del núcleo de la célula. Puesto que el ADN esta siempre en el núcleo de la célula, la transcripción permite que la información viaje desde el núcleo celular al citoplasma en forma de ARN mensajero. De esta forma el ADN siempre se mantiene seguro. Mientras tanto el ARN asume el riesgo de dejar el núcleo de la célula y salir al citoplasma. El ARN mensajero instruye a la maquinaria que elabora las proteínas, para que ensamble los aminoácidos en el orden preciso para armar la proteína.

El ARN se transcribe a partir de una de las dos hebras de la molécula de ADN. Que sólo se transcriba una hebra no significa que siempre sea la misma a lo largo de todo el cromosoma. Puede transcribirse una hebra en un sitio y otra en otro.

Es importante mencionar que en la transcripción se cambian las bases nitrogenadas de Timina (T) presentes en el ADN por la base nitrogenada Uracilo (U) y se obtiene una molécula de ARNm de simple cadena.

El proceso de transcripción implica los siguientes subprocesos, que ocurren en el orden que a continuación se describen:

Proceso de Iniciación:

El gen que se va a expresar debe ser localizado. Un grupo de enzimas buscan el Promotor de la unidad de transcripción, cuando lo encuentran la ARN polimerasa abre la molécula de ADN y lee la primera base nitrogenada de la unidad de transcripción. Cuando se forma el complejo abierto, la ARN polimerasa comienza a unir ribonucleótidos. La etapa de iniciación termina al formarse el primer enlace entre nucleótidos.

Proceso de Elongación:

La Elongación, es el proceso en el cual el ARN Polimerasa sigue abriendo la hélice de ADN, y sintetizando ARNm hasta que se transcribe toda la unidad de transcripción.

Proceso de Terminación:

Este proceso se inicia cuando la unidad de transcripción ha sido transcrita completamente y la ARN Polimerasa encuentra una secuencia (Terminador) que le indica que debe parar de transcribir. Una vez transcrita esta secuencia, para la transcripción.

Después de realizarse el proceso de Terminación el ARNm se separa completamente del ADN (que recupera su forma original) y también de la ARN polimerasa.

Es importante observar que las unidades de transcripción contienen secuencias codificantes alternadas con secuencias no codificantes, conocidas como Exones e Intrones respectivamente.

- Exones: Contienen la información para producir la proteína codificada en el gen, cada exón codifica una porción específica de la proteína completa, de manera que el conjunto de exones forma la región codificante del gen.

- Intrones: Región del ADN que debe ser eliminada del ARNm, pues son fragmentos de ADN carentes de información.

Con la presencia de secuencias codificantes (exones) y secuencias no codificantes (intrones) en las unidades de transcripción, se hace necesaria la realización de un proceso de edición de las secuencias después de que el proceso de transcripción haya finalizado.

Proceso Splicing

El proceso de edición consiste en remover los intrones de la secuencia que describe la unidad de transcripción y quedarse únicamente con los exones que son la región codificante. Este proceso también es llamado *Splicing* (Fig. 18)**,** en el cual se eliminan todos los Intrones y se unen los exones continuamente sin interrupciones entre ellos para así formar el ARNm maduro.

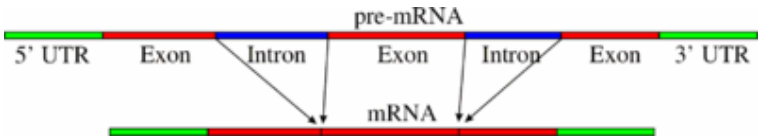


Fig. 18 Proceso Splicing.

Sin embargo, también es posible que ocurra un Splicing de Intrones y Exones, conocido como Splicing Alternativo, produciendo diferentes ARNm de un mismo gen. Gracias al Splicing Alternativo (Fig. 19), los 30.000 genes del ser Humano pueden producir alrededor de 90.000 proteínas diferentes.

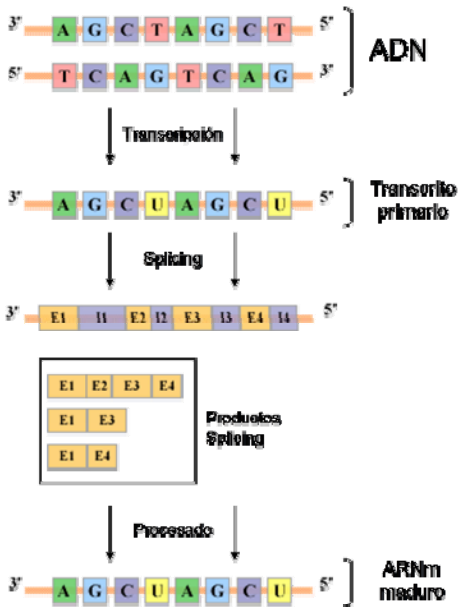


Fig. 19 Proceso de transcripción seguido de splicing alternativo.

Es importante mencionar que un determinado transcrito primario, integrado a partir de un proceso Splicing, define una determinada composición de intron/exon. La longitud de la secuencia de intrones y exones en una determinada composición pueden diferir entre otras composiciones incluso si pertenecen al mismo gen en el mismo momento.

Después de que ocurre el Splicing y todos los exones se han unido en conjunto, la molécula de ARNm tiene que pasar un proceso de maduración posteriormente emigra fuera del núcleo de la célula, para iniciar con el proceso de la traducción, el último paso para convertir las instrucciones genéticas del ADN en proteínas.

Proceso de Traducción

La traducción se enfoca principalmente en el proceso de fabricar una proteína siguiendo las instrucciones almacenadas en el ARNm que pasa de un lenguaje de 4 letras (los 4 nucleótidos), en que está transcrito el ARNm, al lenguaje de 20 letras (los 20 aminoácidos) en que están escritas las proteínas. Es evidente que la relación no puede ser de un aminoácido por cada nucleótido, ni tampoco por cada dos nucleótidos, porque los cuatro tomados de dos en dos, sólo dan dieciséis posibilidades. La correspondencia debe establecerse como mínimo entre cada aminoácido y tripletes de nucleótidos y es lo que se denomina Código Genético. Se tienen 64 tripletes diferentes (Fig. 20) que resultan de la combinación de los cuatro nucleótidos tomados de tres en tres con repetición, es obvio que algunos aminoácidos deben corresponderse con varios tripletes diferentes. Los tripletes que codifican aminoácidos se denominan codones.

		2ª base			
		U	C	A	G
1ª base	U	UUU Fenilalanina	UCU Serina	UAU Tirosina	UGU Cisteína
		UUC Fenilalanina	UCC Serina	UAC Tirosina	UGC Cisteína
		UUA Leucina	UCA Serina	UAA Ocre *Parada*	UGA [2]Ópalo *Parada*
		UUG Leucina	UCG Serina	UAG [3]Ámbar *Parada*	UGG Triptófano
	C	CUU Leucina	CCU Prolina	CAU Histidina	CGU Arginina
		CUC Leucina	CCC Prolina	CAC Histidina	CGC Arginina
		CUA Leucina	CCA Prolina	CAA Glutamina	CGA Arginina
		CUG [4]Leucina	CCG Prolina	CAG Glutamina	CGG Arginina
	A	AUU Isoleucina	ACU Treonina	AAU Asparagina	AGU Serina
		AUC Isoleucina	ACC Treonina	AAC Asparagina	AGC Serina
		AUA Isoleucina	ACA Treonina	AAA Lisina	AGA Arginina
		AUG [1]Metionina	ACG Treonina	AAG Lisina	AGG Arginina
	G	GUU Valina	GCU Alanina	GAU ácido aspártico	GGU Glicina
		GUC Valina	GCC Alanina	GAC ácido aspártico	GGC Glicina
		GUA Valina	GCA Alanina	GAA ácido glutámico	GGA Glicina
		GUG Valina	GCG Alanina	GAG ácido glutámico	GGG Glicina

Fig. 20 Código genético.

La síntesis de proteínas o traducción tiene lugar en los ribosomas del citoplasma celular. Los ribosomas son complejos supramoleculares encargados de ensamblar proteínas a partir de la información genética que les llega del ARNm. Los aminoácidos son transportados por el ARN de transferencia (ARNt), específico para cada uno de ellos, y

llevados hasta el ARN mensajero (ARNm), dónde se aparean el codón de éste y el anticodón del ARN de transferencia por complementariedad de bases y de ésta forma, se sitúan en la posición que les corresponde. Una vez finalizada la síntesis de una proteína, el ARN mensajero queda libre y puede ser leído de nuevo. De hecho, es muy frecuente que antes de que finalice una proteína, comienza otra, por lo que, una misma molécula de ARN mensajero, puede utilizarse por varios ribosomas simultáneamente.

Es importante mencionar que no toda la secuencia descrita por el ARNm es traducida a aminoácidos. La parte del ARNm que se traduce se denomina ORF (Open Reading Frame) que corresponde a cada una de las subsecuencias del ARNm comprendidas entre un codón de inicio y un codón de terminación. Por lo tanto, son las secuencias de nucleótidos dentro de los ORFs las que se alinean con los ribosomas para que puedan ser traducidas y formar proteínas.

El resultado de la traducción del ARNm por los ribosomas es una secuencia de aminoácidos que aún no es una proteína. Esta se denomina Polipéptido Primario que después de plegarse en su estructura 2d y 3d forma una molécula descrita por una cadena lineal de más de 10 y menos de 50 aminoácidos. Debido a esto varias cadenas de polipéptidos primarios pueden estar asociadas para formar una proteína. Ver Fig. 21.

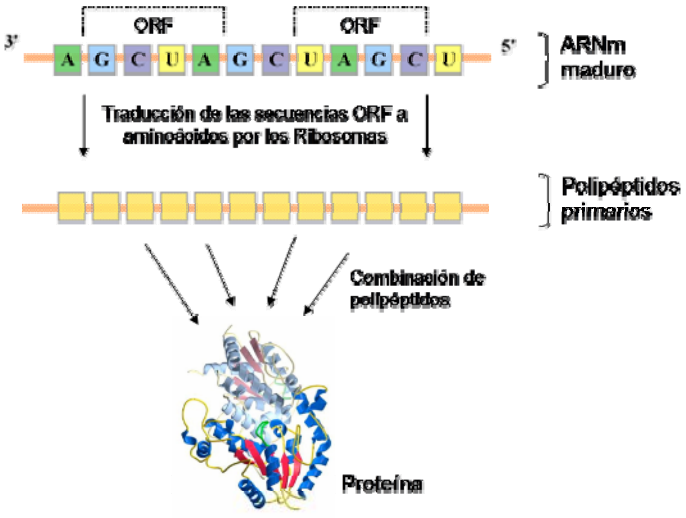


Fig. 21 Proceso de traducción.

Cuando los procesos de transcripción y traducción han sido terminados se obtiene como resultado la elaboración de una proteína que realizará una función determinada dentro de la célula.

2.2.10 Variaciones Genéticas

La molécula de ADN tiene la capacidad de desdoblarse, servir como molde y dar lugar a otra molécula idéntica, así es como pasa la información genética de padres a hijos. En general, una célula tiene una maquinaria muy sofisticada que permite hacer copias muy precisas de la molécula de ADN, incluso existen diversos sistemas de reparación. No obstante, en ocasiones se producen fallos. Cuando esto ocurre, se dice que ocurrido una mutación -es decir, un error en la copia de la secuencia - y si es suficientemente importante, puede llegar a cambiar la información que se transmite en la creación de la molécula de ADN de un nuevo organismo.

Los genomas de los seres humanos son 99.9% iguales, sin embargo se observan variaciones genéticas que hace que cada ser humano sea único; estas variaciones son las que modifican nuestras características físicas y nos permiten observar las características que se heredan.

Las variaciones genéticas son alteraciones en las secuencias de nucleótidos que conllevan a la síntesis de una proteína diferente y por tanto con función celular alterada o incluso a que esta no se pueda sintetizar. Esto se refleja en un funcionamiento diferente en el organismo. Debido a esto los estudios sobre las variaciones genéticas son importantes y la información que resulta de ellos debe ser tratada para poder identificar y corregir funciones diferentes dentro de la normalidad de un organismo determinado.

La variación genética se denomina comúnmente como mutación dentro de la genética y biología. Se define como una alteración o cambio en la información genética (genotipo) de un ser vivo y que, por lo tanto, va a producir un cambio de características, que se presenta súbita y espontáneamente, y que se puede transmitir o heredar a la descendencia. La unidad genética capaz de mutar es el gen que es la unidad de información hereditaria que forma parte del ADN. Una consecuencia de las mutaciones puede ser una enfermedad genética, sin embargo, aunque en el corto plazo pueden

parecer perjudiciales, a largo plazo las mutaciones son esenciales para nuestra existencia. Sin mutación no habría cambio y sin cambio la vida no podría evolucionar.

La definición de mutación a partir del conocimiento de que el material hereditario es el ADN y de la propuesta de la doble hélice para explicar la estructura del material hereditario [31], sería que una mutación es cualquier cambio en la secuencia de nucleótidos del ADN.

Las consecuencias fenotípicas de las mutaciones son muy variadas, desde grandes cambios hasta pequeñas diferencias tan sutiles que es necesario emplear técnicas muy elaboradas para su detección. Sin embargo, las mutaciones presentan características similares entres si que permiten clasificarlas.

Es importante mencionar, que el hecho de que ocurra una mutación en la secuencia de nucleótidos del ADN, no necesariamente produce un efecto patológico en el organismo pero de cualquier manera es una mutación. Debido a esto se ha considerado llamar variación genética a cualquier cambio que ocurra en la secuencia de nucleótidos del ADN y se denomina mutación a la variación genética que produce un efecto patológico en el organismo. Esta forma de denominación es exclusiva para este trabajo de investigación.

Al realizar una búsqueda exhaustiva por las bases de datos biológicos públicas, principalmente HGMD: Human Gene Mutation Database [7] y NCBI: National Center for Biotechnology Information [6], se observa que las variaciones genéticas ocurren respecto a alelos de referencia. Como se ha mencionado anteriormente los alelos son todas las posibles expresiones que un gen puede tener. Por lo tanto se establece un alelo natural como alelo de referencia y todas las demás expresiones del gen son consideradas variantes alélicas. La especificación del alelo natural es subjetiva y genérica, por lo que no todo lo que se considera alelo natural o de referencia pertenece a un solo individuo. Además el porcentaje de diferencia entre los genomas de los seres humanos, que nos identifica como seres únicos, presentará variaciones alélicas en un individuo respecto a otro. Estas variaciones no son siempre patológicas pero si pueden observarse diferentes características fenotípicas.

De acuerdo con la investigación realizada en las bases de datos públicas mencionadas anteriormente, se establece una clasificación para las variaciones alélicas basada en tres criterios principales:

1. *Descripción*: Esta clasificación es realizada a partir de la información que se conoce acerca de la variación, es decir, si la variación es precisa o imprecisa. Considerando que una variación que es precisa ha sido estudiada y concretamente definida, es posible identificar la posición de su ocurrencia en la secuencia de nucleótidos y precisar el tipo de cambio que presenta.

   Los cambios presentados en estas variaciones son:

   - *Inserción*: Describe los detalles de las variantes alélicas que presentan inserciones de nucleótidos respecto a la secuencia descrita en un alelo de referencia determinado.

   - *Deleción*: Describe los detalles de las variantes alélicas en las que ciertos nucleótidos fueron borrados en una posición específica de acuerdo con la secuencia descrita en un alelo de referencia determinado.

   - *Indel*: Describe los detalles de las variantes alélicas en las que ocurrió una deleción de ciertos de nucleótidos y después ocurrió una inserción de uno o varios nucleótidos un determinado numero de veces.

   - *Inversión*: Describe los detalles de las variantes alélicas en las que ocurre una inversión en el orden de los nucleótidos respecto a la secuencia que describe un alelo de referencia determinado.

2. *Ubicación*: Esta clasificación es realizada a partir del rango que abarca la variación, es decir, si la variación se asocia únicamente a un gen y sus diferentes expresiones descritas en sus alelos, o a una parte del cromosoma. Una variación de este tipo, al afectar a una parte del cromosoma, tiene varios alelos de varios genes involucrados por lo que agrupará un conjunto de variaciones de este tipo que están asociadas a alelos de un gen respectivamente.

El rango de estas variaciones es el siguiente:

- *Génico*: Cuando la variación esta asociada con un solo gen.

- *Cromosómico*: Cuando la variación esta asociada con varios genes y por lo tanto afecta a partes del cromosoma.

3. *Efecto*: Este criterio clasifica a las variantes alélicas de acuerdo con los efectos que estas producen en el fenotipo; además determina si la variación es Mutante (un efecto negativo o patológico); si es Polimorfismo neutral (un efecto neutral); o si es una Consecuencia desconocida (un efecto que aun no esta definido)

   Si la variación es mutante puede presentar los siguientes efectos en el proceso de la construcción de proteínas:

   - *Splicing*: Cuando la mutación afecta el proceso de Splicing.

   - *Regulación*: Cuando la mutación afecta la regulación de la expresión de un gen.

   - *Missense*: Cuando la mutación consiste en que un nucleótido es cambiado dando como resultado un codón que codifica para diferente amino acido y por lo tanto es producida una proteína no funcional

   - *Otros*: Detalles de variantes alélicas que tienen un efecto en el fenotipo pero que no se puede clasificar en ninguna de las tres especializaciones anteriores

En resumen puede decirse que las variaciones presentes en la información genética que describen las secuencias de ADN, son variaciones que se observan cuando los genes se expresan, es por eso que las variaciones pueden identificarse en los alelos cuando se compara un alelo natural determinado contra otro alelo del mismo gen. Las diferencias encontradas entre los alelos producirán características diferentes entre un organismo y

otro. Algunas de estas características pueden ser patológicas, sin embargo, algunas de estas características hacen que los seres humanos seamos diferentes unos de otros.

2.3 *Trabajos Relacionados*

Al realizar una búsqueda exhaustiva de trabajos relacionados con la aplicación de técnicas de modelado conceptual en temas de biología molecular es muy interesante notar que no existen demasiadas referencias relevantes donde el genoma humano, o algún otro genoma, pueda ser visto como un Sistema de Información desde una perspectiva de modelado conceptual.

Una de las contribuciones mas relevante en este campo es sin duda la propuesta hecha por Paton et al. [19]. Este trabajo es una referencia importante para la especificación del genoma humano en el esquema conceptual propuesto en este trabajo de investigación. Ya que es considerada el punto de partida para la construcción de dicho esquema. La propuesta de Paton se presenta una colección de modelos para datos genómicos. Estos modelos describen elementos involucrados en los procesos de transcripción y traducción que conllevan a la construcción de proteínas y también describen los efectos generados a partir de la ejecución de los procesos.

Como se ha mencionado anteriormente, este trabajo de investigación extiende las ideas establecidas en la colección de modelos realizada por Paton y propone un esquema conceptual completo que puede ser visto como un repositorio conceptual de información genómica. Con esta idea se extienden los objetivos de la propuesta de Paton de modelar solo una parte de la información genómica.

Otro enfoque para modelar genomas se observa en la iniciativa e-Fungi [22, 23] fundamentada en un análisis sistemático comparativo de genomas de hongos. e-Fungi es una base de datos que integra una información de mas de 30 genomas de hongos. Proporciona a los biólogos un poderoso recurso de estudios comparativos entre un gran rango de genomas. La iniciativa e-Fungi muestra una forma clara de cómo la información biológica puede ser explotada lo que lleva a pensar que esto puede instanciarse a información referente al genoma humano. Por lo que construir un

esquema conceptual para el genoma humano como repositorio conceptual de información genómica puede contribuir directamente para la explotación de información de genomas de los seres humanos.

Un ejemplo interesante sobre la aplicación de técnicas de modelado conceptual en la biología molecular es sin duda la propuesta de Ram [24] que consiste en la especificación de un modelo de la proteína. Esta propuesta aborda una parte muy específica de un genoma, sin embargo, esto permite pensar que es necesario realizar un trabajo de modelado donde se incluyan todas las partes del genoma. Por lo tanto la propuesta de Ram ha sido bastante útil para la construcción del esquema conceptual del genoma humano completo presentado en este trabajo de investigación.

Adicionalmente, un conjunto relevante de implementaciones bioinformáticas están desarrolladas en mayor o menor grado sobre técnicas de modelado conceptual y estas implementaciones han sido aceptadas favorablemente. Un ejemplo de esto es el trabajo realizado por Kevin Garwood et al. [20] el cual tiene un enfoque MDA (Model-driven Architecture) [11] para la generación parcial de interfaces de usuario que sirvan para buscar y navegar repositorios de datos biológicos. Este trabajo demuestra que los esquemas conceptuales pueden ser usados para producir muchas aplicaciones en el futuro. De nuevo, comparado con el trabajo de investigación realizado, esta propuesta enuncia el uso de técnicas de modelado conceptual enfocada sobre una parte muy especifica del proceso de desarrollo de software (el diseño interfases de usuario), mientras que modelar el genoma humano conceptualmente permitirá aprovechar muchas más de las ventajas de los SI tradicionales.

Otro ejemplo del uso de modelado conceptual en trabajos bioinformáticos es la propuesta de Erich Bornberg-Bauer and Norman W. Paton [21] que esta enfocada la búsqueda y recuperación de la información biológica disponible. Es importante mencionar que la información biológica disponible actualmente no es homogénea en cuanto a sus formatos y estructuras. Además el vocabulario y la semántica de los datos es diferente para cada repositorio de datos biológicos lo que provoca que su búsqueda y recuperación de información sean una tarea difícil. Sin embargo, Bornberg-Bauer y Paton explican en su propuesta que el modelado conceptual de la información involucra la construcción de modelos independientemente de su implementación que permite

conceptualizar las propiedades estructurales de los datos. Modelan las estructuras de datos con los lenguajes de modelado conceptual mas utilizados: UML (Unified Modelling Language) [17] y ER (Entity–Relationship). Al considerar la forma en como ellos modelaron las estructuras de los datos bioinformáticos disponibles, se observa que el modelado conceptual debe realizarse a un nivel mas alto de abstracción puesto que las fuentes de datos aumentarán o disminuirán a partir de la investigación que se realice en biología molecular.

Estos trabajos son algunos de los ejemplos existentes acerca del uso de modelado conceptual en aplicaciones relacionadas con trabajos bioinformáticos. Esto permite observar que el modelado conceptual es un enfoque efectivo para impulsar la investigación en biología. Por lo tanto la intención de este trabajo de investigación es lograr el entendimiento del genoma humano a través de la especificación de los conceptos relevantes en un esquema conceptual. Con esto será posible almacenar los contenidos correctos para manejarlos eficientemente y entender las relaciones que existen entre el fenotipo (manifestación externa de características en los humanos) y el genotipo (su correspondiente código genómico).

3 El esquema conceptual del genoma humano (ECGH)

Resumen: El presente capitulo incluye la descripción completa del esquema conceptual del genoma humano (ECGH) (ver Anexo 1), que ha sido diseñado aplicando técnicas de Modelado Conceptual, específicamente diagramas de clases, a un dominio determinado: el genoma humano. La descripción es realizada de forma narrativa con la intención de asociar los elementos del dominio con las clases, atributos y asociaciones diseñadas en el ECGH. Sin embargo, en el anexo 2 de esta memoria, se incluye una descripción menos detallada. Los elementos del ECGH se presentan organizados en tablas con el objetivo de que puedan ser identificados con mayor facilidad. Este capitulo inicia con una introducción sobre la aplicación de Modelado Conceptual en los Sistemas de Información y posteriormente se menciona el contenido de las secciones del capitulo.

3.1 *Introducción*

En la moderna Ingeniería de Software la aplicación de técnicas de modelado conceptual ha permitido crear sistemas de mayor calidad debido a que la descripción y entendimiento del dominio del problema se realiza antes de llevar a cabo las representaciones software (implementaciones) correspondientes.

Los dominios a los que se han aplicado las técnicas de modelado conceptual son muchos, y probablemente el ámbito de aplicación más conocido y trabajado sea el relacionado con los Sistemas Organizacionales [16]. Cuando se exploran nuevos dominios de aplicación, aparecen desafíos nuevos que dependen de la complejidad del dominio. El dominio del Genoma Humano es uno de estos dominios, en él llama la atención el hecho de que no se hayan aprovechado las ventajas que ofrece el modelado conceptual, para conseguir su correcta y completa especificación.

Los conceptos biológicos en este dominio pueden ser descritos por medio de un esquema conceptual que permita un mejor entendimiento del genoma humano: las

relaciones estructurales y funcionales de los genes con el proceso de traducción del ADN y los procesos de transcripción implicados en la síntesis de proteínas.

Tradicionalmente, los desarrollos en el campo de la Bioinformática han estado orientados a la resolución de problemas algorítmicos y computacionales, subestimando la importancia que tiene para el área la existencia de sistemas de información genómicos que sean fiables y que estén preparados para asumir los continuos cambios a los que está sometido este campo de investigación.

Los cambios experimentados por un SI pueden ser debidos a cambios en sus requisitos de uso o bien a la incorporación de nuevos requisitos. Un SI se concibe como un conjunto de elementos relacionados entre sí, que deben estar perfectamente integrados para evitar inconsistencias durante su uso. La existencia de un esquema conceptual es fundamental para evitar estos efectos no deseados; ya que asegura la correcta integración entre todos los componentes del sistema. Sin esta integración el SI comienza a estar fragmentado y el mantenimiento de una vista global y su consistencia empiezan a ser tareas complicadas. Usualmente, la responsabilidad de mantener la consistencia del sistema es asumida por actores humanos, y desafortunadamente la intervención humana es muy costosa y es fuente continua de errores que generan sistemas de baja calidad [18].

Además las propiedades básicas de evolución y modularidad de los SI [18] que ayudan al manejo de los cambios, se ajustan perfectamente a la naturaleza evolutiva de la investigación en biología molecular.

En la primera sección de este capítulo se define un esquema conceptual donde se representan los conceptos básicos del genoma humano utilizados por los biólogos cuando abordan procesos relacionados con los análisis genéticos. En este esquema la descripción realizada tiene la intención de identificar los conceptos relevantes que están involucrados en la estructura y funcionamiento del organismo humano, desde el ADN hasta la producción de proteínas que mantienen la estructura y actividad celular en el organismo humano.

En la siguiente sección del capítulo se describe como el esquema conceptual del genoma humano ha ido evolucionando a medida que se ha incorporado nuevo conocimiento sobre el dominio; también se menciona la forma en como las definiciones poco precisas de los elementos del dominio representan un desafío al momento de clasificarlos en conceptos. Sin embargo, a medida que la investigación en biología molecular avanza se hacen definiciones más precisas de los elementos presentes en el genoma humano, esto también permite que el ECGH evolucione, ya que el esquema debe ser siempre una representación completa y correcta del dominio.

Finalmente en la última sección se menciona la necesidad de adecuar este esquema conceptual para que el sistema de información que describe pueda interactuar con los recursos de datos biológicos que actualmente están publicados. Con estas adecuaciones se obtiene un nuevo esquema conceptual a partir del cual pueden realizarse implementaciones que convivan amigablemente con los recursos publicados actualmente. Esto origina la definición del actual esquema conceptual como esquema "ideal" y la creación de un esquema "real" que incluya las adecuaciones necesarias para interactuar con lo que la información disponible hoy en día.

3.2 *Descripción del esquema conceptual*

El esquema conceptual del genoma humano (ECGH) que se presenta, se corresponde con un estado intermedio y estable del conocimiento actual sobre el dominio. El esquema está preparado para su evolución y para la incorporación de nuevos conceptos. El ECGH se describe usando el estándar UML concretamente se han utilizado los diagramas de clase como lenguaje de modelado [17]. Para conseguir una descripción mas precisa se incluyen expresiones en OCL [25] para expresar algunas propiedades (restricciones de integridad y leyes de derivación).

El ECGH presentado en Fig. 22 esta divido en tres vistas principales para facilitar su diseño, visualización y comprensión (ver Anexo 1). Estas vistas son las siguientes:

1. *Gene–Mutation View:* En esta vista se modelan los conceptos utilizados para describir la estructura interna de los genes. Se describen las secuencias de ADN

de alelos naturales y de alelos variantes, así como la descripción de las mutaciones de éstos últimos.

2. *Transcription View:* En esta vista se modelan los conceptos involucrados en el proceso de transcripción y de síntesis de proteínas.

Las vistas *Gene-Mutation View* y *Transcription View* del esquema, tienen como objetivo registrar información genérica sobre genes, mutaciones y procesos de transcripción, según el estado de este conocimiento aceptado por la comunidad científica.

3. *Genome View:* En esta vista se modelan información relativa a la composición estructural del genoma de individuos concretos.

La vista *Genome View* del SI, tiene como objetivo registrar información genética de futuros pacientes o clientes del sistema.

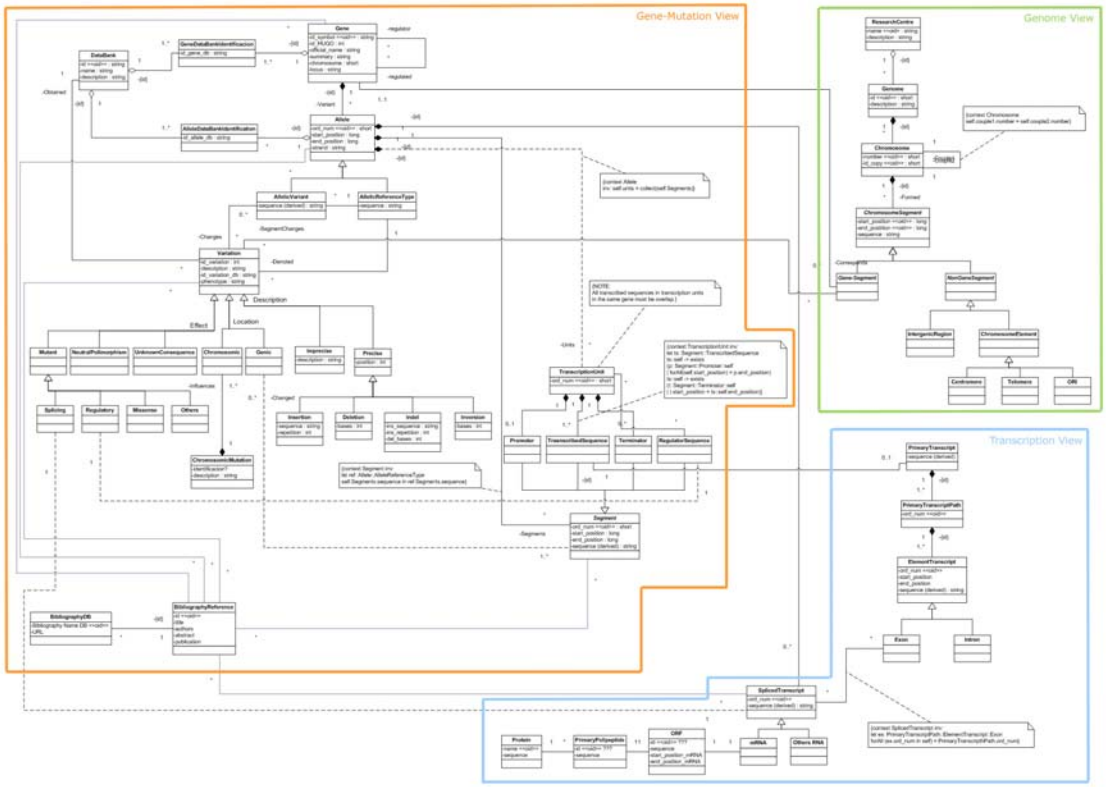


Fig. 22 Esquema Conceptual para la especificación del genoma humano.

A continuación se describe cada vista del ECGH con un alto nivel de detalle. Esta descripción logra una vinculación estrecha entre la aplicación de técnicas de modelado conceptual y el dominio involucrado: el genoma humano.

3.2.1 Gene – Mutation View

En la vista *Gene-Mutation* presentada en Fig. 23 está representado todo el conocimiento acerca de los genes, su estructura y sus variantes alélicas. Las clases principales en esta vista son la clase *Gene* y la clase *Allele*.

Descripción de la vista:

La clase *Gene* modela el concepto de gen genérico independiente de los ejemplos que de él se encuentran en las bases de datos biológicas en las que la muestra del gen pertenece a un individuo. La descripción del gen se realiza desde un enfoque estructural.

En esta clase están presentes atributos como *id_symbol* que representa un código alfanumérico (Por ejemplo NF1 para el gen de la neurofibromatosis) y que al mismo tiempo es el identificador de la clase. Este código coincide con el código que HGNC (Human Genome Nomenclature Committee) [8] asigna a cada gen; esta clase también contiene el atributo i*d_HUGO* que es un código numérico que representa el código universal para los genes de acuerdo con el HGNC. Otro atributo es *oficial_name* el cuál almacena el nombre común con el que se denomina al gen. El atributo *summary* que contiene un resumen del gen, extraído de la descripción dada por la base de datos del NCBI [6]. El atributo *choromosome* que representa el número de cromosoma donde el gen está localizado. Finalmente el atributo *locus* que indica la posición del gen dentro del cromosoma esta información se extrae también del NCBI.

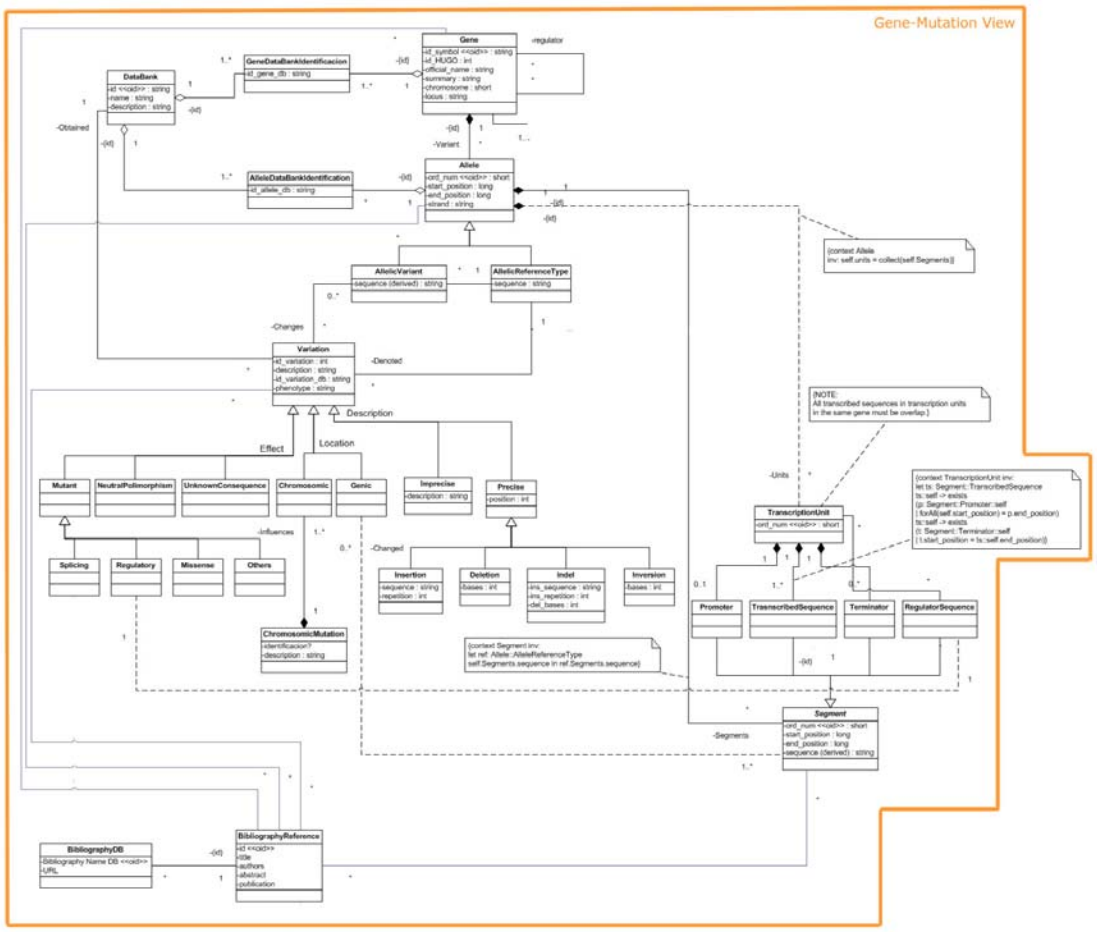


Fig. 23 Vista Gene-Mutation del ECGH.

Esta clase presenta una relación de recursividad denominada *regulator/regulated* con cardinalidad (1..N:1..N) para describir el hecho de que un gen puede regular otros genes y al mismo tiempo ser regulado por uno o varios genes.

Una de las principales clases incluidas en el ECGH es la clase *Allele* que permite modelar un gen a partir de su expresión. Por lo tanto la clase *Allele* corresponde con instancias de un gen genérico para describir las formas alternativas que un gen presenta cuando se expresa. Esta es la clase más importante en el esquema ya que toda la información depende de ella. A partir de esta clase se describe información relevante sobre los alelos por ejemplo las bases de datos públicas que describen secuencias alélicas, tipos de alelos (referencias y variaciones), las secuencias de ADN correspondientes y el producto final del proceso de transcripción. Los atributos que posee esta clase son el atributo *ord_num* que es el identificador de la clase en el sistema,

los atributos *start_position* y *end_position* que describen el número de nucleótido en el cual se establecen la posición inicial y final que ocupa del alelo respecto a la secuencia de ADN del cromosoma y el atributo *strand* que almacena el valor *plus* o *minus* de acuerdo con la hebra de ADN en la cuál se ubica el alelo.

Existe una relación de composición que va desde la clase *Allele* a la clase *Gene.* Esta asociación es denominada *Variant* y tiene una cardinalidad (1..1: 1..N) para describir que un alelo determinado solo puede ser instancia de un gen genérico. Por ejemplo, El gen NF1 tiene varios alelos, sin embargo, cada uno de estos alelos solo pertenece al gen NF1, es decir, un alelo no puede describir la misma secuencia de ADN para dos genes diferentes.

Las referencias de las fuentes externas de las que se obtiene la información juegan un papel fundamental en el ECGH. La clase *DataBank* agrupa características de las bases de datos públicas que se utilizan para obtener la información sobre los alelos. Los atributos de esta clase son el atributo *id* que funciona como el identificador de la fuente de datos, el atributo *name* para almacenar el nombre determinado y el atributo *description* que contiene una descripción acerca de la fuente de datos.

La clase *GeneDataBankIdentification* representa la identificación de un gen genérico en las diferentes bases de datos públicas. Su único atributo es *id_gene_db* para almacenar el identificador que le otorga la fuente de datos externa a un determinado gen. Por otra parte la clase *AlleleDataBankReference* representa la identificación de un alelo en las diferentes bases de datos públicas. También tiene un único atributo *id_allele_db* que almacena el identificador que le otorga la fuente de datos externa a un determinado alelo.

La clase *DataBank* presenta una relación de agregación hacia la clase *AllelicDataBankIdentification* y otra hacia la clase *GeneDataBankIdentification* para agrupar a los diferentes alelos o genes que un determinado banco de datos ha identificado.

De la misma forma las clases *Allele* y *Gene* están asociadas con las clases *AlleleDataBankIdentification* y *GeneDataBankIdentification* respectivamente. Esto con

la intención de relacionar al gen o al alelo con su fuente de información externa y la forma en como esta fuente lo identifica.

En cuanto a las variantes alélicas toda la información relevante esta descrita en el esquema a partir de la especialización de la clase *Allele* en las clases *AllelicVariant* y *AllelicReferenceType*. Esta especialización expresa el hecho de que las variantes alélicas se consideran variantes con respecto a un alelo determinado considerando un alelo de referencia en la correspondiente base de datos.

La clase *AllelicReferenceType* representa los alelos que se utilizan como referencias para describir las variantes alélicas. Esta clase tiene el atributo *sequence* que almacena la secuencia de nucleótidos del alelo; de igual forma la clase *AllelicVariant* representa a los alelos que son considerados variantes alélicas respecto a un alelo de referencia determinado, por ello, la clase *AllelicVariant* tiene una relación de asociación respecto a la clase *AllelicReferenceType* con una cardinalidad (1..1:0..N) con esto se describe el hecho de que una variante alélica tiene siempre un único alelo de referencia mientras que un alelo de referencia puede tener muchas variantes alélicas asociadas. El atributo *sequence* de la clase *AllelicVariant* tiene un comportamiento especial, ya que su valor será derivado si todas las variaciones asociadas al alelo variante son precisas (conocidas) de tal forma que el valor del atributo se derivará utilizando la descripción de estas variaciones y el atributo *sequence* de la instancia de *AllelicReferenceType* asociada. Cuando las variaciones asociadas a *AllelicVariant* no sean precisas el valor para su atributo *sequence* no será derivado.

Para tener agrupadas las posibles variaciones asociadas a un alelo de referencia, independientemente de si se conoce con precisión el alelo variante, se incluye en el esquema la clase *Variation*.

La clase *Variation* tiene el atributo *id_variation* que funciona identificador de la clase; el atributo *description* para incluir información acerca de la variante alélica; también posee el atributo *id_variation_db* para identificar la base de datos externa en la que esa variación esta registrada; y el atributo *phenotype* para almacenar la especificación fenotípica del efecto que produce la variación.

La clase *Variation* presenta la asociación *Denoted* respecto a la clase *AllelicReferenceType* para describir la agrupación de las variaciones alélicas asociadas a un alelo de referencia. Esta asociación tiene la cardinalidad (1..1;N..1) de tal forma que una variación tiene siempre un único alelo de referencia mientras que un alelo de referencia tiene una o muchas variaciones asociadas. La clase *Variation* también presenta la asociación *Changes* respecto a la clase *AllelicVariant* para describir la relación que existe entre la variación y su secuencia alélica variante conocida. Esta asociación tiene la cardinalidad (0..N:1..N) de tal forma que una variación puede tener asociado uno o muchos alelos variantes, cuando esta variación es precisa, o bien puede no tener asociados alelos variantes. Por su parte un alelo variante siempre tendrá asociada una o muchas variaciones.

Las variaciones alélicas, independientemente del conocimiento preciso de la secuencia del alelo variante, se encuentran registradas en bases de datos externas. Esto se describe en el ECGH con la asociación *Obtained* de la clase *Variation* respecto a la clase *DataBank* para conocer características de la fuente externa de la que se obtiene la variación.

La clasificación de las variaciones alélicas instancias de la clase *Variation* se representa en el ECGH a través de una especialización de la clase *Variation*. Esto permite clasificar las variantes alélicas en diferentes criterios que se describen en tres especializaciones. La primera especialización (*Location*) representa el alcance de la variación, es decir, si la variación afecta únicamente a un gen o a una parte del cromosoma. La segunda especialización (*Description*) representa el nivel de conocimiento acerca de la variación, es decir, si la variación es precisa o imprecisa. Finalmente la tercera especialización (*Effect*) clasifica las variantes alélicas de acuerdo con los efectos que estas producen en el fenotipo; además determina si la variación es Mutante (un efecto negativo o patológico); si es Polimorfismo neutral (un efecto neutral); o si tiene una Consecuencia desconocida (un efecto que aun no esta definido).

Las clases especializadas de *Variation* en la especialización *Location* son: *Genic*, cuando la variación afecta a un solo gen; y *Chromosomic*, cuando la variación afecta partes del cromosoma; también está presente la clase *ChromosomicMutation* que agrupa las variantes alélicas que están presentes en la variación cromosómica y que afectan a

uno o muchos genes dentro del mismo cromosoma. Esta clase tiene los atributos *identification* para la identificación de esta variación y *description* para incluir una descripción de la variación. Al mismo tiempo la clase *ChromosomicMutation* tiene la relación de composición "*Influences*" respecto a la clase *Chromosomic* con una cardinalidad (1..N: 1..1) para describir el hecho de que una variación alélica que afecta partes del cromosoma estará compuesta por una o muchas variaciones que afecta a varios genes y a su vez cada una de estas variaciones alélicas localizadas y clasificadas como participantes de una variación cromosómica solo podrán estar asociadas a una determinada variación cromosómica. El número de instancias de la clase *Chromosomic* asociadas a la clase *ChromosomicMutation* dependerá del número de variantes alélicas de diferentes genes que intervengan en la variación cromosómica.

Las clases especializadas de *Variation* en la especialización *Description* son: *Impresice*, cuando los detalles de la variación son desconocidos; y *Precise*, cuando los detalles de la variación son conocidos, es decir, se conoce la posición donde ocurre la variación en la secuencia de nucleótidos. La clase *Impresice* tiene el atributo *description* que almacena una descripción de la variación. La clase *Precise* tiene el atributo *position* que indica la posición de la ocurrencia de la variación.

Las variaciones precisas pueden ser clasificadas para el tipo: *Insertion*, *Deletion*, *Indel*, *Inversion* que describen cada uno de los diferentes tipos de variaciones.

La clase *Insertion* describe los detalles de las variantes alélicas que presentan inserciones de nucleótidos respecto al alelo de referencia. Tiene el atributo *sequence* que almacena la secuencia de nucleótidos que se ha insertado; y el atributo *repetition* que almacena el numero de veces que se ha insertado dicha secuencia.

La clase *Deletion* describe los detalles de las variantes alélicas en las que ciertos nucleótidos han sido borrados en una posición específica del alelo de referencia. Tiene el atributo *bases* que almacena el número de nucleótidos o bases que han sido borrados.

La clase *Indel* describe los detalles de las variantes alélicas en las que ocurrió un borrado de nucleótidos y una inserción de uno o varios nucleótidos un determinado numero de veces en la misma posición de la cadena, esta clase tiene los atributos

ins_sequence, que describe la secuencia de nucleótidos que ha sido insertada; *ins_repetition*, que almacena el numero de veces que se ha insertado la secuencia; y el atributo *del_bases*, que almacena el numero de nucleótidos que han sido borrados.

La clase *Inversion* describe los detalles de las variantes alélicas en las que ocurre una inversión en el orden de los nucleótidos respecto a la secuencia de referencia. Tiene el atributo *bases* para almacenar el número de nucleótidos que se han invertido.

La posición concreta de una variación (inserción, borrado, inserción-borrado o bien inversión) respecto al alelo de referencia se almacena en el atributo *position* de la clase *Precise*.

Las clases especializadas de *Variation* en la especialización *Effect* son: *Mutant*, *NeutralPolimorphism* y *UnknownConsequence* que son la clasificación de las variantes alélicas de acuerdo con el efecto que producen en el fenotipo: mutante (patológica), polimorfismo neutral y consecuencias desconocidas.

La clase *Mutant* describe las variantes alélicas que provocan un determinado efecto en el proceso de síntesis de proteínas. Para describir estos efectos la clase *Mutant* es especializada en las clases *Splicing*, cuando la mutación afecta el proceso de splicing; *Regulatory*, cuando la mutación afecta la regulación de un gen; *Missense*, cuando la mutación consiste en que un nucleótido es cambiado dando como resultado un codón que codifica para diferente amino acido y por lo tanto es producida una proteína no funcional; *Others*, para almacenar detalles de variantes alélicas que tienen un efecto en el fenotipo pero que no se puede clasificar en ninguna de las tres clases anteriores.

La clase *NeutralPolimorphism* describe las variantes alélicas que no afectan al fenotipo, por ejemplo en una mutación missense cuando un determinado nucleótido es cambiado y provoca la presencia de un nuevo amino acido que es similar en propiedades químicas al amino acido que debería estar presente lo que conlleva a que la proteína pueda funcionar normalmente.

La clase *UnknownConsequence* describe las variantes alélicas de consecuencias que no son conocidas todavía, es decir, se sabe que son mutaciones pero los efectos que provocan aun no se conocen.

Además de almacenar información sobre las fuentes externas de donde provienen los datos es importante asociar referencias bibliográficas que respalden la información almacenada en el SI. Debido a esto se incluye en el esquema la clase *BibliographyDB* que representa características relevantes de las bases de datos de referencias bibliográficas de las que se obtiene la información. Los atributos de esta clase son *BibliographyNameDB* para almacenar el nombre de la base de datos y el atributo *URL* para almacenar la dirección electrónica de esta base de datos; para almacenar información sobre las publicaciones relevantes contenidas en las bases de datos bibliográficas se incluye en el esquema la clase *BibliographyReference*. Esta clase tiene los atributos *id*, que funciona como identificador de la clase; el atributo *title* que almacena el titulo del artículo científico; el atributo *authors*, que almacena los nombres de los autores del artículo científico determinado; el atributo *abstract*, que almacena el resumen del articulo; y el atributo *publication*, que almacena el articulo completo.

En esta vista también se modela la estructura interna de los alelos con la finalidad de describir la forma en como cada uno de los elementos de esta estructura alélica participa en el proceso de transcripción del ADN.

En principio se define la clase *Segment* que representa un segmento alélico que posee una secuencia significativa e indivisible de ADN. Esta clase tiene el atributo *ord_num*, que funciona como identificador de la clase y distingue a los segmentos alélicos; el atributo *start_position*, para almacenar la posición inicial del segmento alélico dentro del cromosoma; el atributo *end_position*, para almacenar la posición final del segmento alélico dentro del cromosoma; el atributo *sequence*, para almacenar la secuencia de nucleótidos delimitada por los atributos *start_position* y *end_position*. El valor de este atributo es un valor derivado que se obtiene como una subsecuencia de la secuencia descrita para el alelo con el que el segmento alélico está asociado. De tal forma que el valor del atributo *sequence* para una instancia de la clase *Segment* es una subsecuencia que se deriva del valor del atributo *sequence* de una instancia determinada de la clase *AllelicReferenceType*, si el segmento corresponde a un alelo de referencia; o de una

instancia determinada de la clase *AllelicVariant*, si el segmento corresponde a una variante alélica. Cualquiera de estas dos instancias como especialización de la clase *Allele,* por lo que esta instancia especifica de la clase *Allele* esta asociada con la instancia de la clase *Segment.* Además para la especificación del valor de este atributo debe ser considerada la delimitación que indiquen los valores de los atributos *start_position* y *end_position* de la instancia de la clase *Segment*.

Para especificar que los segmentos alélicos solamente deben estar asociados con su alelo de referencia correspondiente se incluye la siguiente restricción de integridad:

"The allele segments are associated only with their reference allele."

La expresión en OCL es la siguiente:

context Segment inv:
let ref: Allele::AlleleReferenceType
self.Segments.sequence in ref.Segments.sequence

Para describir la función de un segmento alélico en el proceso de transcripción, la clase *Segment* se especializa en las clases *Promoter*, *TranscribedSequence*, *Terminator* y *RegulatorSequence*. La clase *Promoter*, describe la secuencia de ADN que marca el inicio del proceso de transcripción; la clase *TranscribedSequence*, describe la secuencia de ADN transcrita por el ARN polimerasa; la clase *Terminator*, describe la secuencia de ADN que marca el fin del proceso de transcripción; la clase *RegulatorSequence*, describe un segmento alélico que contiene las secuencias de nucleótidos de las funciones de regulación de uno o mas procesos de transcripción. Es importante mencionar que esta clase no forma parte de la estructura del alelo, por tanto su identificación y su secuencia no están contenidas en dicho alelo.

Es posible que la ubicación del promotor de una unidad de transcripción no este localizado en la secuencia que describe el alelo. Sin embargo, si la secuencia de este promotor es conocida, todas las unidades de transcripción que este promotor tenga asociadas se inician en la misma posición, esta posición será la posición final de la secuencia del promotor. De la misma forma la secuencia del terminador de la unidad de

transcripción se iniciará en la posición final de la secuencia que describa la unidad de transcripción. Para expresar esta propiedad en el esquema se incluyen la siguiente restricción de integridad:

"All TranscribedSequence associated with the same TranscriptionUnit start in the same position. If the Promoter exists in this TranscriptionUnit, all these TranscribedSequeces start in the final position of this Promoter. If the Terminator exists in this TranscriptionUnit, the final position of the TranscribedSequence is the start position to this Terminator."

La expresión de esta restricción en OCL es la siguiente:

context TranscriptionUnit inv:
let ts: Segment::TranscribedSequence
ts::self -> exists (p: Segment::Promoter::self | forAll(self.start_position) = p.end_position)
ts::self -> exists (t: Segment::Terminator::self | t.start_position = ts::self.end_position)

A partir de los distintos tipos de segmento se define la unidad de transcripción. La clase *TranscriptionUnit* que representa a la unidad de transcripción. Esta clase tiene el atributo *ord_num* que funciona como identificador para diferenciar las unidades de transcripción presentes en el mismo alelo. Esta clase tiene una relación de composición respecto a las clases *Promoter*, *TranscribedSequence* y *Terminator* con esto se describe en el esquema la parte estructural de una unidad de transcripción. Puesto que la secuencia del promotor puede o no conocerse, la cardinalidad de la relación de composición desde la clase *TranscriptUnit* a la clase *Promoter* tiene una cardinalidad (0..1:1:1) para especificar que una unidad de transcripción puede tener o no una secuencia conocida para su promotor pero siempre tendrá un único promotor mientras que una secuencia de promotor conocida podrá ser promotor de varias secuencias transcribibles pero solo se almacenará una sola vez; la relación de composición desde la clase *TranscriptUnit* hacia la clase *TranscribedSequence* tiene una cardinalidad (1..N:1:1) para describir que una unidad de trascripción tiene una o muchas secuencias transcribibles asociadas pero una secuencia transcribible solo puede estar asociada con una unidad de transcripción; la relación de composición desde la clase *TrascriptUnit* hacia la clase *Terminator* tiene una cardinalidad (0..N:1..1) que indica que una unidad de transcripción puede estar asociada con varias o incluso ninguna secuencia de

terminación mientras que una secuencia de terminación conocida será almacenada una sola vez.

La clase *Allele* tiene una relación de composición hacia la clase *TranscriptionUnit*, esta asociación es dependiente ya que en un alelo siempre existen una o varias unidades de transcripción, este concepto se especifica en el esquema con la descripción de la cardinalidad de la relación entre estas clases (1..N:1..1) que muestra claramente como los alelos están compuestos por varias unidades de transcripción mientras que una unidad de transcripción solo pertenece a un alelo.

Puesto que las secuencias de las unidades de transcripción están compuestas a partir de las secuencias de los segmentos alélicos (subsecuencias de un alelo determinado), es necesario puntualizar el hecho de que las unidades de transcripción solo pueden estar asociadas a un determinado alelo el cual es el mismo alelo con el que están asociados los segmentos alélicos a partir de los cuales se componen dichas unidades de transcripción. Para describir este concepto en el esquema se incluye la siguiente restricción de integridad:

"An Allele must be associated only with transcription units than are composed by allelic segments associated with same allele."

La expresión de esta restricción en OCL es la siguiente:

context Allele
inv: self.units = collect(self.Segments)

Como se ha mencionado anteriormente, a partir de la especialización de la clase *Segment*, también es necesario especificar lo que se ha conceptualizado en el esquema como secuencia reguladora. Para esto la clase *TranscriptionUnit* presenta una asociación con la clase *RegulatorSequence* para describir que una unidad de transcripción puede tener muchos segmentos reguladores que al mismo tiempo comparte con otras unidades de transcripción de distintos genes.

La clase *Genic*, como especialización jerárquica de la clase *Variation*, presenta una asociación dependiente hacia la clase *Segment*. La intención de esta asociación es conocer el segmento alélico en el que se produce la variación génica.

De igual forma la clase *Regulatory*, como especialización jerárquica de la clase *Variation*, presenta una asociación dependiente hacia la clase *RegulatorSequence*. La intención de esta asociación es conocer la secuencia reguladora en la que se produce la variación.

Las clases *Gene*, *Allele*, *Variation* y *Segment* presentan una asociación respecto a la clase *BibliographyReference* para describir el hecho de que la información de las instancias de cada una de estas clases esta avalada bibliográficamente.

3.2.2 Transcription View

En la vista *Transcription* presentada en Fig. 24 se modela el conocimiento asociado a los elementos básicos implicados en el proceso de síntesis de proteínas.

Descripción de la vista:

La primera clase presentada en esta vista es la clase *PrimaryTranscript*, que representa la copia transcrita de ADN a ARN de la secuencia transcribible, clase *TranscribedSequence*. Este es uno de los conceptos que relaciona a la vista *Gene-Mutation* con la vista *Transcription*. Para asociar la secuencia transcribible con los primarios transcritos se incluye en el ECGH una relación de asociación entre la clase *TranscribedSequence* y la clase *PrimaryTranscript* con cardinalidad (0..1:1..1) para describir el hecho de que una secuencia transcribible puede o no estar asociada con su correspondiente transcrito primario mientras que un transcrito primario siempre estará asociado a una secuencia transcribible. La clase *PrimaryTranscript* tiene el atributo *sequence* el cual tiene un valor derivado del atributo *sequence* de la clase *Segment* y que es obtenido a partir de la asociación de la clase *PrimaryTranscript* con la clase *TranscribedSequence*.

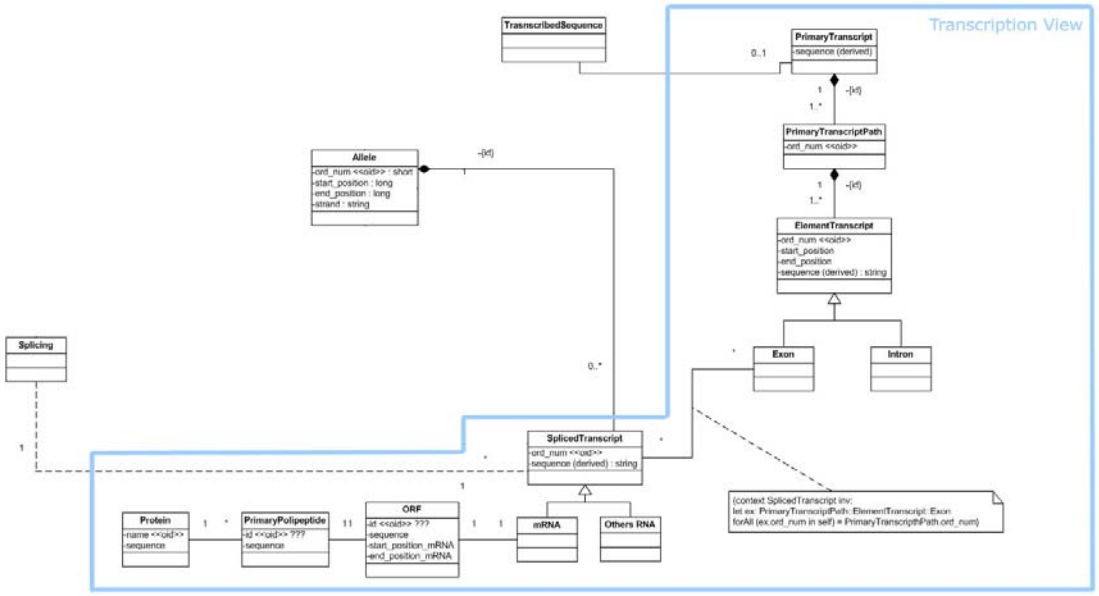


Fig. 24 Vista Transcription del ECGH.

Para modelar las diferentes particiones del primario transcrito se incluye en el esquema la clase *PrimaryTranscriptPath*. Esta clase representa las diferentes particiones, impulsadas por el factor splicing, que tiene un transcrito primario. Por lo tanto la clase *PrimaryTranscript* presenta una relación de composición hacia la clase *PrimaryTranscriptPath* con la cardinalidad (1..N:1:1) para describir que un primario transcrito esta asociado a una o muchas particiones, impulsadas por el factor splicing, mientras que una partición solo puede pertenecer a un primario transcrito.

La clase *PrimaryTranscriptPath* tiene el atributo *ord_num* que permite identificar una partición especifica de las diferentes particiones de un transcrito primario. Para representar en el esquema los elementos de transcripción (exones/intrones) presentes en una partición del transcrito primario se incluye la clase *ElementTranscript* hacia la cual la clase *PrimaryTranscriptPath* presenta una relación de composición con la cardinalidad (1..N:1:1) para describir que una partición esta integrada por uno o muchos elementos de transcripción mientras que un determinado elemento de transcripción solo pertenece a una partición especifica.

La clase *ElementTranscript* tiene el atributo *ord_num*, para identificar un fragmento específico de todos los fragmentos de las particiones; también tiene los atributos *star_position* y *end_position*, que delimitan la secuencia que describe el fragmento; y el atributo *sequence* que representa la secuencia del fragmento y que además es un valor derivado del atributo *sequence* de la clase *PrimaryTranscript*.

Para especificar en el ECGH los elementos de transcripción presentes en una partición determinada se especializa la clase *ElementTrascript* en las clases *Exon* e *Intron*, que representan a un exon o un intron respectivamente presentes en una partición.

Las diferentes combinaciones de los exones presentes en una determinada partición del primario transcrito se representan en el esquema con la clase *SplicedTranscript*. Por lo tanto la clase *Exon* presenta una asociación hacia la clase *SplicedTranscript* para describir la participación de un determinado exon en las diferentes combinaciones de exones. A partir de esta asociación es necesario especificar en el esquema que todos los exones asociados a una combinación determinada deben pertenecer a la misma partición de un primario transcrito específico. Por lo tanto se incluye en el ECGH la siguiente restricción de integridad:

"All exons associated with a SplicedTranscription belong to the same PrimaryTranscript partition."

La expresión en OCL es la siguiente:

context SplicedTranscript inv:
let ex: PrimaryTranscriptPath::ElementTranscript::Exon
forAll (ex.ord_num in self) = PrimaryTranscripthPath.ord_num

La clase *SplicedTranscript* tiene el atributo *ord_num* para identificar una combinación de las diferentes combinaciones de exones; también tiene el atributo *sequence* que tiene un valor derivado a partir de la combinación de las secuencias de cada exon presente en esa combinación.

La clase *Allele* presenta una relación de composición respecto a la clase *SplicedTranscript*. Esta relación tiene la intención de asociar a los exones producto de un proceso splicing con su alelo de referencia o bien con su variante alélica.

La clase *Splicing*, como especialización de la clase *Mutant* que es una especialización jerárquica de la clase *Variation*, presenta una asociación dependiente hacia la clase *SplicedTranscript*. La intención de esta asociación es conocer en que combinación de exones, producto de un proceso splicing, se produce la variación determinada como mutación splicing.

El resultado de las combinaciones de exones, los diferentes tipos de ARN, representados en la clase *SplicedTranscript* conlleva a diferenciar el ARNm (ARN mensajero), implicado en el proceso de la síntesis de proteínas, de cualquier otro tipo de ARN. Por lo tanto en el esquema se incluye la especialización de la clase *SplicedTranscript* en la clase *mRNA*, que representa al ARNm; y la clase *OthersRNA*, que representa a los otros tipos de ARN.

El ARNm contiene secuencias de nucleótidos que pueden codificar potencialmente una proteína, esto es conocido como ORF (Open Reading Frame). Este concepto se especifica en el ECGH con la creación de la clase *ORF* la cual se asocia a la clase *mRNA*. La clase ORF tiene el atributo *id* que funciona como identificador de la clase; el atributo *sequence*, que describe la secuencia codificante; y los atributos *start_position* y *end_position* que especifican la posición de la secuencia del ORF en el ARNm.

La estructura primaria de la proteína es la cadena de aminoácidos obtenida después de la traducción de un ORF. Este concepto se representa en el esquema con la clase *PrimaryPolypeptide* que tiene el atributo *id*, que funciona como identificador de la clase; y el atributo *sequence* que almacena la cadena de aminoácidos. La clase *PrimaryPolypeptide* esta asociada con la clase *ORF* con la cardinalidad (1..1:1..1) para describir el hecho de que un polipéptido primario solo esta asociado a un determinado ORF al mismo tiempo que un ORF solo da origen a un polipéptido primario. La cadena de aminoácidos sufre algunas transformaciones químicas y el resultado final es una proteína funcional. Esto se representa en el esquema con la clase *Protein* que tiene el atributo *name*, que funciona como identificador de la clase y al mismo tiempo describe

el nombre de la proteína; y el atributo *sequence*, que almacena la cadena de aminoácidos que describen una proteína. La clase *Protein* esta asociada a la clase *PrimaryPolypeptide* con la cardinalidad (1..N:1..1) para describir que una proteína puede estar formada por uno o muchos polipéptidos primarios pero un polipéptido primario solo participa en la formación de una proteína.

3.2.3 Genome View

En esta vista del ECGH presentada en Fig. 25. se describe un genoma completo de un humano determinado. Esta vista del esquema es muy interesante para aplicaciones futuras, considerando el momento en el que las tecnologías de secuenciación paralela masivas permitan la secuenciación completa de genomas individuales a precios bajos. Cuando esta información llegue a estar disponible será posible almacenarla en el esquema.

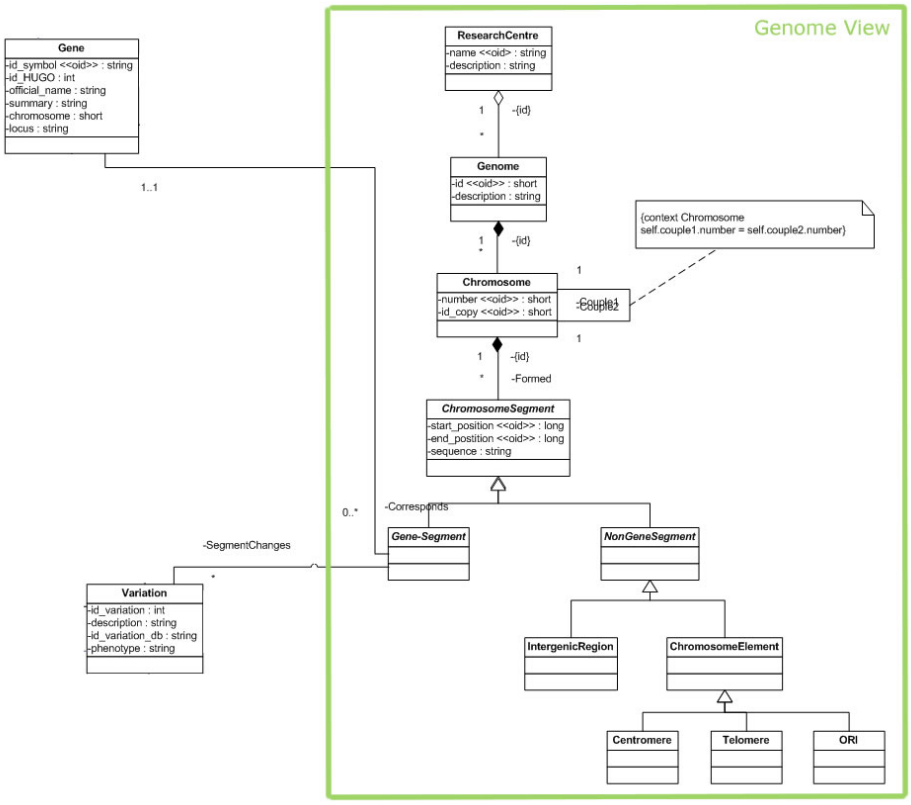


Fig. 25 Vista Genome del ECGH

Descripción de la vista:

La primera clase descrita en esta vista es la clase *ResearchCentre*, que representa el laboratorio o centro de investigación donde el genoma de un humano ha sido secuenciado. Esta clase tiene el atributo *name*, que funciona como identificador de la clase; y el atributo *description*; en conjunto estos atributos permiten almacenar el nombre y una descripción del laboratorio o centro de investigación que ha secuenciado el genoma de un humano determinado. Esta clase presenta una relación de agregación hacia la clase *Genome* con una cardinalidad (0..N:1..1) para describir el hecho de que un centro de investigación puede tener secuenciados genomas de diferentes humanos pero que una secuenciación determinada solo estará realizada por un centro de investigación o laboratorio específicamente.

Las clases *Genome* y *Chromosome*, representan un genoma secuenciado completamente por un centro de investigación determinado. La clase *Genome* tiene el atributo *id*, que

funciona como identificador de la clase y permite diferenciar los genomas secuenciados de los diferentes humanos; también tiene el atributo *description* para incluir información relevante sobre esta secuenciación. La clase *Chromosome*, representa a los cromosomas presentes en el genoma secuenciado. Tiene el atributo *number* para almacenar el número de cromosomas de tal forma que determine un cromosoma especifico para un genoma especifico; tiene el atributo *id_copy* para poder identificar a cual de las copias del par homologo se refiere ese cromosoma. En conjunto los atributos *number* e *id_copy* funcionan como identificador de la clase y permiten diferenciar a los cromosomas unos de otros así como a las diferentes copias del par homologo.

El concepto correspondiente al hecho de que un genoma esta compuesto por un conjunto de cromosomas esta representado en el ECGH por la relación de composición que presenta la clase *Genome* hacia la clase *Chromosome* con cardinalidad (1..N:1..1) para describir que un genoma secuenciado tiene asociados uno o muchos cromosomas mientras que un cromosoma esta asociado únicamente a un genoma secuenciado.

La clase *Chromosome* presenta la asociación recursiva *Couple* para describir el concepto de par homologo, lo que significa que cada célula humana tendrá dos cromosomas equivalentes (uno del padre y uno de la madre) con los mismos genes pero diferentes alelos para cada gen. Puesto que el número de cromosoma para cada una de las dos copias, presentes en el par homologo, es el mismo se establece la siguiente restricción de integridad en el esquema:

"The Couple 1 has the same chromosome's number than the chromosome's number of Couple 2."

La expresión de la restricción en OCL es la siguiente:

context Chromosome
self.couple1.number = self.couple2.number

A partir de la conceptualización de los cromosomas en el esquema puede establecerse el concepto de segmento cromosómico, el cuál será una subsecuencia formalmente

delimitada de la secuencia de ADN completa del cromosoma. Para describir esto se incluye en el ECGH la clase *ChromosomicSegment*, que representa a todos los segmentos que integran a un cromosoma. Esta clase tiene los atributos *start_position* y *end_position* que delimitan la subsecuencia que este segmento tendrá a partir de la secuencia completa del cromosoma. También tiene el atributo *sequence* que almacena la subsecuencia del cromosoma delimitada por las posiciones contenidas como valores en los atributos *start_position* y *end_position*. Por lo tanto la secuencia de ADN completa del cromosoma estará compuesta por la unión de todos los segmentos cromosómicos asociados a ese cromosoma determinado. Esto es representado en el esquema con la relación de composición *"Formated"* desde la clase *Chromosome* a la clase *ChromosomicSegment* con cardinalidad (1..N:1:1) para especificar que un cromosoma esta compuesto por uno o mas segmentos cromosómicos mientras que un segmento cromosómico solo pertenece a un cromosoma.

En el ECGH también se representa la estructura del cromosoma desde la idea de identificar dentro del cromosoma lo que es considerado como material génico o región codificante, que posibilita los procesos de transcripción y traducción que conllevan a la síntesis de proteínas; y lo que es considerado material no génico o región no codificante, que es todo lo que no influye en estos procesos. Este concepto se representa en el esquema con la especialización de la clase *ChromosomicSegment* en las clases *GeneSegment* y *NonGeneSegment*.

La clase *GeneSegmet* presenta la asociación *"Corresponds"* con la clase *Gene* de la vista *Gene-Mutation*. Esta asociación tiene la cardinalidad (1..1:0..N) para describir el hecho de que un segmento de la región codificante en un cromosoma determinado corresponde a un gen específicamente mientras que el gen, modelado genéricamente, puede tener asociados muchos o incluso ningún segmento de la región codificante del cromosoma. La intención de esta asociación es identificar con precisión lo que representa el segmento de región codificante, en este caso, determinar que representa un gen.

De igual forma la clase *GeneSegment* presenta la asociación *"SegmentChanges"* respecto a la clase *Variation* de la vista *Gene-Mutation* para representar el hecho de que en un segmento de región codificante, el cual pertenece a un gen específicamente,

pueden encontrarse muchas variaciones alélicas. La intención de esta asociación es determinar lo que ocurre en ese segmento, es decir, que variaciones pueden ser identificadas dentro de una instancia de la clase *GeneSegment* determinada.

A partir de la representación del concepto de la región no codificante, presente en un cromosoma, es posible identificar una serie de conceptos asociados a esta región. Para describir esto en el esquema la clase *NonGeneSegment* es especializada en las clases *IntergenicRegion*, que representa el espacio entre las secuencias de los genes; y *ChromosomeElement*, que describe elementos dentro de los cromosomas que representan secuencias que pueden identificarse y pueden ser consideradas como ORI, Centrómero, o Telómero. Para estructurar los diferentes elementos que están presentes en instancias de la clase *ChromosomeElement*, esta clase es especializada en las clases *Centromere*, *Telomere* y *ORI*. Esta especialización es realizada en el esquema con la intención de conservar la funcionalidad del cromosoma que no esta involucrada en la producción de proteínas.

3.3 *Evolución del esquema conceptual*

Utilizar técnicas de modelado conceptual en dominios poco convencionales, y mucho más desafiantes que los dominios comunes, implica que la forma tradicional de especificar los dominios en esquemas conceptuales sea abordada de manera diferente.

Al realizar un esquema conceptual para el genoma humano, es completamente observable que los conceptos relevantes en el dominio no pueden identificarse a partir de un conjunto de propiedades que los elementos tengan de manera estática sino que el conjunto de propiedades de los elementos se identifican a partir de las propiedades que adquieren dichos elementos cuando se comportan de cierta forma. Por ejemplo un gen no existe como tal sino se expresa. Es decir, el gen es definido a partir de la proteína para la que codifica. Las propiedades del gen se identifican en la formación de un polipéptido específico que es producto de la actividad celular en la que la información del ADN es leída [27]. Por lo tanto conceptualizar un gen ha sido una tarea desafiante.

Además de la forma especial de identificar clases de conceptos en este dominio, también ha sido importante considerar que todos los conceptos evolucionan

continuamente. Esta evolución de conceptos se debe a los avances que van sucediendo en la investigación en Biología Molecular. Como ejemplo puede mencionarse la visión básica y moderna de la definición de gen. Antes de que se descubriera el rol codificante del ADN, un gen era identificado con un rasgo fenotípico especifico; hoy en día, un gen se identifica a partir de conocer un polipéptido especifico y como su expresión es controlada [27].

Por lo tanto, al especificar conceptualmente el genoma humano, ha sido muy importante tener en cuenta que el esquema experimenta una constante evolución. Dicha evolución esta relacionada con la información que se adquiera a partir de conocer más el genoma humano, y también está determinada por la naturaleza evolutiva de los elementos del dominio.

En esta sección se presenta la evolución que ha experimentado el esquema conceptual del genoma humano (ECGH) reflejada en diferentes versiones. Es importante observar que la forma en como el ECGH ha hecho frente a esta evolución corresponde con el principio de que la existencia de un modelo conceptual permite la eficiencia de un Sistema de Información a partir de su eficiencia para manejar la modularidad y evolución del SI [18, 5].

La evolución del ECGH presentada inicia con la explicación de una propuesta de Norman Paton [19] que es el punto de partida para la construcción del ECGH. Posteriormente se muestra la primera versión que se realizó sobre el esquema que ha sido presentada por Oscar Pastor [26]. Finalmente se cuentan cuatro iteraciones del ECGH antes de tener la versión que se ha presentado en la sección 4.1 de este trabajo de investigación. En la descripción de las iteraciones se destacan los cambios más relevantes, sin embargo, se observa como un cambio sencillo repercute demasiado en la expresividad de la especificación del dominio: el genoma humano.

3.3.1 Un esquema conceptual inicial para el genoma humano

Un esquema conceptual para el genoma humano fue propuesto por Norman W. Paton en Febrero del 2000 [19]. (Ver Fig.26). En este esquema se muestra una vista conceptual inicial para el genoma humano. Esta vista conceptual describe al genoma humano como

un conjunto de cromosomas divididos en fragmentos que pertenecen a un determinado gen, para lo cual este gen es una fragmentación intergénica. Paton propone la clasificación de un fragmento cromosómico en dos segmentos: una región transcribible (*Transcribed Region*) y una región no transcribible (*Non-Transcribed Region*).

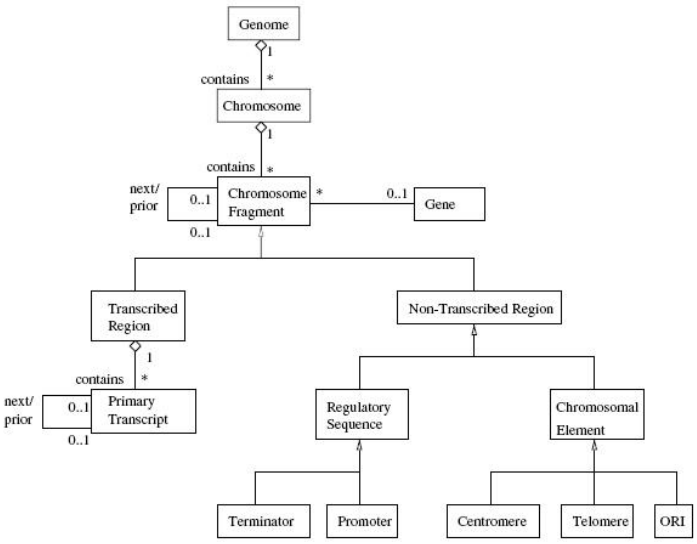


Fig. 26 Esquema Conceptual de Norman Paton.

En este esquema, *Transcribed Region* representa la secuencia transcribible de un gen excluyendo los elementos que no están involucrados en el proceso de transcripción como el promotor o el terminador. *Non-Transcribed Region* incluye las secuencias reguladoras y los elementos cromosomales que lógicamente son parte del cromosoma pero no están involucrados en el proceso de transcripción. Adicionalmente en el esquema de Paton, un conjunto de primarios transcritos puede ser generado desde lo que se identifica como *Transcribed Region*. A partir de este esquema, se elabora un nuevo esquema conceptual del genoma humano, que extiende los conceptos presentados por Paton en este punto de partida.

El esquema conceptual inicial es presentado en [26], proporciona una combinación de las principales características que son consideradas relevantes para conceptualizar los componentes básicos del genoma humano. La principal diferencia sobre el esquema propuesto por Paton es la clasificación de segmentos cromosómicos. En el esquema conceptual inicial del genoma humano, los segmentos cromosómicos son clasificados

como material génico (*Genic*) o material no génico (*Non-Genic*), donde un segmento génico es visto como una composición de un promotor, una secuencia transcribible, un terminador y muchas secuencias potenciadoras (*enhancer*) en contraste con el esquema de Paton donde solamente la región transcribible es considerada como un fragmento génico. Los componentes génicos presentados en la versión inicial del ECGH comparten una relación funcional derivada desde el proceso de la síntesis de proteínas, como resultado de los procesos de transcripción y traducción. Cualquier otra secuencia cromosomal es considerada como segmento no génico. (Ver Fig. 27).

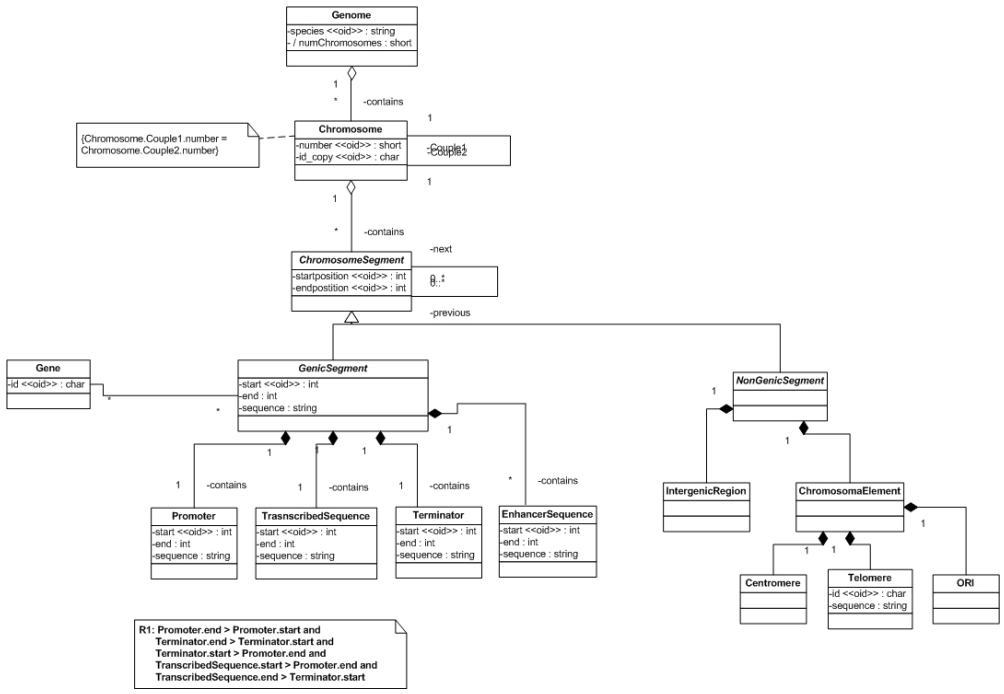


Fig. 27 Un esquema conceptual inicial para la especificación del genoma humano.

3.3.2 Descripción de la evolución del ECGH

A continuación se describirán los cambios más relevantes y los desafíos conceptuales que han llevado a que la versión inicial del ECGH tenga varias iteraciones. Esta evolución ha tenido como objetivo hacer una descripción precisa del genoma humano por lo que ha medida que se adquiere conocimiento sobre el dominio versiones del ECGH son generadas.

La evolución del ECGH es un tema interesante cuando se aplican técnicas de modelado conceptual, en este caso conceptualizar usando diagramas de clases, al genoma humano. Considerando la experiencia de lo complicado de especificar definiciones precisas en este dominio donde día a día se introduce conocimiento nuevo. Al modelar sistemas organizacionales convencionales, sus componentes principales, sus relaciones y sus procesos funcionales son generalmente conocidos; esto hace que el proceso de construir el esquema conceptual sea viable y factible. Cuando se enfrenta el problema de modelar el genoma humano, se observa que los conceptos y su representación pueden variar tanto como el conocimiento bio-genómico crece.

Por lo tanto, la descripción de la evolución del ECGH que se hace a continuación enuncia desafíos conceptuales relevantes involucrados en el proceso que se ha seguido para identificar clases de conceptos para elementos altamente conocidos y usados en el dominio: genoma humano.

El primer desafío conceptual a tener en cuenta fue la conceptualización de un gen. La definición precisa de un gen es un tópico interesante que los biólogos aun debaten. El término ha evolucionado, desde que fue propuesto por W. Johansen en 1909 basado en el concepto desarrollado por Gregor Mendel, para integrar los diferentes paradigmas y teorías que habían ocurrido en la historia de la genética (para una revisión completa ver [29]). El enfoque clásico de un gen es una pieza lineal de ADN heredada de los padres y que codifica para una determinada proteína. Este enfoque ha cambiado considerablemente reflejando la complejidad del término. En la definición actual propuesta por Gerstein en el 2007 [29], un gen es "una unión de secuencias genómicas que codifican un conjunto de productos funcionales solapados potencialmente". Esta evolución del concepto de gen acarrea muchas implicaciones dentro de la definición precisa del mismo.

Durante el trabajo realizado para modelar el genoma humano, una de las cosas mas relevantes, independientemente de la naturaleza evolutiva de los conceptos del dominio, ha sido conocer que la forma en como los biólogos definen a los elementos presentes en el genoma humano es muy diferente a la forma en como los ingenieros de software definen clases de conceptos. Por ejemplo el hecho de considerar que la secuencia de un gen es diferente debido a que tiene formas alternativas de expresión, la idea de que dos

genes pueden compartir el mismo lugar en el cromosoma o que el gen después de la transcripción genera múltiples productos. Sin duda alguna ha sido una tarea difícil identificar las clases de conceptos cuando algún elemento del dominio, en este caso el gen, no tiene una definición única y precisa tanto de sus propiedades como de su comportamiento. Todo esto se ve reflejado en los cambios que ha experimentado este esquema conceptual.

Primera iteración

En la versión inicial del ECGH la clase *Gene* estaba asociada a la clase *GenicSegment*, la cual era un segmento muy grande de ADN compuesto por un promotor, una secuencia de transcripción y un terminador, que además estaba regulado por muchas secuencias potenciadoras (*enhancer*). En una versión siguiente del ECGH, la clase *GenicSegment* llega a ser más pequeña, generalizando segmentos génicos con alguna funcionalidad (promotor, secuencia transcribible, etc.). Después la clase *Gene* fue asociada a la clase *TranscriptionUnit*, la cuál era una estructura rígida que combinaba segmentos génicos involucrados en el proceso de transcripción. Para las versiones siguientes del ECGH, la clase *TranscriptionUnit* llega a ser un concepto más amplio y se incluyen composiciones múltiples de segmentos génicos para integrar una unidad de transcripción. Esto es derivado de la definición actualizada de gen propuesta por Gerstein [29].

La primera iteración del ECGH se observa en la Fig. 28, lo más importante a notar en este nuevo esquema es la aparición de la clase *TranscriptionUnit* como una composición de segmentos cromosómicos involucrados en el proceso de transcripción.

Además permite describir la inexistencia de las secuencias de los promotores o terminadores por la cardinalidad (0,1) en las clases *Promoter* y *Terminator*. Esta especificación en el ECGH es importante considerando que los promotores y terminadores que integran la unidad de transcripción existen pero frecuentemente no son indicados en las fuentes de datos. Sin embargo, la secuencia transcribible es identificada regularmente, por lo que la cardinalidad de la relación de composición con la clase *TranscriptionUnit* es (1..*) con la clase *TranscribedSequence*.

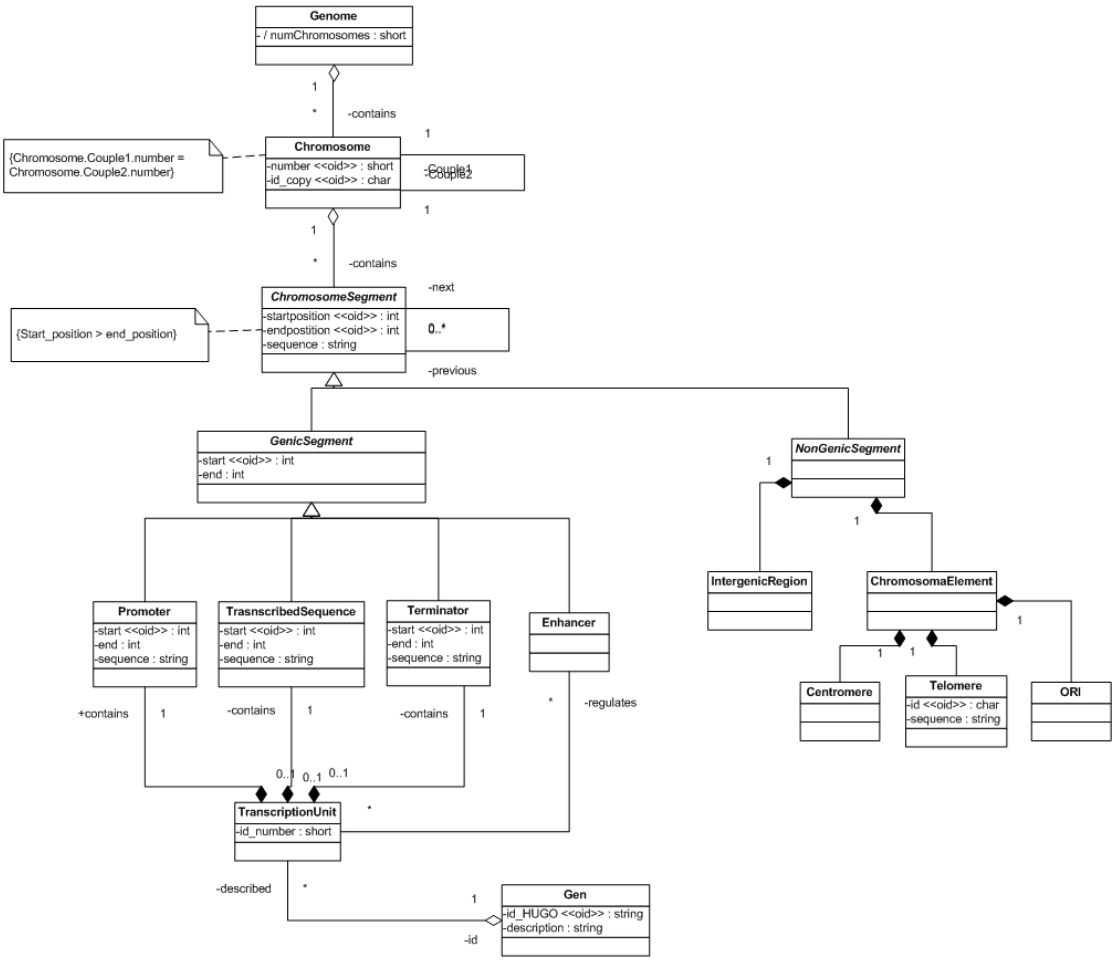


Fig. 28 Primera iteración del ECGH.

Segunda iteración

Siguiendo con el análisis del dominio, se identifica un nuevo concepto importante que implica la realización de cambios en la primera iteración del ECGH. Una secuencia potenciadora (*enhancer*) puede no estar directamente relacionada con ciertas proteínas para potenciar los niveles de transcripción de los genes correspondientes. Esta idea implica que los potenciadores están incluidos en un grupo de secuencias reguladoras que controlan el proceso de la transcripción del ADN, por lo tanto la clase *Enhancer* es reemplazada por la clase *RegulatorSequence*. Considerando que la secuencia reguladora

puede estar relacionada con una o más unidades de transcripción, la clase *RegulatorSequence* es asociada con la clase *TranscriptionUnit*.

El esquema conceptual que refleja los cambios de la primera iteración del ECGH esta representado en la Fig. 29.

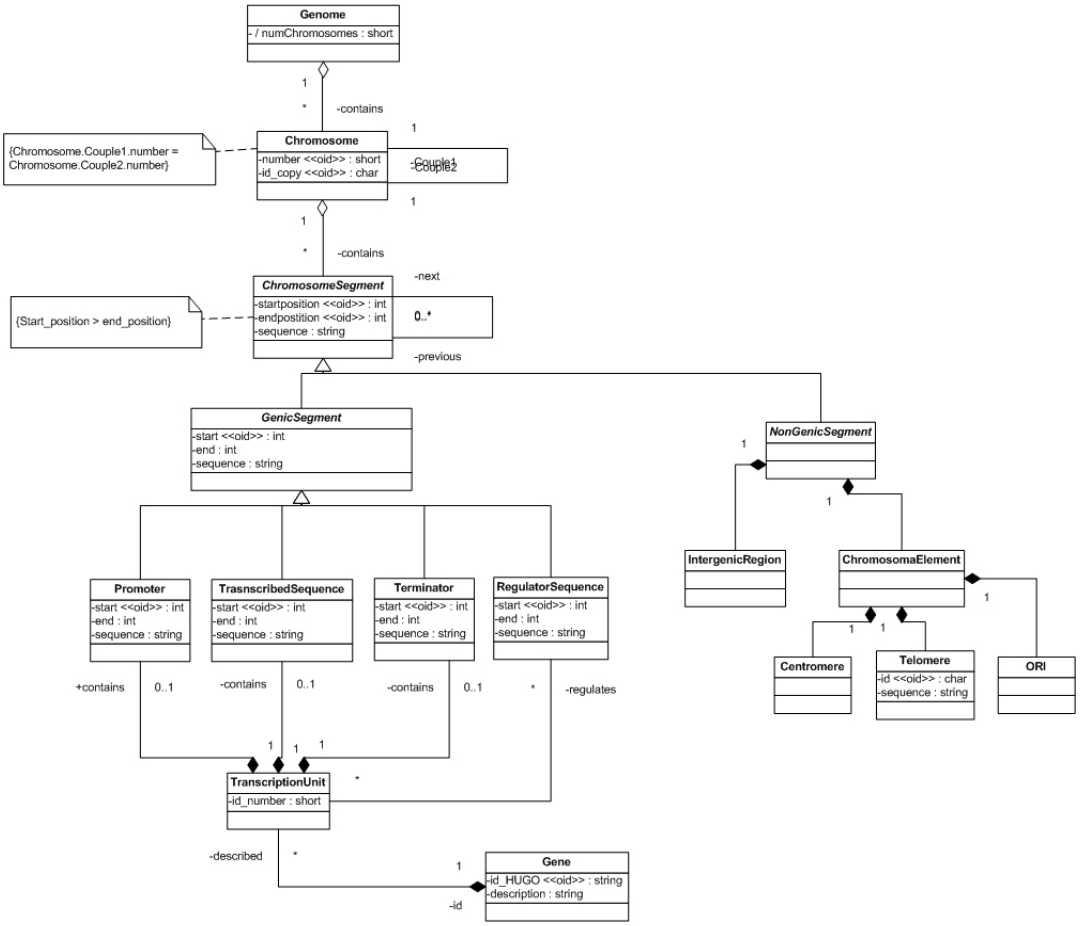


Fig. 29 Segunda iteración del ECGH.

Tercera iteración

Hasta este punto, el ECGH solamente incluía el concepto de gen y su estructura desde un punto de vista del proceso de transcripción. Cuando los procesos de transcripción y traducción implicados en la producción de proteínas fueron estudiados profundamente, se define una nueva iteración del ECGH para especificar en el esquema estos conceptos relacionados con los procesos de transcripción y traducción y los productos obtenidos a

partir de estos procesos (para mas detalles de estos procesos ver [28]). La tercera iteración del ECGH se muestra en la Fig. 30.

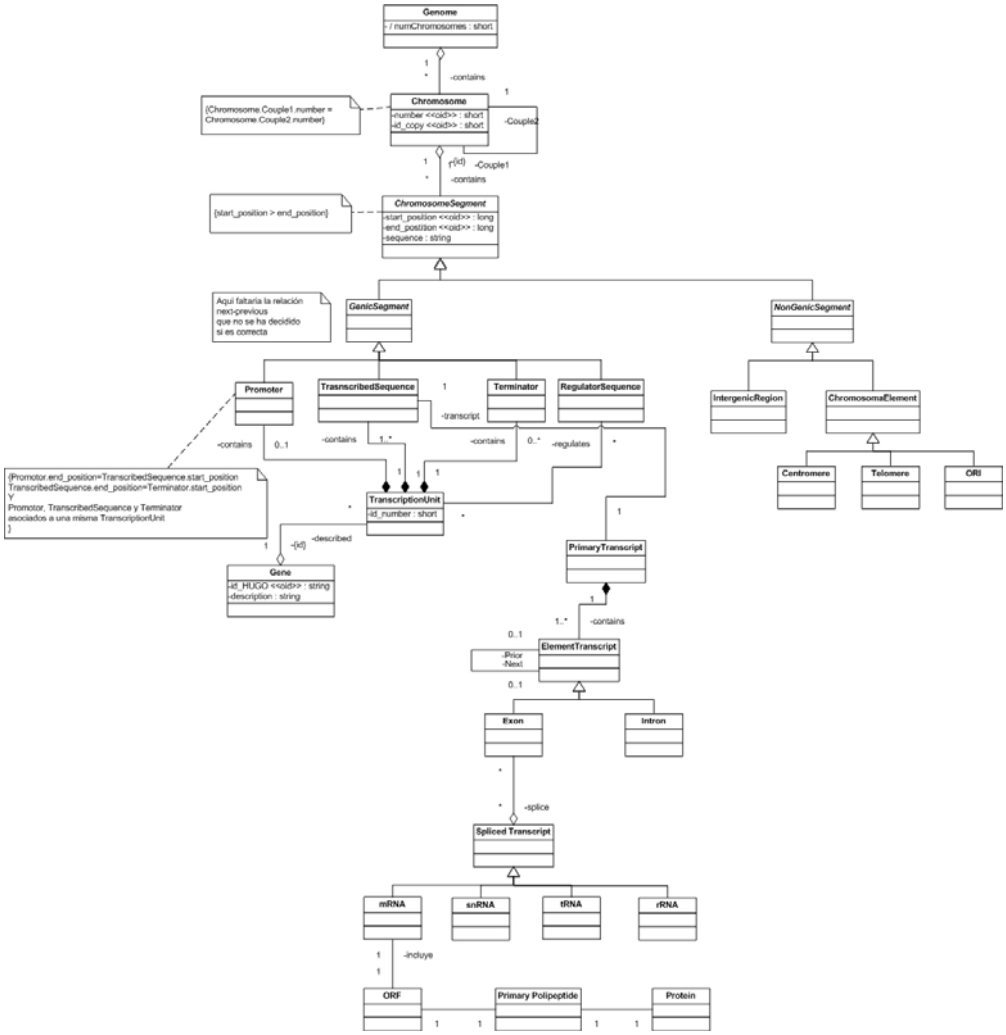


Fig. 30 Tercera iteración del ECGH.

El proceso de síntesis de proteínas inicia con la transcripción: una secuencia de ADN es transcrita a una secuencia complementaria de ARN. Esta molécula de ARN es conocida como primario transcrito. En esta iteración del ECGH, este concepto es representado por la clase *PrimaryTranscript*, la cual esta asociada con un conjunto de instancias de la nueva clase *ElementTranscript*. La especialización de esta clase en la clase *Exon* y la clase *Intron* describe la integración de exones e intrones en la estructura molecular de un primario transcrito. La clase *ElementTranscript* presenta la asociación reflexiva

"Prior,Next" para indicar el orden de los exones o intrones en la secuencia de un primario transcrito.

El proceso Splicing consiste en la combinación de ciertos exones del primario transcrito produciendo un ARNm maduro. La aplicación del proceso Splicing a un transcrito primario es representada en el ECGH por la clase *SplicedTranscript* que es especializada en las clases *mRNA* y *Others RNA types*. La relación de agregación *"splice"* entre las clases *SplicedTranscript* y la clase *Exon* permite identificar a los exones que son el resultado del proceso Splicing.

La clase *mRNA* representa al ARN mensajero, una molécula que contiene la información necesaria para sintetizar una proteína. La secuencia ARN determina el orden de los aminoácidos en la proteína. La clases de los otros ARNs, representan a las secuencias que no son necesariamente traducidas a secuencias de aminoácidos: *snRNA* (small nuclear RNA, participa en importantes procesos nucleares), *tRNA* (transfer RNA, es una molecula muy importante en el proceso de transcripción) y *rRNA* (ribosomal RNA, es parte de un ribosoma).

El siguiente paso en el proceso de síntesis de proteínas es la migración del ARNm maduro desde el núcleo al citoplasma. Aquí el ARNm asociado a los ribosomas inicia el proceso de traducción. La traducción es la producción de proteínas a partir de la decodificación del ARNm producido en la transcripción. Las reglas de decodificación están especificadas en el código genético. Sin embargo, la molécula de ARNm no es traducida completamente, el ORF (Open Reading Frame) es la parte de la secuencia del ARNm usada en el proceso de traducción.

Por lo tanto, la clase *PrimaryPolypeptide* es creada para describir la estructura primaria de la proteína: la cadena de aminoácidos obtenida después de la traducción de un ORF. Esta cadena de aminoácidos sufre algunas transformaciones químicas y el resultado final es una proteína funcional la cual se represente en el ECGH con la clase *Protein*. Por lo tanto la asociación entre la clase *PrimaryPolipeptide* y la clase *Protein* es incluida para conceptualizar que un polipéptido primario da origen a una proteína.

Todas las clases y relaciones entre ellas añadidas al ECGH proporcionaron las bases para un esquema del genoma humano más completo, debido a esto el esquema se hizo muy grande por lo que tuvo que se empezó a pensar en organizarlo en vistas.

Cuarta iteración

La conceptualización del gen en el ECGH se había realizado considerando su estructura, sin embargo la conceptualización de gen es remplazada por la idea de modelar un gen genérico que conceptualice las propiedades de un gen que permanezcan estables y separadas de cualquiera de sus múltiples expresiones. Un alelo es cada una de las formas alternativas que un gen puede tener en la naturaleza. Estas formas alternativas son diferenciadas por sus secuencias y cualquiera de ellas puede producir cambios sobre la función del gen. Una de estas formas es considerada la referencia natural y esta distinguida de cualquiera de las otras variantes alélicas por ser la más abundante en la naturaleza.

La estructura del segmento génico es conservada independientemente de las formas alternativas de un gen. Un segmento génico estará asociado con segmentos génicos genéricos. Esta idea permite conceptualizar la relación entre las formas alternativas de un gen y la estructura del gen. Para explicar la variación entre los alelos, se introduce una clasificación de los cambios que ocurren en cualquiera de esas variantes alélicas. La clasificación se realiza siguiendo los siguientes criterios: la precisión con la que esta descrita la variación y el fenotipo que la variación produce.

La primera clasificación de las variantes alélicas esta dividida en dos categorías. Las variaciones precisas que son las que están descritas actualmente en las bases de datos públicas. Las variaciones imprecisas que son las que están descritas en texto en las fuentes de datos.

La segunda clasificación de las variantes alélicas está dividida en cuatro tipos: 1) Mutación génica, en la que se incluyen las variaciones que producen un efecto patológico; 2) Mutación cromosómica, la cual describe la variación que afecta a más de un gen; 3) Polimorfismo natural, que caracteriza una variación neutral; y 4) Cambios

con consecuencias desconocidas, se refiere a las variantes que reportan una variación con consecuencias desconocidas.

Incluir estos conceptos en el ECGH provocó el diseño de una nueva iteración del esquema. Los cambios presentes en esta cuarta iteración pueden observarse en la Fig. 31.

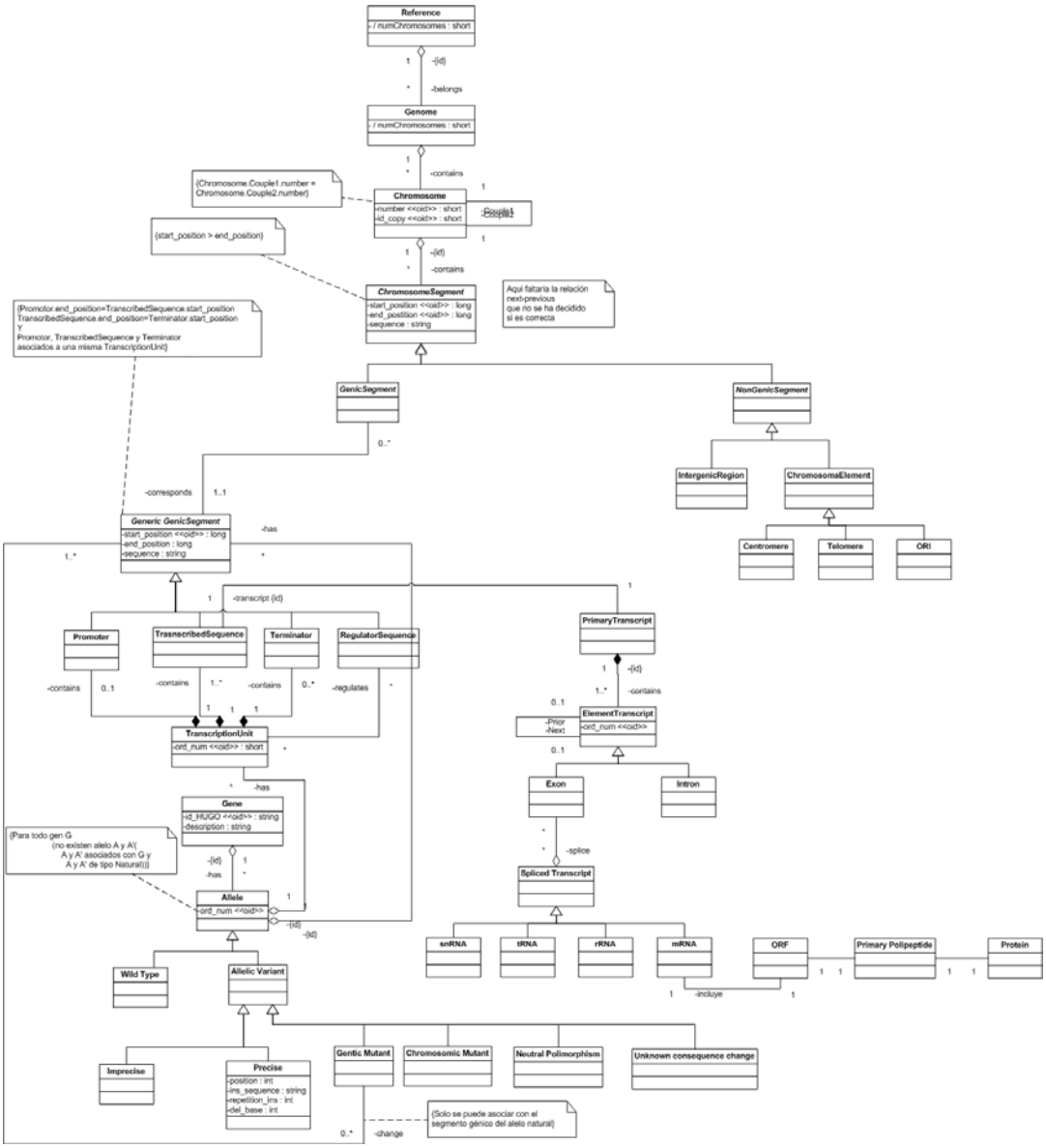


Fig. 31 Cuarta iteración del ECGH.

Una de las clases incluidas en esta iteración del esquema es la clase *Reference*, que representa la fuente que proporciona la información de un genoma determinado. Por lo que también se incluye la relación de agregación *"belongs"* desde la clase *Genome*. La idea de modelar un gen genérico elimina la especialización de la clase *GeneSegment* en las clases *Promoter*, *TranscribedSequence*, *Terminator* y *RegulatorSequence*. Este cambio es realizado debido a que lo que se considera segmento génico no tiene una naturaleza genérica.

La clase *GenericGenicSegment* es añadida al esquema para representar el segmento génico genérico. Un segmento génico puede estar asociado a un segmento génico genérico. Esto se describe en el esquema con la asociación *"corresponds"* entre la clase *GenicSegment* y la clase *GenericGenicSegment*.

La clase *Allele* es agregada al esquema para almacenar las formas alternativas de un gen encontradas en los repositorios de datos. Debido a esto, el gen ya no estará asociado a la unidad de transcripción pero si a los alelos. Por lo que se introduce la relación de agregación *"has"* entre la clase *Gene* y la clase *Allele* y también la relación de agregación *"has"* entre la clase *Allele* y la clase *GenericGenicSegment*.

La clase *Allele* es especializada en las clases *WildType*, para representar el alelo natural; y *AllelicVariant*, para representar las variantes alélicas en comparación con el alelo natural. Los grupos de variantes alélicas son especificados en el esquema a través de la especialización de la clase *AllelicVariant* en dos grupos de clases. El primer grupo es especializado en las clases *Imprecise* y *Precise*. El segundo grupo es especializado en las clases *GenicMutant*, *Chromosomic-Mutant*, *NeutralPolimorphism* y *UnknownConsequenceChange*.

Como se ha mencionado repetidas veces, la naturaleza de los elementos del dominio: genoma humano, permite que el esquema tenga cambios significativos. Lo que se ha descrito sobre la evolución en estas cuatro iteraciones presentadas es solo un resumen de los cambios que han tenido que realizarse en el esquema. Ha sido un proceso de investigación de más de un año en el que expertos en genética y expertos en modelado conceptual unen esfuerzos por hacer una especificación del genoma humano en un esquema conceptual que sea lo suficientemente descriptivo para entender el

funcionamiento y la estructura de este dominio y a partir de eso aplicar los beneficios de los sistemas de información.

4 Una base de datos a partir del ECGH

Resumen: La idea principal de este capítulo consiste en describir el proceso de creación de una base de datos a partir del esquema conceptual ECGH. En la primera sección se proporciona una perspectiva sobre la base de datos y se describen especificaciones técnicas implicadas en su creación. En la segunda sección, se comenta la necesidad de realizar una serie de adecuaciones en el ECGH para la creación de la base de datos. Estas adecuaciones son necesarias para que la base de datos pueda ajustarse a la información biológica publicada actualmente. En este escenario, se presenta la idea de tener dos esquemas conceptuales para el genoma humano; un esquema conceptual ideal y un esquema conceptual real que coexisten concientemente.

4.1 *Perspectiva de la base de datos*

La base de datos *Genome* creada a partir del Esquema Conceptual del Genoma Humano (ECGH) es creada con la intención de que actúe como un repositorio unificado que permita a los biólogos buscar y recuperar de manera eficiente información sobre secuencias del genoma humano.

Cuando el ECGH tuvo una versión estable (presentada en la sección 4.1), se realizó el proceso de transformación del esquema conceptual al esquema relacional. La base de datos *Genome* ha sido creada en un Servidor Oracle 10g [30].

A partir de la creación de la base de datos (tablas, atributos, restricciones, etc.) se inició un proceso de búsqueda y revisión a través de las bases de datos biológicos públicas frecuentemente utilizadas por los biólogos. La intención era identificar información para almacenarla en la base de datos *Genome*. Las bases de datos revisadas fueron GENE NCBI, Nucleotide NCBI, PubMed NCBI [6] y HGMD [7]. Es importante mencionar que estas bases de datos fueron sugeridas por el equipo de biólogos expertos que colaboran en este proceso de investigación.

En la base de datos GENE NCBI, se encontró información genérica acerca de los genes. Esto ha sido de mucha ayuda en la base de datos *Genome* para almacenar información descriptiva del gen genérico que se ha modelado en el ECGH. Esta base de datos

proporciona enlaces a la base de datos Nucleotide NCBI que almacena información mas detallada relacionada a la secuencia del gen.

En la base de datos Nucleotide NCBI, se encontró información detallada acerca de las secuencias de los genes y sus productos derivados (alelo de referencia, ARN o exones).

En la base de datos PubMed NCBI, se encontró almacenada literatura relevante sobre artículos biomédicos. Lo que permite cargar referencias bibliográficas sobre la información de los genes, alelos, variaciones alélicas, ARNs y proteínas en la base de datos *Genome*.

En HGMD, se encontró información relacionada con las mutaciones conocidas sobre los genes. Esta base de datos tiene una versión extendida y más actualizada para usos comerciales y una versión más sencilla y no actualizada para usos académicos. En el momento de iniciar la carga de datos en *Genome* las consultas a HGMD fueron realizadas a través de la vista de usos académicos. Con la información proporcionada por esta fuente es posible establecer comparaciones entre secuencias de referencia y secuencias mutadas.

Al llegar al punto de determinar la información útil, contenida en las diferentes fuentes, para cargar la base de datos *Genome*, se encontraron recursos que algunas de estas fuentes proporcionan para la recuperación de la información. Sin embargo, la idea de cargar automáticamente *Genome* requiere el diseño y la implementación de algoritmos que integren los recursos de recuperación de información que las fuentes proporcionan.

Además del problema de la carga automática de *Genome*, también se presenta el problema de mantenimiento de la información con el objetivo de que esta base de datos tenga siempre información actualizada.

Es importante notar que el funcionamiento eficiente del repositorio unificado *Genome* requiere de un trabajo de investigación exhaustivo que permita hacer frente a los retos de carga automática y mantenimiento de esta información.

4.2 *Un esquema conceptual real frente al esquema conceptual ideal*

El desafío de conceptualizar el conocimiento del genoma humano, entendiendo la forma en como los biólogos definen los conceptos desde un punto de vista tan diferente a la forma en como lo hacen los ingenieros de software, no ha sido la única clase de problemas que se han presentado al realizar este trabajo de investigación.

Cuando la base de datos *Genome*, a partir del esquema conceptual del genoma humano fue creada, se inicia con un proceso de investigación sobre las diferentes fuentes de datos que proporcionan datos biológicos con la finalidad de desarrollar e implementar algoritmos que almacenen automáticamente la información en la base de datos *Genome*. Ha sido posible identificar que la información publicada disponible actualmente no presenta todos los elementos que se han modelado en el ECGH.

Sin embargo, ha sido muy interesante darse cuenta de que muchas de las bases de datos publicas no tienen considerada la naturaleza evolutiva de los elementos presentes en el genoma humano. Por lo que van haciendo adecuaciones conforme necesitan incluir más información lo que provoca que la búsqueda y recuperación de información sea una tarea difícil para los usuarios, en este caso los biólogos.

Como se ha mencionado, al iniciar la carga de datos en la base de datos *Genome* se encuentran un conjunto de situaciones donde los datos extraídos de las fuentes no pueden adecuarse totalmente con lo modelado en el ECGH.

Por lo tanto se adopta la idea de conservar dos vistas en paralelo del esquema conceptual: una más teórica, describiendo el dominio del genoma humano desde una base puramente conceptual; y otra más práctica, considerando la forma en la cual los datos están almacenados actualmente en los repositorios biológicos existentes.

También se ha observado que las representaciones de los datos en los repositorios biológicos han sido hechas sin una conceptualización precisa de los elementos presentes en el genoma humano.

Por estas razones se crea un sistema en el cual dos esquemas coexisten concientemente: Un esquema llamado "Esquema Ideal" (Ideal schema IS), que refleja toda la información relevante conocida sobre las clases, atributos y relaciones entre clases en el dominio del Genoma Humano desde su percepción teórica; y un "Esquema Real" (Real schema RS), que refleja practicas actuales, principalmente representadas por la información que puede ser capturada desde los repositorios biológicos públicos.

Para evitar que la información se pierda entre los dos esquemas, es necesario definir un proceso de sincronización basado en la evolución que el esquema real (RS) experimente. Esto garantiza la coexistencia de los dos esquemas. La intención es facilitar una adecuada evolución del conocimiento: cuando exista un cambio en los datos proporcionados por un repositorio biológico, el RS puede ser cambiado fácilmente para adaptar la nueva realidad y conservar la consistencia del IS.

Un ejemplo bastante claro de esto se muestra cuando se quiere almacenar información en la tabla *PrimaryTranscriptPath* de la base de datos *Genome*. Actualmente no existe información en los repositorios biológicos públicos en la que se almacenen datos sobre las diferentes combinaciones del primario transcrito generadas a partir del proceso Splicing alternativo. Sin embargo, existe mucha bibliografía disponible en las que estas secuencias son referenciadas. Considerando esta situación, la clase *PrimaryTranscriptPath* del esquema ideal no es incluida en el esquema real. De cualquier forma el concepto de combinaciones diferentes de los exones presentes en un transcrito primario existe por lo que debe conservarse en el esquema real. Dado que se define un proceso de sincronización entre los esquemas, en un momento determinado la tabla *PrimaryTranscriptPath* de la base de datos *Genome* podrá ser cargada de información por derivaciones de la información que se tenga en la base de datos implementada a partir del esquema real.

Es evidente que la especificación conceptual del genoma humano no ha sido solo un desafío para modelar los elementos del dominio que se identifican a partir de su funcionamiento. Esta especificación también ha tenido que hacer frente al hecho de que los datos disponibles, sobre lo que se ha estudiado del genoma humano a lo largo del tiempo, es información demasiado heterogénea en cuanto a su estructura y que además hay conceptos importantes del dominio que no tiene en consideración. Sin embargo,

esta información es valiosa y es recuperada y buscada por los biólogos de una forma constante.

5 Conclusiones

La aplicación de técnicas de Modelado Conceptual en dominios poco convencionales, en este caso el genoma humano; es una evidencia clara de las ventajas tan importantes que aportan estas técnicas en el diseño un Sistema de Información (SI). Trabajar a un nivel alto de abstracción ha permitido que conceptos sobre los componentes y procesos que existen en el genoma humano hayan podido ser mejor comprendidos.

El proceso de investigación realizado para delimitar e identificar clases de conceptos en el dominio del problema ha sido una tarea difícil. Principalmente porque la forma en que los biólogos identifican los elementos del genoma humano difiere en gran medida con la forma en que, comúnmente, los ingenieros de software identifican las clases de conceptos en un determinado dominio. Esta diferencia fue observada en el momento de modelar los atributos de las clases que representaban conceptos del dominio. Se observó que para el mismo concepto no podían identificarse propiedades totalmente constantes y que el hecho de que estas propiedades tuvieran valores distintos no significaba que fueran instancias diferentes. Por lo que, no era posible modelar conceptos del genoma humano al igual que conceptos de dominios convencionales que tienen una naturaleza estática en las propiedades de sus elementos. Lo verdaderamente desafiante al diseñar el ECGH ha sido abordar el Modelado Conceptual a partir de elementos del dominio que poseen una naturaleza dinámica en sus propiedades, es decir, que éstos elementos se consideran como elementos del dominio cuando tienen un determinado comportamiento.

Sin embargo, el comportamiento evolutivo del diseño del ECGH, es similar al comportamiento presentado en el diseño de esquemas conceptuales para otros dominios. Este escenario ilustra el hecho de que con la obtención de conocimiento sobre el dominio del problema, el esquema conceptual evoluciona, garantizando que las clases de conceptos identificados están completos y son correctos.

El ECGH y en consecuencia la creación de la base de datos "*Genome*", puede considerarse una herramienta eficiente para el diseño e implementación de un Sistema de Información que contribuya de manera eficiente con los problemas de integración de datos para la búsqueda y recuperación de información valiosa acerca de los estudios realizados sobre la secuenciación de genomas de los seres humanos.

Por lo tanto, el ECGH es el punto inicial para una serie de trabajos futuros. Uno de los más significativos es la definición y el diseño de una arquitectura para un SI en los que se aborden las siguientes ideas, principalmente:

- Integración de datos de fuentes externas y su inclusión en la base de datos "Genoma", considerando principalmente los mecanismos de carga automática y mantenimiento de la información.
- Gestionar la presencia del esquema real (ER) y el esquema ideal (EI) del ECGH, definiendo mecanismos de sincronización que permitan la coexistencia de los dos esquemas.
- Diseñar e implementar aplicaciones software que permitan a los usuarios finales buscar y recuperar de manera eficiente información sobre la secuenciación de los genomas de los seres humanos.

Estos tres tópicos en los que se basa la arquitectura, definen una forma modular de encarar problemas importantes presentes en el trabajo que realizan los biólogos hoy en día. Trabajando en cada modulo de manera paralela es posible obtener avances significativos sin tener que esperar el hecho de proporcionar una solución completa. Sin embargo, el trabajo ha de ser realizado con el objetivo fijo de otorgar una solución integral basada en la arquitectura.

Sobre la integración de datos, un trabajo claramente identificable a realizar, es la investigación y determinación de las fuentes externas de información que proporcionen los datos que han de incluirse en la base de datos *"Genome"*. Al mismo tiempo, deberán analizarse los medios informáticos, que cada fuente proporciona para la recuperación de información. Además deberán diseñarse e implementarse algoritmos de carga automática y mantenimiento de la información que estará contenido en *"Genome"*.

La gestión de la coexistencia de los esquemas ideal y real para el genoma humano, deberá ser manejada con la conciencia plena de que, a medida que la investigación en la secuenciación de genomas de seres humanos avance, el esquema real llegará a ser lo más similar posible al esquema ideal.

Muchas aplicaciones software pueden ser diseñadas e implementadas a partir de contar con información valiosa dentro de la base de datos *"Genome"*. El uso de esta información es sin duda una posibilidad importante para lograr un alto grado de eficiencia en los procesos de búsqueda y recuperación de datos biológicos. Procesos que hoy en día demandan demasiado tiempo en su realización.

Estas aplicaciones software permitirán la explotación de la información contenida en *"Genome"* proporcionando a los biólogos un grupo de utilerías que contribuyan significativamente en su trabajo diario.

De acuerdo con lo mencionado en párrafos anteriores, es posible observar que el diseño del ECGH es el inicio del diseño y desarrollo de un Sistema de Información que estará basado en una arquitectura que al momento se encuentra parcialmente definida. Sin embargo, la piedra angular de este nuevo Sistema de Información será el ECGH. Esto tiene la intención de aprovechar la serie de ventajas que el Modelado Conceptual ha aportado a otros Sistemas de Información incrementando su calidad.

El trabajo de diseñar un esquema conceptual para el genoma humano, surge a partir de la demanda de aplicaciones software que presenta el trabajo que realizan los biólogos actualmente. No obstante, existen varios trabajos de investigación que se han realizado. Esta serie significativa de trabajos bioinformáticos referentes al procesamiento de información del genoma humano es realizada a partir de que la ciencia contó con la secuenciación de un genoma entero. Sin embargo, esta línea de investigación esta constituida por tópicos innovadores que permiten trabajar conjuntamente temas de Informática y Biología Molecular. Lo que ilustra claramente que es una línea de investigación novedosa debido a la necesidad latente de conjuntar principios científicos y el desarrollo de herramientas que contribuyan al la conservación de la salud humana, en el caso particular de este trabajo.

El hecho de que el ECGH tiene las posibilidades necesarias de ser una contribución importante se ilustra en las publicaciones que han podido realizarse de este trabajo. El esquema fue expuesto en el *International Workshop on Data Integration in the Life Sciences (DILS 2009),* realizado en la *Universidad de Manchester* en Julio del 2009. (Ver Anexo 3).

Además, se escribió un capitulo para el libro: *Dagstuhl seminar book on the evolution of conceptual modeling*. El titulo del capitulo es *Model Driven-Based Engineering Applied to the Interpretation of the Human Genome* y en él se incluye información importante sobre la experiencia de aplicar técnicas de modelado conceptual para modelar el genoma humano como un Sistema de Información. El libro es un Springer LNCS y será publicado a finales de Septiembre del 2009. (Ver Anexo 3).

El diseño del ECGH es el punto de partida para un proceso de investigación exhaustivo que permita adquirir el conocimiento necesario sobre el genoma humano y los diferentes medios y mecanismos para estudiarlo, al mismo tiempo, teniendo el ECGH puede abordarse al genoma humano como un Sistema de Información enfrentando un dominio diferente pero muy interesante.

Referencias

[1] Genome programs of the U.S, Departament of Energy Office of Science, Human Genome Project Information, http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml

[2] Kirschner, M.W.: The Meaning of Systems Biology. Cell Press Journal. 121, 503-504(2005)

[3] Joyanes, L.: La Bioinformática como convergencia de la Biotecnología y la Informática. En: Estudios, Biología y Sociedad. no.1. pp. 98-119. Fundación Pablo VI, Madrid (2003)

[4] Sanchez, M.: Facilitating Genomic Medicine for Future Healthcare. Articulo de prensa. J. Biomed. Información (2003)

[5] Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice. Springer-Verlag. Berlin-Heidelberg (2007)

[6] National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov

[7] Human Gene Mutation Database, http://www.hgmd.cf.ac.uk/ac/index.php

[8] Human Genome Nomenclature Committee, http://www.genenames.org

[9] Babu, P., Boddepalli, R., Lakshmi, V., Rao, G.: Dod: Database of databases updated molecular biology databases. Silico. Biol. 5 (2005).

[10] Thorisson, G., Muilu, J., Brookes, A.: Genotype-phenotype databases: challenges and solutions for the post-genomic era. Nature Reviews Genetics. 10, 9-18 (2009)

[11] The MDA reference model, http://ormsc.omg.org/mda_guide_working_page.htm

[12] Object Management Group, http://www.omg.org/

[13] Mylopoulus, J.: Information modeling in the time of the revolution. Information Systems 23, 3-4 (1998)

[14] Uschold, M., Gruninger, M.: Ontologies: principles, methods and applications. The knowledge engineering review 11, 2 (1996)

[15] Olivé, T.: An introduction to conceptual modeling of information systems. Piatini, Diaz (eds) Avanced database technology and design. Artech House. 2000

[16] Falkenberg E., Hesse, W., Lindgreen, P., Nilsson B., Han Oei, J., Rolland, C., Stamper, R., Van Assche, F., Verrijn-Stuart A., Voss, K.: A Framework Of Information System Concepts: FRISCO. Technical report, IFIP Web Edition (1998).

[17] Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language User Guide. Addison-Wesley professional. Pearson Education, New Jersey (1999)

[18] Olivé, A.: Conceptual Modelling of Information Systems. Springer-Verlag. Berlin-Heidelberg (2007); Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice. Springer-Verlag. Berlin-Heidelberg (2007)

[19] Paton, W.N., Khan, S., Hayes A., Moussouni, F., Brass, A., Eilbeck, K., Globe, C., Hubbard, S., Oliver, S.: Conceptual modelling of genomic information. In: Oxford University Press 2000. vol. 16, no.6, pp. 548—557. Bioinformatics, Manchester (2000)

[20] Garwood, K., Garwood, C., Hedeler, C., Griffiths, T., Swainston, N., Oliver S., Paton, W.: Model-driven user interface for bioinformatics data resources: regeneriting the wheel as an alternative to reinventing it. In: Biomed Central. vol.7, no. 532, pp. 1--14. Bioinformatics, Manchester (2006)

[21] Bornberg-Bauer, E., Paton, N.: Conceptual data modelling for bioinformatics. In: HENRY STEWART PUBLICATIONS 1467-5463, vol. 3, no. 2, pp. 166--180. Briefings in bioinformatics, Manchester (2002)

[22] Hedeler, C., Wong, H.M., Cornell, M.J., Alam, I., Soanes, D., Rattray, M., Hubbrad, S.J., Talbot, N.J., Oliver, S.G., Paton, N.: e-Fungi: a data resource for comparative analysis of fungal genomes. BMC Genomics. 8, 426, 1–15 (2007)

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

[24] Ram, S.:Toward Semantic Interoperability of Heterogenous Biological Data Sources. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS. vol.3520, p.32. Springer. Heidelberg (2005)

[25] Object Constraint Language Specification, http://umlcenter.visual-paradigm.com/umlresources/obje_11.pdf

[26] Pastor, O.: Conceptual Modeling Meets the Human Genome. Conceptual Modeling - ER 2008. LNCS. vol.5231, p.1. Springer-Verlag. Berlin-Heidelberg (2008).

[27] Scherrer, K., Jost, J.: Gene and genon concept: coding versus regulation. Theory Biosci. 126, 65–113 (2007)

[28] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P.: Molecular Biology of the cell. Garland Science, New York (2002), http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=mboc4

[29] Gerstein, M.B., Bruce, C., Rozowsky, J., Zheng, D., Du, J., Korbel, J., Emanuelsson,
O., Zhang, Z., Weissman, S., Snyder, M.: What is a gene, post-ENCODE? History and updated definition. Genome Res. 17, 669–681 (2007)

[30] ORACLE organization, http://www.oracle.com/index.html

[31] Watson, J., Crick, F.: Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. Nature. 171, 737-738 (1953)

[32] Chargaff, E.: Calculated Composition of a 'Messenger' Ribonucleic Acid. Nature 194, 86-87 (1962)

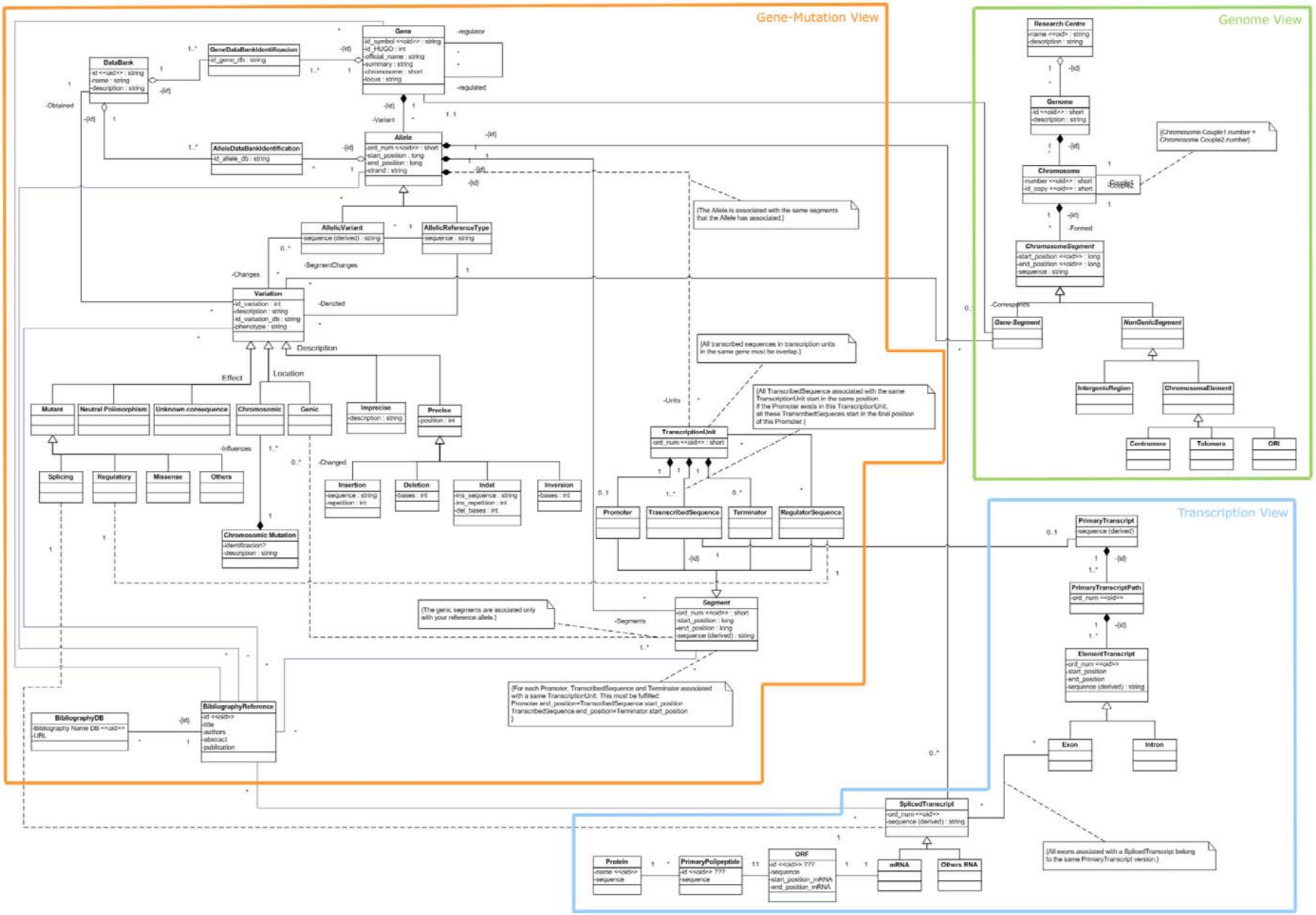
[33] Human Genome Project Information, http://www.ornl.gov/sci/techresources/Human_Genome/primer_pic.shtml

[34] Wikipedia, http://commons.wikimedia.org/wiki/File:DNA_chemical_structure_es.svg

Anexos

Anexo 1

Esquema Conceptual del Genoma Humano (ECGH).



Anexo 2

DESCRIPCION DE LOS ELEMENTOS DEL ESQUEMA CONCEPTUAL DEL GENOMA HUMANO.

El Esquema Conceptual del Genoma Humano (ECGH) ha sido diseñado a partir de los principios del estándar UML, específicamente se han utilizado los diagramas de clase como lenguaje de modelado. Para mayor expresividad de los elementos del esquema conceptual, se incluyen expresiones OCL. Estas expresiones permiten describir propiedades importantes del dominio a través de restricciones de integridad y leyes de derivación.

El ECGH presentado en el Anexo 1 esta divido en tres vistas principales para facilitar su visualización y comprensión. En la primera parte de este anexo se hace una descripción técnica de cada vista haciendo énfasis en su objetivo y en sus clases que en conjunto representan al Sistema de Información. Posteriormente, en la segunda parte, se enuncian las asociaciones entre las clases así como las restricciones de integridad que se presentan.

1. *Descripción de las clases y atributos del ECGH a partir de su organización en vistas.*

El esquema conceptual del genoma humano, esta organizado en tres vistas con la intención de facilitar la comprensión del esquema. Las tres vistas son: *Gene – Mutation View*, *Transcription View* y *Genome View*.

A continuación se enuncian y describen a detalle las clases del ECGH que están presentes en cada vista. Al mismo tiempo se mencionan los atributos de cada clase y lo que representan. Estas descripciones están organizadas en un formato tabular para facilitar su manejo.

Nombre de la vista:	**Gene – Mutation View**
Objetivo:	Conceptualizar la estructura interna de los genes y la actividad biológica que realizan al expresarse.

Clase	**Descripción**	**Atributos**		
		Nombre	**Tipo**	**Descripción**
Gene	Representa a un Gen genérico, independiente a su expresión.	*id_symbol <<oid>>*	string	Código alfanumérico que coincide con el símbolo que HGNC otorga a los genes.
		i*d_HUGO*	int	Código numérico que representa el código universal para los genes de acuerdo con el HGNC.
		oficial_name	string	Nombre común con el que se denomina al gen.
		summary	string	Resumen del gen.
		choromosome	short	Número de cromosoma donde el gen está localizado.
		locus	string	Posición del gen dentro del cromosoma.
Allele	Conceptualiza las características que surgen a partir de la expresión del gen. (Los alelos).	*ord_num <<oid>>*	short	Identificador de la clase.
		start_position	long	Posición inicial respecto al cromosoma.
		end_position	long	Posición final respecto al cromosoma.
		strand	string	Almacena el valor *plus* o *minus* de acuerdo con la hebra del ADN en la que se ubica el alelo.
DataBank	Agrupa las referencias externas de la información para los genes y los alelos.	*id<<oid>>*	string	Identificador para cada fuente de datos.
		name	string	Almacena el nombre de la fuente de datos.
		description	string	Contiene una descripción acerca de la fuente de datos
GeneDataBankIdentification	Representa la identificación de un gen en las diferentes bases de datos públicas.	*id_gene_db*	string	Almacena el identificador que le otorga la fuente de datos externa a un determinado gen.
AlleleDataBankReference	Representa la identificación de un alelo en las diferentes bases de datos públicas.	*id_allele_db*	string	Almacena el identificador que le otorga la fuente de datos externa a un determinado alelo.
AllelicVariant	Es una especialización de la clase *Allele.* Representa las secuencias de nucleótidos que tienen los alelos que se utilizan como referencias para las variantes alélicas.	*Sequence (derived)*	string	Almacena la secuencia de nucleótidos del alelo.

Variation	Agrupa las posibles variaciones asociadas a un alelo de referencia.	*id_variation<<oid>>*	int	Identificador de la clase.
		description	string	Almacena información acerca de la variante alélicas.
		id_variation_db	string	Almacena el identificador de la base de datos externa en la que esa variación esta registrada.
		phenotype	string	Almacena la especificación fenotípica del efecto que produce la variación.

La clasificación de las variantes alélicas esta representada por una especialización jerárquica de la clase *Variation*, utilizando los discrimadores *Location* (rango), *Description* (grado de conocimiento) y *Effect* (efecto que produce) para agrupar las variaciones.

Clase	**Descripción**	**Atributos**		
		Nombre	**Tipo**	**Descripción**
Genic	*(Location)* Representa a las variaciones que afectan a un solo gen.			
Chromosomic	*(Location)* Representa a las variaciones que afectan partes del cromosoma.			
ChromosomicMutation	Agrupa las variantes alélicas presentes en la variación cromosómica y que afectan a uno o muchos genes dentro del mismo cromosoma.	*identification*		Identifica la variación cromosómica.
		description	string	Almacena una descripción de la variación cromosómica.
Impresice	*(Description)* Representa a las variaciones para las cuales sus detalles son desconocidos.	*description*	string	Almacena la información acerca de la variación.
Precise	*(Description)* Representa a las variaciones para las cuales sus detalles son conocidos.	*position*	int	Almacena un valor numérico que indica la posición de la ocurrencia de la variación.
Insertion	Es una especialización de la clase *Precise*. Describe los detalles de las variantes alélicas que presentan inserciones de nucleótidos respecto a la secuencia descrita en un alelo de referencia determinado.	*sequence*	string	Almacena la secuencia de nucleótidos o bases que se ha insertado.
		repetition	int	Almacena el numero de veces que se ha insertado la secuencia descrita en el atributo *sequence.*

Deletion	Es una especialización de la clase *Precise*. Describe los detalles de las variantes alélicas en las que ciertos nucleótidos fueron borrados en una posición específica de acuerdo con la secuencia descrita en un alelo de referencia.	*bases*	int	Almacena el número de nucleótidos o bases que han sido borrados.
Indel	Es una especialización de la clase *Precise*. Describe los detalles de las variantes alélicas en las que ocurrió una deleción de ciertos de nucleótidos y después ocurrió una inserción de uno o varios nucleótidos un determinado numero de veces.	*ins_sequence*	string	Describe la secuencia de nucleótidos que ha sido insertada.
		ins_repetition	int	Almacena el numero de veces que se ha insertado la secuencia descrita en el atributo *ins_repetition.*
		del_bases	int	Almacena el número de nucleótidos que han sido borrados.
Mutant	*(Effect)* Describe las variantes alélicas que provocan un determinado efecto en el proceso de síntesis de proteínas.			
Splicing	Es una especialización de la clase *Mutant*. Describe las mutaciones que afectan el proceso de Splicing.			
Regulatory	Es una especialización de la clase *Mutant*. Describe las mutaciones que afectan la regulación de un gen.			
Missense	Es una especialización de la clase *Mutant*. Describe las mutaciones que surgen cuando un nucleótido es cambiado dando como resultado un codón que codifica para diferente amino acido y por lo tanto es producida una proteína no funcional.			
Others	Es una especialización de la clase *Mutant*. Agrupa detalles de variantes alélicas que tienen un efecto en el fenotipo pero que no se puede clasificar.			

NeutralPolimorphism	*(Effect)* Describe las variantes alélicas que no afectan al fenotipo.			
UnknownConsequence	*(Effect)* Describe las variantes alélicas de consecuencias que no son conocidas todavía.			
BibliographyDB	Representa características de las bases de datos de referencias bibliográficas de las que se obtiene la información.	*BibliographyNameDB <<oid>>*		Almacena el nombre de la base de datos.
		URL		Almacena la dirección electrónica de la base de datos.
BibliographyReference	Representa información sobre las publicaciones relevantes contenidas en las bases de datos bibliográficas.	*Id <<oid>>*		Identificador de la clase.
		title		Almacena el titulo del artículo científico.
		authors		Almacena los nombres de los autores.
		abstract		Almacena el resumen del artículo.
		publication		Almacena el artículo completo.
Segment	Representa un segmento alélico que posee una secuencia significativa e indivisible de ADN.	*ord_num <<oid>>*	short	Funciona como identificador de la clase.
		start_position	long	Almacena la posición inicial del segmento alélico dentro del cromosoma.
		end_position	long	Almacena la posición final del segmento alélico dentro del cromosoma.
		Sequence (derived)	string	Almacena la secuencia de nucleótidos delimitada por los atributos *start_position* y *end_position.*
Promoter	Es una especialización de la clase *Segment*. Describe la secuencia de ADN que marca el inicio del proceso de transcripción.			
TranscribedSequence	Es una especialización de la clase *Segment*. Describe la secuencia de ADN transcrita por el ARN polimerasa.			
Terminator	Es una especialización de la clase *Segment*. Describe la secuencia de ADN que marca el fin del proceso de transcripción.			
RegulatorSequence	Describe un segmento alélico que contiene las secuencias de nucleótidos de las funciones de regulación de uno o mas procesos de transcripción.			Funciona como identificador.

TranscriptionUnit	Es una composición de las clases *Promoter*, *TranscribedSequence* y *Terminator*. Representa a la unidad de transcripción.	*ord_num <<oid>>*		

Las vistas *Gene-Mutation View* y *Transcription View* del SI, tienen como finalidad registrar información genérica sobre genes, mutaciones y procesos de transcripción, aceptada por la comunidad científica.

Nombre de la vista:	**Transcription View**
Objetivo:	Conceptualizar los elementos involucrados en el proceso de transcripción y de síntesis de proteínas.

Clase	**Descripción**	**Atributos**		
		Nombre	**Tipo**	**Descripción**
PrimaryTranscript	Representa la copia transcrita desde el ADN a ARN de la secuencia que describe la clase *TranscribedSequence*.	*sequence (derived)*		Representa la secuencia de nucleótidos de la unidad de transcripción.
PrimaryTranscriptPath	Representa las diferentes particiones, impulsadas por el factor splicing, que tiene un transcrito primario.	*ord_num <<oid>>*		Identifica una partición específica de un transcrito primario.
ElementTranscript	Representa los elementos de transcripción (exones/intrones) presentes en una partición del transcrito primario.	*ord_num <<oid>>*		Identifica un fragmento específico de todos los fragmentos de las particiones.
		star_position	int	Posición inicial de la secuencia en la partición.
		end_position	int	Posición final de la secuencia en la partición.
		sequence (derived)	string	Describe la secuencia de nucleótidos de la partición.
Exon	Es una especialización de la clase *ElementTranscript* . Representa a los exones.			
Intron	Es una especialización de la clase *ElementTranscript* . Representa a los intrones.			

SplicedTranscript	Representa as diferentes combinaciones de los exones presentes en una determinada partición del primario transcrito.	*ord_num <<oid>>*		Identificar una combinación de las diferentes combinaciones de exones.
		sequence (derived)	string	Describe secuencias de exones.
mRNA	Es una especialización de la clase *SplicedTranscript.* Representa al ARN mensajero.			
OthersRNA	Es una especialización de la clase *SplicedTranscript.* Representa a los diferentes tipos de ARN sin incluir al ARN mensajero.			
ORF	Describe las secuencias de nucleótidos que pueden codificar potencialmente una proteína.	*Id <<oid>>*		Funciona como identificador de la clase.
		sequence		Describe la secuencia codificante.
		start_position		Especifica la posición inicial de la secuencia del ORF en el ARNm.
		end_position		Especifica la posición final de la secuencia del ORF en el ARNm.
PrimaryPolypeptide	Representa la estructura primaria de la proteína descrita en la cadena de aminoácidos obtenida después de la traducción de un ORF.	*Id <<oid>>*		Funciona como el identificador de la clase.
		sequence		Almacena la cadena de aminoácidos.
Protein	Representa la cadena de aminoácidos que ha sufrido transformaciones químicas y el resultado final es una proteína funcional.	*name <<oid>>*		Funciona como identificador de la clase y al mismo tiempo describe el nombre de la proteína.
		sequence		Almacena la cadena de aminoácidos que describen una proteína.

Nombre de la vista:	**Genome View**
Objetivo:	Conceptualizar información relativa la composición estructural del genoma humano y registrar información genética de futuros pacientes o clientes del sistema.

Clase	Descripción	Atributos		
		Nombre	**Tipo**	**Descripción**
ResearchCentre	Representa el laboratorio o centro de investigación donde el genoma de un humano ha sido secuenciado.	*Name <<oid>>*	string	Funciona como identificador de la clase.
		description	string	Almacena una descripción del laboratorio o centro de investigaciones.
Genome	Representa a un genoma humano determinado que ha sido secuenciado completamente.	*Id <<oid>>*	short	Funciona como identificador de la clase y permite diferenciar los genomas secuenciados de diferentes humanos.
		description	string	Almacena información relevante sobre una determinada secuenciación.
Chromosome	Representa a los cromosomas presentes en el genoma secuenciado.	*Lumber <<oid>>*	short	Almacena el número de cromosomas de tal forma que determine un cromosoma específico para un genoma especifico.
		id_copy <<oid>>	short	Identifica cual de las copias del par homologo se refiere el cromosoma.
ChromosomicSegment	Representa a todos los segmentos que integran a un cromosoma.	*start_position <<oid>>*	long	Describe la posición inicial de la subsecuencia que un segmento.
		end_position <<oid>>	long	Describe la posición final de la subsecuencia que un segmento.
		sequence		Almacena la subsecuencia del cromosoma delimitada por las posiciones contenidas como valores en los atributos *start_position* y *end_position*.
GeneSegmet	Es una especialización de la clase *ChromosomicSegment.* Representa al material génico o región codificante, dentro del cromosoma.			
NonGeneSegment	Es una especialización de la clase *ChromosomicSegment.* Representa a los elementos que no forman parte de la región codificante, dentro del cromosoma.			
IntergenicRegion	Es una especialización de la clase *NonGeneSegment*. Representa el espacio entre las secuencias de los genes.			

ChromosomeElement	Es una especialización de la clase *NonGeneSegment*. Describe elementos dentro de los cromosomas que representan secuencias que pueden identificarse.			
Centromere, *Telomere* y *ORI*	Son especializaciones de la clase *ChromosomeElement* y tienen la intención de almacenar información para conservar la funcionalidad del cromosoma que no esta involucrada en la producción de proteínas.			

2. ***Descripción de las asociaciones entre las clases y restricciones de integridad presentes en el ECGH.***

Asociaciones de las clases del ECGH.

Nombre	Clases	Cardinalidad	Descripción
regulator/regulated	*Gene*	(1..N:1..N)	Relación de recursividad. Describe el hecho de que un gen puede regular otros genes y al mismo tiempo ser regulado por uno o varios genes.
Variant	*Allele - Gene*	(1..1: 1..N)	Relación de composición. Su intención es describir que un gen esta compuesto por uno o muchos alelos mientras que un alelo determinado solo puede estar asociado a un gen.
---	*DataBank - AllelicDataBankIdentification*	(1..N:1..0)	Relación de agregación. Agrupa los diferentes alelos que un determinado banco de datos ha identificado.
---	*DataBank - GeneDataBankIdentification*	(1..N:1..0)	Relación de agregación. Agrupa los diferentes genes que un determinado banco de datos ha identificado.
---	*Allele - AlleleDataBankIdentification*	(0..N:1..1)	Relación de agregación. Su intención es relacionar al alelo con su fuente de información externa y la forma en como esta fuente lo identifica.

---	*Gene - GeneDataBankIdentification*	(0..N:1..1)	Relación de agregación. Su intención es relacionar al gen con su fuente de información externa y la forma en como esta fuente lo identifica.
---	*AllelicVariant - AllelicReferenceType*	(1..1:0..N)	Describe el hecho de que una variante alélica tiene siempre un único alelo de referencia mientras que un alelo de referencia puede tener muchas variantes alélicas asociadas.
Denoted	*Variation - AllelicReferenceType*	(1..1:N..1)	Describe la agrupación de las variaciones alélicas asociadas a un alelo de referencia. Una variación tiene siempre un único alelo de referencia mientras que un alelo de referencia tiene una o muchas variaciones asociadas.
Changes	*Variation - AllelicVariant*	(0..N:1..N)	Describe la relación que existe entre la variación y su secuencia alélica variante conocida. una variación puede tener asociado uno o muchos alelos variantes, cuando esta variación es precisa, o bien puede no tener asociados alelos variantes. Por su parte un alelo variante siempre tendrá asociada una o muchas variaciones.
Obtained	*Variation - DataBank*	(1..1:0..N)	Su intención es conocer características de la fuente externa de la que se obtiene la variación.
Influences	*ChromosomicMutation - Chromosomic*	(1..N:1..1)	Relación de composición. Describe una variación alélica que afecta partes del cromosoma que está compuesta por una o muchas variaciones que afectan a varios genes. De tal forma que el número de instancias de la clase *Chromosomic* asociadas a la clase *ChromosomicMutation* dependerá del número de variantes alélicas de diferentes genes que intervengan en la variación cromosómica.
---	*TranscriptUnit - Promoter*	(0..1:1:1)	Relación de composición. Especifica que una unidad de transcripción puede tener o no una secuencia conocida para su promotor pero siempre tendrá un único promotor mientras que una secuencia de promotor conocida podrá ser promotor de varias secuencias transcribibles.
---	*TranscriptUnit - TranscribedSequence*	(1..N:1:1)	Relación de composición. Describe que una unidad de trascripción tiene una o muchas secuencias transcribibles asociadas pero una secuencia transcribible solo puede estar asociada con una unidad de transcripción.

---	*TrascriptUnit – Terminator*	(0..N:1..1)	Relación de composición. Introduce el concepto de que una unidad de transcripción puede estar asociada con varias o incluso ninguna secuencia de terminador.
---	*Allele - TranscriptionUnit*	(1..N:1..1)	Relación dependiente. Especifica que los alelos están compuestos por varias unidades de transcripción mientras que una unidad de transcripción solo pertenece a un alelo.
---	*TranscriptionUnit - RegulatorSequence*	(N..N:N..N)	Describe que una unidad de transcripción puede tener muchos segmentos reguladores que al mismo tiempo comparte con otras unidades de transcripción de varios genes.
Changed	*Genic - Segment*	(1..N:0..N)	La intención de esta asociación es conocer el segmento alélico en el que se produce la variación génica.
---	*Regulatory - Variation*	(1..1:1..1)	La intención de esta asociación es conocer la secuencia reguladora en la que se produce la variación.
---	*Gene - BibliographyReference*	(1..1:1:N)	Describe el hecho de que la información de las instancias de cada una de estas clases esta avalada bibliográficamente.
---	*Allele - BibliographyReference*	(1..1:1:N)	
---	*Variation - BibliographyReference*	(1..1:1:N)	
---	*TranscribedSequence - PrimaryTranscript*	(0..1:1..1)	Describe que una secuencia transcribible puede o no estar asociada con su correspondiente transcrito primario mientras que un transcrito primario siempre estará asociado a una secuencia transcribible.
---	*PrimaryTranscript - PrimaryTranscriptPath*	(1..N:1:1)	Relación de composición. Describe que un primario transcrito esta asociado a una o muchas particiones, impulsadas por el factor splicing, mientras que una partición solo puede pertenecer a un primario transcrito.
---	*PrimaryTranscriptPath - ElementTranscript*	(1..N:1:1)	Relación de composición. Describe que una partición esta integrada por uno o muchos elementos de transcripción mientras que un determinado elemento de transcripción solo pertenece a una partición especifica.
---	*Exon – SplicedTranscript*	(1..N:1..N)	Describe la participación de un determinado exon en las diferentes combinaciones de exones.

---	*Allele - SplicedTranscript*	(0..N:1:1)	Relación de composición. Esta relación tiene la intención de asociar a los exones producto de un proceso splicing con su alelo de referencia o bien con su variante alélica.
---	*Splicing - SplicedTranscript*	(1..1:0..1)	Relación dependiente. La intención de esta asociación es conocer en que combinación de exones, producto de un proceso splicing, se produce la variación determinada como mutación splicing.
---	*PrimaryPolypeptide - ORF*	(1..1:1..1)	Describe que un polipéptido primario solo esta asociado a un determinado ORF al mismo tiempo que un ORF solo da origen a un polipéptido primario
---	*Protein - PrimaryPolypeptide*	(1..N:1..1)	Describe que una proteína está formada por uno o muchos polipéptidos primarios pero un polipéptido primario solo participa en la formación de una proteína.
---	*ResearchCentre - Genome*	(0..N:1..1)	Relación de agregación. Describe el hecho de que un centro de investigación puede tener secuenciados genomas de diferentes humanos.
---	*Genome - Chromosome*	(1..N:1..1)	Relación de composición. Especifica que un genoma esta compuesto por un conjunto de cromosomas.
Couple	*Chromosome*	(1..1:1..1)	Describe el concepto de par homologo, lo que significa que cada célula humana tendrá dos cromosomas equivalentes (uno del padre y uno de la madre).
Formated	*Chromosome - ChromosomicSegment*	(1..N:1:1)	Relación de composición. Especifica que un cromosoma esta compuesto por uno o mas segmentos cromosómicos mientras que un segmento cromosómico solo pertenece a un cromosoma.
Corresponds	*GeneSegmet - Gene*	(1..1:0..N)	Describe que un segmento de la región codificante en un cromosoma determinado corresponde a un gen específicamente.
SegmentChanges	*GeneSegment - Variation*	(0..N:0..N)	La intención de esta asociación es determinar lo que ocurre en un segmento genico, es decir, que variaciones pueden ser identificadas dentro de una instancia de la clase *GeneSegment*.

Restricciones de integridad presentes en el ECGH.

No.	Expresión OCL	Descripción
R1	*context Segment inv:* *let ref: Allele::AlleleReferenceType* *self.Segments.sequence in ref.Segments.sequence*	Los segmentos alélicos solamente deben estar asociados con su alelo de referencia correspondiente.
R2	*context TranscriptionUnit inv:* *let ts: Segment::TranscribedSequence* *ts::self -> exists (p: Segment::Promoter::self | forAll(self.start_position) = p.end_position)* *ts::self -> exists (t: Segment::Terminator::self | t.start_position = ts::self.end_position)*	Todas las instancias de *TranscribedSequence* asociadas con la misma instancia de *TranscriptionUnit*, inician en la misma posición. Si *Promoter* existe en esta *TranscriptionUnit*, todas esas instancias de *TranscribedSequence* inician en la posición final de su promotor. Si *Terminator* existe en esta *TranscriptionUnit*, la posición final de *TranscribedSequence* es la posición inicial de este Terminator.
R3	*context Allele* *inv: self.units = collect(self.Segments)*	Un alelo puede estar asociado unicamente con unidades de transcripción que estan compuestas por segmentos alélicos asociados al mismo alelo.
R4	*context SplicedTranscript inv:* *let ex: PrimaryTranscriptPath::ElementTranscript::Exon* *forAll (ex.ord_num in self) = PrimaryTranscripthPath.ord_num*	Todos los exones asociados con una instancia de SplicedTranscription pertenecen a la misma partición representada por una instancia de *PrimaryTranscriptPath.*
R5	*context Chromosome* *self.couple1.number = self.couple2.number*	La pareja 1 (Couple1) tiene el mismo número de cromosoma que el número de cromosoma de la pareja 2 (Couple 2).

Anexo 3

Póster expuesto en el DILS 2009

Enforcing Conceptual Modelling to improve the understanding of Human Genome.

Aremy Virrueta
Research Center on Software Production Methods (ProS). DSIC. Universidad Politecnica de Valencia.

ProS Centro de Investigación en Métodos de Producción de Software

INTRODUCTION

Motivation:

It is widely known that the application of Conceptual Modeling techniques to the design and development of Organizational Information Systems leads to high-quality systems. In this context, the work presented here is carried out using the advantages of Conceptual Modeling applied to a different and challenging domain: The Human Genome.

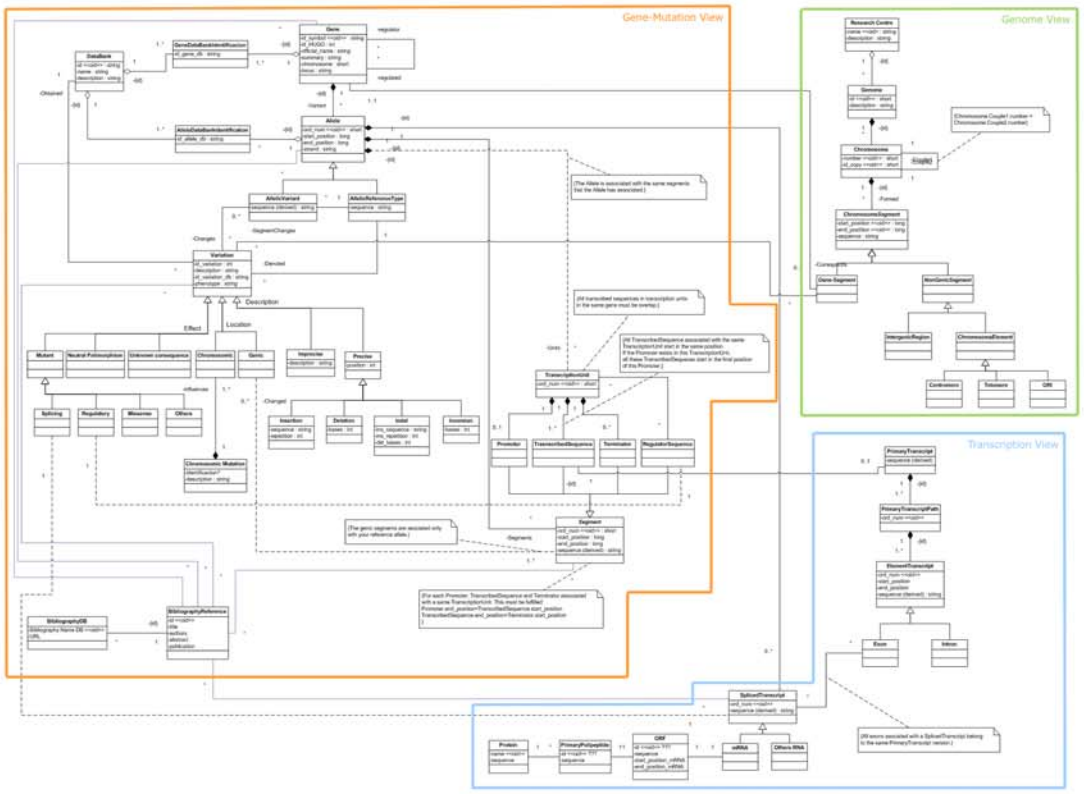
Proposal:

To study the relevant concepts of the human genome domain and group them in a conceptual model. This model will be a precise description of the field and will lead to a proper understanding of the domain.

Objetive:

To build a proper and complete conceptual model of human genome. This model enables to apply the advantages and good properties of the conceptual modeling techniques to software development implementations. These implementations will help the Molecular Biology researchers, specifically those studying the human genome.

CONCEPTUAL MODEL OF HUMAN GENOME



Gene-mutation view

This view represents the concepts about the gene's composition and the functional relations between genes and its mutations.

Transcription view

This view represents the concepts involved in the protein synthesis process.

Genome view

This view represents the concepts of the structural composition of the human genome.

Conclusion

In this work, human genome is treated as an information system through a conceptual modeling perspective. A deep study of this challenging domain produces a conceptual model, which is a clear and precise description of the human genome. Besides this, it allows to understand the global process from DNA sequences to fully functional proteins.

DILS 2009, University of Manchester, United Kingdom, 20th -22nd July.

Model Driven-Based Engineering Applied to the Interpretation of the Human Genome

Oscar Pastor[1], Ana M. Levin[1], Matilde Celma[1],
Juan Carlos Casamayor[1], Aremy Virrueta[1]and Luis E. Eraso[1]

Email: opastor@pros.upv.es

Abstract. In modern software engineering it is widely accepted that the use of Conceptual Modeling techniques provides an accurate description of the problem domain. Applying these techniques before developing their associated software representation (implementations) allows for the development of high quality software systems. The application of these ideas to new, challenging domains –as the one provided by the modern Genomics- is a fascinating task. In particular, this chapter shows how the complexity of human genome interpretation can be faced from a pure conceptual modeling perspective to describe and understand it more clearly and precisely. With that, we pretend to show that a conceptual schema of the human genome will allow us to better understand the functional and structural relations that exist between the genes and the DNA translation and transcription processes, intended to explain the protein synthesis. Genome, genes, alleles, genic mutations... all these concepts should be properly specified through the creation of the corresponding Conceptual Schema, and the result of these efforts is presented here. First, an initial conceptual schema is suggested. It includes a first version of the basic genomic notions intended to define those basic concepts that characterize the description of the Human Genome. A set of challenging concepts is detected: they refer to representations that require a more detailed specification. As the knowledge about the domain increases, the model evolution is properly introduced and justified, with the final intention of obtaining a stable, final version for the Conceptual Schema of the Human Genome. During all this process, the more critical concepts are outlined, and the final decision adopted to model them adequately is discussed. Having such a Conceptual Schema makes possible to create a corresponding data base. This database could include the required contents needed to exploit bio-genomic information in the structured and precise way historically provided by the Database domains. That strategy is far from the current biological data source ontologies that are heterogeneous, imprecise and too often even inconsistent.

Key words: Modelling, Human Genome

1 Introduction

It is widely accepted that in modern Software Engineering the use of conceptual modeling techniques applied to software development create higher quality

systems [1]. This is due to the fact that description and understanding of the problem domain is done before the implementation occurs, thus the work is carried out at higher abstraction levels.

Conceptual modeling techniques have been applied successfully to many different Information Systems domains from Business to medical applications. A well known example of that successful use is the one related with Organizational Systems [2]. The main idea is to provide a suitable conceptual framework, including possible, simple, clear and unambiguous definitions of the most fundamental concepts in the information system field and a suitable terminology for them. Notions as information and communication, organization and information system, etc.. are properly defined. The Information Systems (IS) experience –meaning by that all the efforts done by the IS in order to design, develop and manage Organizational Systems- has provided a relevant set of results, and it is sill a domain where a huge amount of academic and industrial work is in progress. But what could be considered to be the "next" for Conceptual Modeling? When exploring the world looking for new, challenging and suitable domains for the application of Conceptual Modeling techniques, one specific domain pops up that is surprisingly not included: Genomics, and in particular Human Genome proper interpretation. A main issue of this chapter is to show how well IS concepts and practices can work on this Human Genome domain.

It is widely accepted that the existence of a Conceptual Model improves the efficiency of Information Systems since it helps to manage modularity and evolution [1,3]. Traditionally, Software Engineering applied to Genomics field has been more oriented to the design of powerful and efficient search algorithms, working in the solution space rather than in the problem space. The complexity of the domain added to the explainable lack of knowledge that software engineers experience in this field may explain the absence of Conceptual Modeling contributions in the Bioinformatics domain. We mean by that, that too often the Bioinformatics work is based on heterogeneous repositories of information that contain inconsistencies, redundancies, partial information, different representation of the same data,... These type of problems are well-known and well-reported in the IS domain, and they have been intensively explored, especially in the Data Base context, where a relevant set of sound solutions has been provided during years. This is one of the reasons why we have created a multidisciplinary group in which software engineers with a strong IS background in Conceptual Modeling and biologists experts in Genomics worked close together in order to look for novel and effective solutions that illuminate the understanding of Human Genome. The result of this collaborative work has been the first ideas around a Conceptual Schema of the Human Genome [4].

To unify the knowledge needed to understand the Human Genome is a difficult task for software engineers. The conceptualization of the relevant concepts of these domains implies that software engineers change the way things are done. A big effort is required to fix concepts definitions in a context where even the knowledge associated with basic concepts is continuously evolving and changing. As we will see, even fundamental notions as the notion of gene are discussed in

deep when we want to have an exact definition. Often, the functional result of some partially unknown structure is just what characterizes that given structure. The subsequent conceptual challenge here is to model concepts taking into account the behavior of the domain elements. In this context, these elements are identified by their functional participation in the genome and not merely by their attributes and the relations between them. All the mentioned above implies a new way of conceptualize domains, where model evolution is a need, and where we could speak about a "fuzzy conceptual modeling" strategy, because many parts of the domain that it is being modeled, are being understood step by step and day after day. In this chapter we describe the Conceptual Schema evolution from its starting point to the present model, including the different versions needed to describe the domain precisely. Those concepts whose interpretation is more problematic are emphasized. In order to achieve our objectives, the chapter starts with an analysis of the most relevant related work; in section 3 an initial conceptual schema of the human genome is introduced to launch the discussion of how to build such a conceptual schema. After that, in section 4 a set of iterations is presented, intended to show and understand how the evolution of the most relevant, representative conceptual challenges guide the corresponding schema evolution process. The result of this discussion is the introduction of a current version of the conceptual schema of the human genome, which is ready to be properly exploited in an applied bioinformatic domain. Concluding remarks and the list of used references close the chapter.

2 Related Work

It is interesting to realize that we have not found too many relevant references where the Human Genome –or any other Genome- is faced from a IS Conceptual Modeling perspective. Even if the idea of applying conceptual modeling techniques on molecular biology has been tackled by some informatics and biologist in the last years, the approach has a lot of space to explore and discuss. The more relevant contributions in this field are those made by Paton et al. [5]. This work is an important reference for the labor developed in this chapter, because it can be considered as a starting point schema. In this proposal a collection of data models for genomic data is presented. These models describe elements involved in transcriptional and translational processes as well as the variant effects generated by them. As we will see through the chapter, the work presented here extends these ideas, and proposes a complete Conceptual Schema intended to be seen as a central, conceptual repository of Genomic information, which was not the objective in the referenced work.

Some other attempts to model more viable genomes have been reported. An interesting experience is provided by the e-Fungi initiative [14,15], where a systematic comparative analysis of fungal genomes is supported. The e-Fungi database integrates a variety of data for more that 30 fungal genomes and it provide fungal biologists with a powerful resource for comparative studies of a large range of fungal genomes. This work is developed in a different domain

–the fungi genome instead of the more complex human genome- but it shows a clear path of results exploitation that could be perfectly projected to our Human Genome modeling effort.

There are some other interesting examples of conceptual modeling techniques used in molecular biology, although with a more specific view on a particular part, as the work of Ram [6] to model the protein. This proposal, even if it is a little bit out of our scope, includes a part related to genome which can be very useful for the development of a conceptual schema of a complete genome. In any case, our attempt goes beyond the particular view of modeling proteins, because we want to provide a whole conceptual schema for the Human Genome.

Additionally, a relevant set of bioinformatic implementations are based in major or minor degree on conceptual modeling techniques, and they had been accepted favorably. One example of that is the work of Kevin Garwood et al. [7] which is a model-driven approach for the partial generation of user interfaces for searching and browsing bioinformatics data repositories. This work demonstrates that conceptual schemas can be used to produce many applications in the future. Again, when compared with our work the use of conceptual modeling techniques focuses on some very specific part of a software production process –the user interface design- in the bioinformatics domain, while we want to provide a whole, unified conceptual view in the tradition of the IS modeling experience.

These works are a few of the existing examples about the use of conceptual modeling in bioinformatics applications. They can be used to prove that conceptual modeling is an effective approach to help to improve biologic research. It is our belief that the work described in this chapter is an important contribution to the global understanding of the human genome, because only having a Conceptual Schema to characterize it, it will be possible to store the right contents, to manage them efficiently, and to understand the precise relationships existing between phenotype (external manifestation of human properties) and genotype (their corresponding genomic code).

3 An initial Conceptual Schema for the Human Genome

A conceptual schema of the human genome was proposed by Normal W. Paton in February 2000 [5]. (See Fig. 1).

In this model, an initial conceptual view of the human genome is described as a set of chromosomes divided in fragments that belongs to a certain gene, which is an intragenic fragmentation. Paton [5] proposes to classify a chromosomic fragment as either a Transcribed Region fragment or a Non-Transcribed Region fragment. In this model, a Transcribed Region fragment represents the transcribed sequence of the gene excluding elements that are also involved in the transcription process, like promoter o terminator. A Non-Transcribed Region fragment includes the regulatory sequences and the chromosomal elements that are part of the chromosome but are not involved in the transcription process. Additionally, in this model a set of primary transcripts can be generated

from a Transcribed Region fragment. We elaborated a new conceptual schema for the Human Genome, derived from Paton's model.

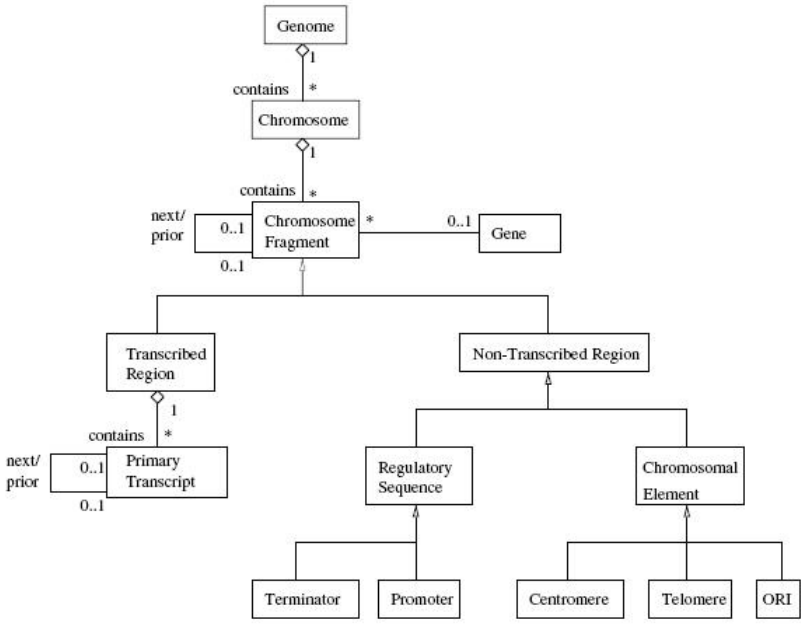


Fig. 1. Paton's conceptual model.

This model, presented in [4], provides a basis to fix the main features that are considered relevant to characterize the human genome basic components. The new conceptual schema introduces changes in the description of a genome.The main difference is the classification of chromosome segments. These chromosome segments are classified as Genic or Non–Genic, where a genic segment is seen as the composition of one promoter, one transcribed sequence, one terminator and many enhancer sequences in contrast with Paton's model where only the transcribed region in the gene is considered a genic segment. These genic components share a functional relation derived from the protein synthesis process, as the result of transcriptional and translational processes. Any other chromosomal sequence is considered as a non-genic segment (See Fig. 2).

4 Conceptual Schema Evolution

In this section the evolution "suffered" by the model will be presented. The most relevant changes and conceptual challenges will be discussed to understand the learning process from the initial model to the current stable conceptual schema

of the Human Genome. This is an interesting issue when applying Conceptual Modeling in this domain, because we have experimented how complicated fixing precise definitions can be in a context where new knowledge is discovered day after day. When modeling conventional organizational systems, its main components, their relationships and the functional processes are mainly known, and this makes the conceptual schema construction process viable and feasible. When facing the problem of modeling the human genome, we have seen how concepts and their representation can vary as the bio-genomic knowledge increases. In this section we will comment and discuss some relevant conceptual challenges, meaning by that the process followed to decide how to conceptualize some widely-known and used concepts of the human genome domain. As a very basic notion in this context, we will start with the notion of gene.

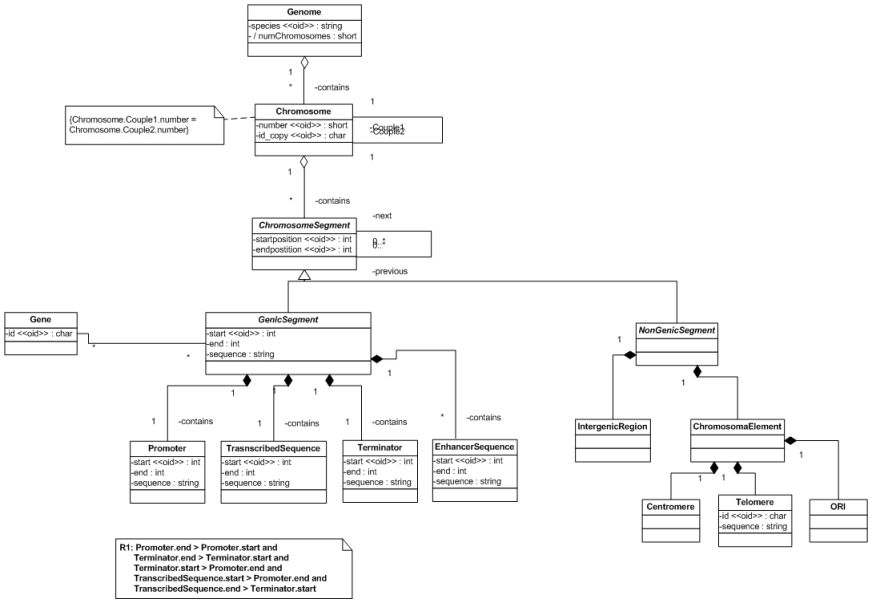


Fig. 2. An initial Conceptual Schema for the Human Genome.

The precise definition of gene is a very interesting topic that biologists still debate. The term has evolved, since it was first proposed by W. Johansen in 1909 based on the concept developed by Gregor Mendel, to fit the different paradigms and theories that have occurred in the history of Genetics (For a complete review see [10]). The classical view of a gene -a linear piece of DNA inherited from the parents who coded for a single protein- has considerably changed reflecting the complexity of the term. In the updated definition proposed by Gerstein in 2007 [10], a gene is "a union of genomic sequences encoding a set

of potentially overlapping functional products" and this evolution of the concept carries important implications inside.

Trying to explain such a fuzzy term to "conceptual modellers" is a very challenging and difficult task. During the conceptual modeling of the human genome, our group had to deal with many misunderstandings produced by the dramatically different way of facing concepts of biologist and software engineers. The fact that a gene sequence varies in different databases, the idea that two genes may share the same locus or that post-transcriptional events generate multiple products from one genetic locus are some of the concepts that puzzled software engineers. This reflects the changes that the concept had during the evolution of the model. In early versions the *Gene* class was associated to *Genicsegment* class, which was a big DNA segment composed by one promoter, one transcribed sequence and one terminator and regulated by many enhancer sequences. In the following versions *Genicsegment* class became smaller, generalizing any genic segment with some functionality (promoter, transcribed sequence, etc...). Then the *Gene* class was associated to *TranscriptionUnit* class, which was a really rigid structure that combined the genic segments involved in the transcription process. In later versions, the concept of *TranscriptionUnit* class became broader and multiple compositions appeared for a single transcription unit, in agreement with Gerstein's updated definition of *Gene*. This way we can model a concept whose definition is still discussed and assume that some of its characteristics may be ambiguous, unknown or even wrong.

The *TranscriptionUnit* class, as it appears in the third version of the model (see Fig. 3), evolved in such a way that allows for the inexistence of promoter and terminator sequences, represented by cardinality constraints (0,1) in *Promoter* and *Terminator* classes. This change is relevant, considering that these transcription unit elements exist but are frequently not indicated in the data repositories. However, at least a transcribed sequence must be present, represented by cardinality constraint (1..*) in *TranscribedSequence* class.

Following with the analysis of the domain, we found a new important concept that implies a change in the conceptual schema. An Enhancer sequence can be bound with certain proteins to enhance the transcription levels of the corresponding genes. This idea implies that enhancers are included in a group of regulator sequences that regulate the DNA transcription process, therefore the *Enhancer* class is replaced by *RegulatorSequence* class. Furthermore, if we take into account that a regulator sequence must be related with one or many transcription units then the *RegulatorSequence* class must be associated with *TranscriptionUnit* class. Besides, a regulator sequence is regulated by one or many regulator sequences.

At this point, the model only included the concept of gene and its structure from a transcriptional point of view. Once the Transcription and the Translation were studied in deep, a new conceptual schema was defined to include concepts and products related to them (see Fig. 4). For a more detailed description of these processes see [11].

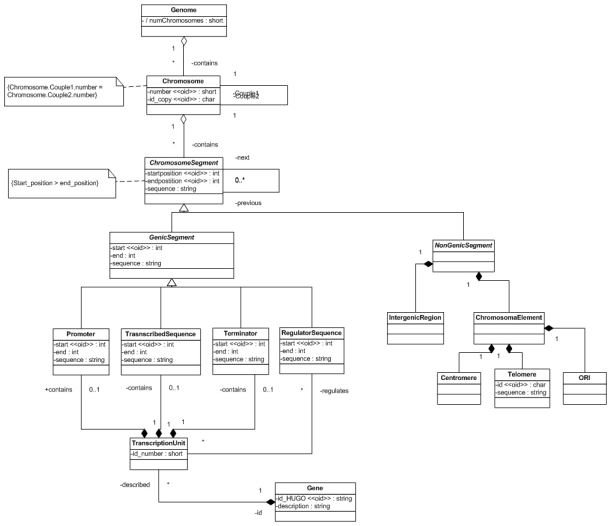


Fig. 3. Second version of the conceptual schema.

The protein synthesis process starts with the transcription: a DNA sequence is transcribed to a complementary RNA sequence. This RNA molecule is known as primary transcript. In the model, this new concept is represented by *PrimaryTranscript* class, which is associated with a set of instances of a new class: *ElementTranscript*. The specialization of this class in *Exon* and *Intron* classes describes the integration of exons and introns in the molecular structure of the primary transcript and represents the different types of partition fragments. A reflexive association *"Prior,Next"* for *ElementTranscript* class is added to indicate the order of exons and introns in a primary transcript sequence.

The splicing process consists in the combination of certain exons of the primary transcript resulting in a mature mRNA. The outcome of the splicing process application to a primary transcript is represented by the *SplicedTranscript* class that is specialized in the classes *mRNA* and other RNA types. The aggregation relation *"splice"* between *SplicedTranscript* class and *Exon* class allows to identify the exons that are the result of the splicing process.

The *mRNA* class represents the messenger RNA, a molecule that contains the information needed to synthesize a protein. The RNA sequence determines the amino acid order in the protein. The other classes include RNAs that are not necessarily translated into amino acid sequences: *snRNA* (small nuclear RNA, that participates in important nuclear processes), *tRNA* (transfer RNA, an important molecule in the transcription process) and *rRNA* (ribosomal RNA, which is part of the ribosome).

The next step in the protein synthesis is the migration of the mature mRNA from the nucleus to the cytoplasm. There it associates to the ribosome and the translation process starts. The translation is the production of proteins by

decoding of the mRNA produced in the transcription. In this process, the mRNA molecule acts as a template for the synthesis of the corresponding amino acid chain. The decodification rules are specified by the Genetic Code. Notably, the mRNA molecule is not translated completely; the ORF (Open Reading Frame) is the part of the mRNA sequence used in the translation process.

The *PrimaryPolypeptide* class is created to describe the protein primary structure: the amino acid chain obtained after the translation of an ORF. This amino acid chains suffers some chemical transformations and the final result is a functional protein which is represented in our model as *Protein* class. The association between *Protein* class and *PrimaryPolipeptide* class is included to conceptualize that a primary polypeptide originates a protein and a protein is synthesized as a primary polypeptide.

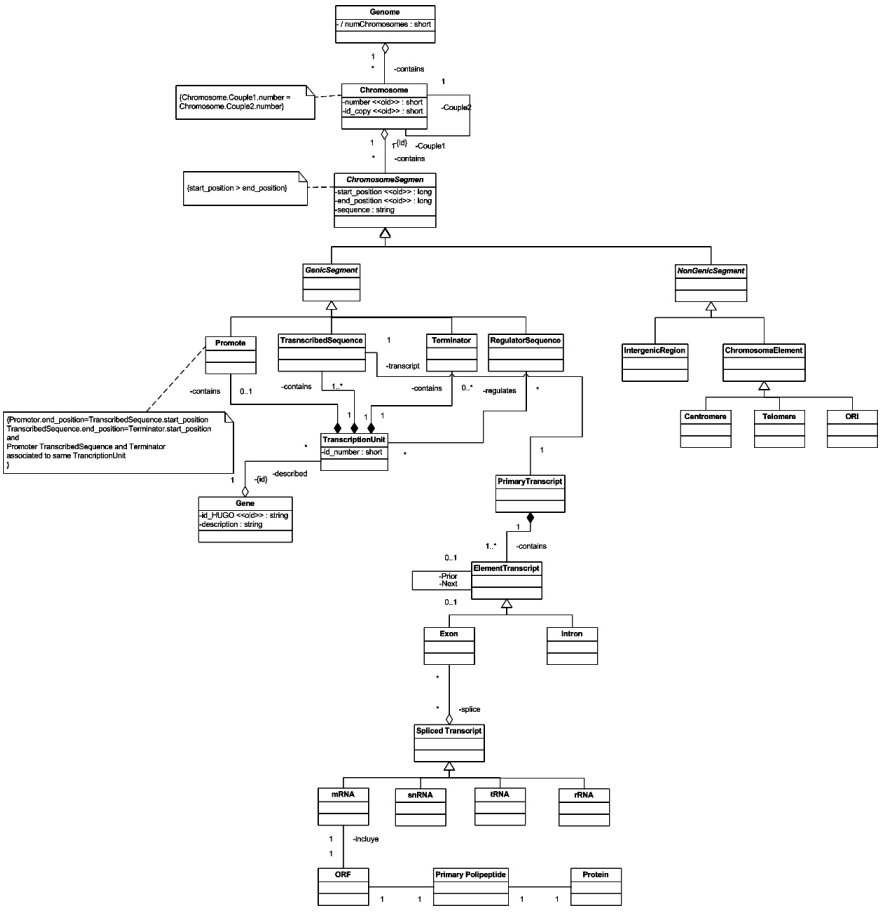


Fig. 4. Third version of conceptual schema.

Besides this new view, some constraints were defined in the *TranscriptionUnit* class. The *Promoter* class has an integrity constraint which describes the order of the elements included in the same transcription unit. This constraint specifies that the final position of the promoter sequence is always the initial position of the transcribed sequence and, at the same time, the final position of the transcribed sequence is always the initial position of the terminator sequence.

All the added classes and relations between classes listed below provide the bases for a more complete genomic model that, due to the size of the schema should be organized in two different views: the *Genome view* that includes the concepts from genome to transcription unit and the *Transcription and Translation view* that enclose all the concepts from the primary transcript to the fully functional protein.

As a result of the evolution of the discussions, some more important changes aroused. In previous versions of the schema, a gene was modeled taking into account its structure, which was considered fix. But in the new model, depicted in Fig. 5, the conceptualization of the gene is replaced by the idea of modeling a generic gene. At this point, the *Allele* class was introduced. An allele is each one of the alternative forms a gene may have in nature. These alternative forms are differenced by their sequences and any of them may produce changes on gene function. One of these forms is considered the wild type and is distinguished from any other variant allele for being the most abundant in nature. The generic genic structure is conserved independently of the alternative forms of a gene. A genic segment will be associated with generic genic segments. This idea allows conceptualizing the relation between the alternative forms of a gene and the gene structure. To explain the variation between alleles, we introduced a classification of the mutations occurred in any of these allele variants. The classification is made following two different criteria: the precision of the variation description and the phenotype that it produces.

The first classification of the variations is divided in two categories. The precise variations are those that are described accurately in the data bases. The imprecise variations are those that are described by text in the data sources, therefore it is not possible to detect the exact position of the variation.

The second classification of the allelic variants is divided in four types: 1) Genic Mutation, which names the variation that produces a pathologic effect. 2) Chromosomic Mutations, which describes the variation that affects more than one gene. 3) Natural Polymorphism, that characterizes a neutral variation, and 4) Unknown Consequence Changes, referred to those that report a variation with undiscovered consequences.

To represent in the model all the newly acquired concepts, many new classes were introduced. The first of them is the concept of reference sample. A *Reference* class is created and the aggregation relation *"belongs"* is included to describe that a genome has a reference.The idea of modeling a generic gene removes the specialization of *GenicSegment* class in *Promoter*, *TranscribedSequence*, *Terminator* and *RegulatorSequence* classes. This change is made because a genic segment is not generic.

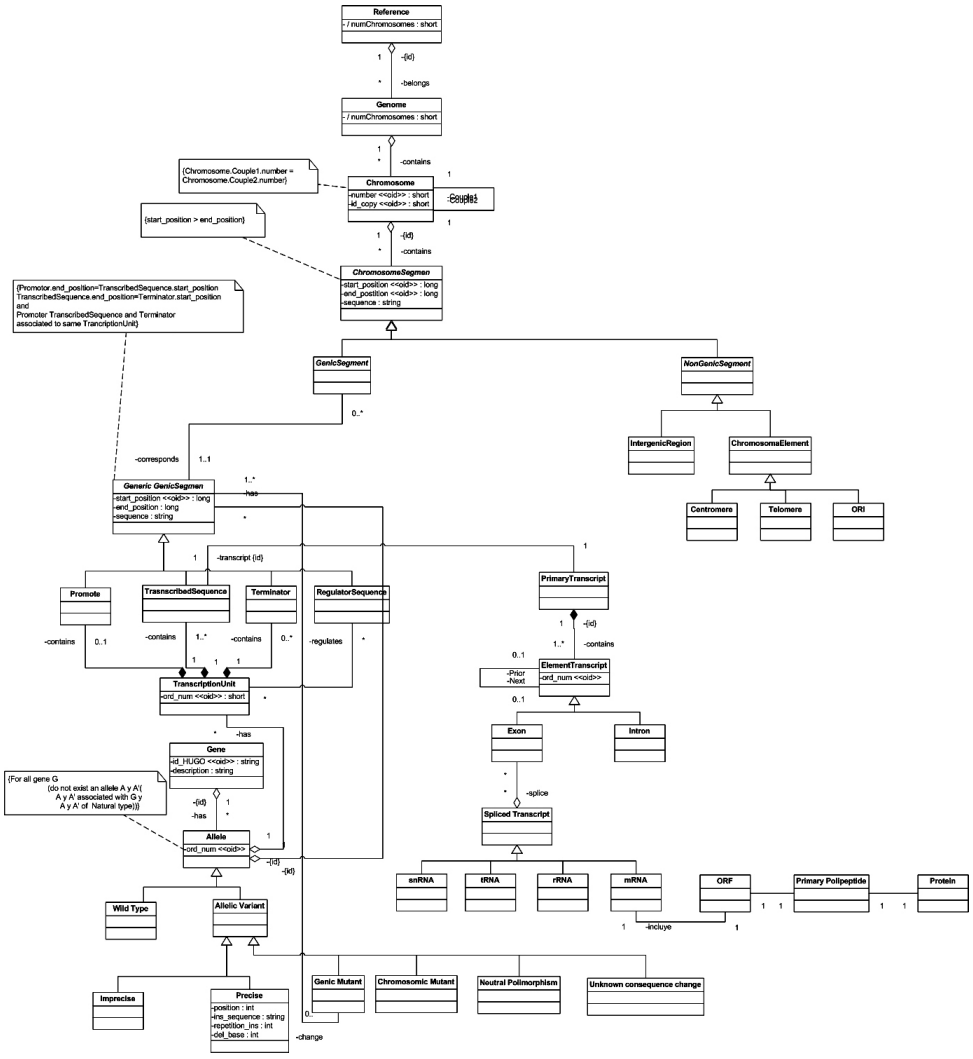


Fig. 5. Fourth version of conceptual schema.

The *GenericGenicSegment* class is added to the model to represent a generic genic segment. A genic segment can be associated to a generic genic segment, what is represented by the association relation *"corresponds"* between *GenicSegment* class and *GenericGenicSegment* class.

This relation describes the concept that a genic segment always corresponds to a generic genic segment and a generic genic segment can be associated (or not) to one or many genic segments.

The Allele class is added to the model to store the alternative forms of a gene found on the data repositories. Due to this, the gene is not associated anymore to the transcription unit but to alleles and the association relation between *TrascriptionUnit* class and *Gene* class is removed. The aggregation relation *"has"* between *Gene* class and *Allele* class is added to denote that a gene will always have one or many alleles associated. The aggregation relation *"has"* between *TranscriptionUnit* class and *Allele* class is included to specify that an allele is always associated one or many transcription units. In this context, other aggregation relation *"has"* between *Allele* class and *GenericGenicSegment* class is included in this model to describe that an allele has one or many associated generic genic segments.

Allele class is specialized in *WildType* class and *AllelicVariant* class. The *WildType* class represents the natural allele sequences and the *AllelicVariant* class represents the variant sequences that are not the wild type. The allelic variant groups are specified in the model by the specialization of *AllelicVariant* class in two groups of classes. The first group is specialized in *Imprecise* class and *Precise* class. The second group is specialized in *GenicMutant*, *ChromosomicMutant*, *NeutralPolimorphism*, and *UnknownConsequenceChange*.

5 The current version of Conceptual Schema

After the introduction of the main aspects related to the schema evolution, the current version of the Conceptual Schema of the Human Genome is presented in this section. It can be seen as the intermediate stable state that contains the current relevant information. at the same time the schema is ready for both evolving with the incorporation (or modification) of new conceptual issues, and being properly exploited through the load the adequate biogenomic contents using its corresponding database. In the sake of understandability, it is divided in three main views (*Gene-Mutation, Transcription and Genome*). Fig. 6 is the general view of the model and is intended to show the interconnections between each one of the different views that will be described in detail in the following subsections.

5.1 Gene-Mutation view

In the Gene-Mutation view presented in Fig. 7, all the knowledge about genes, their structure and their allelic variants based on the different public databases is modeled. The principal entities in this view are *Gene* class and *Allele* class.

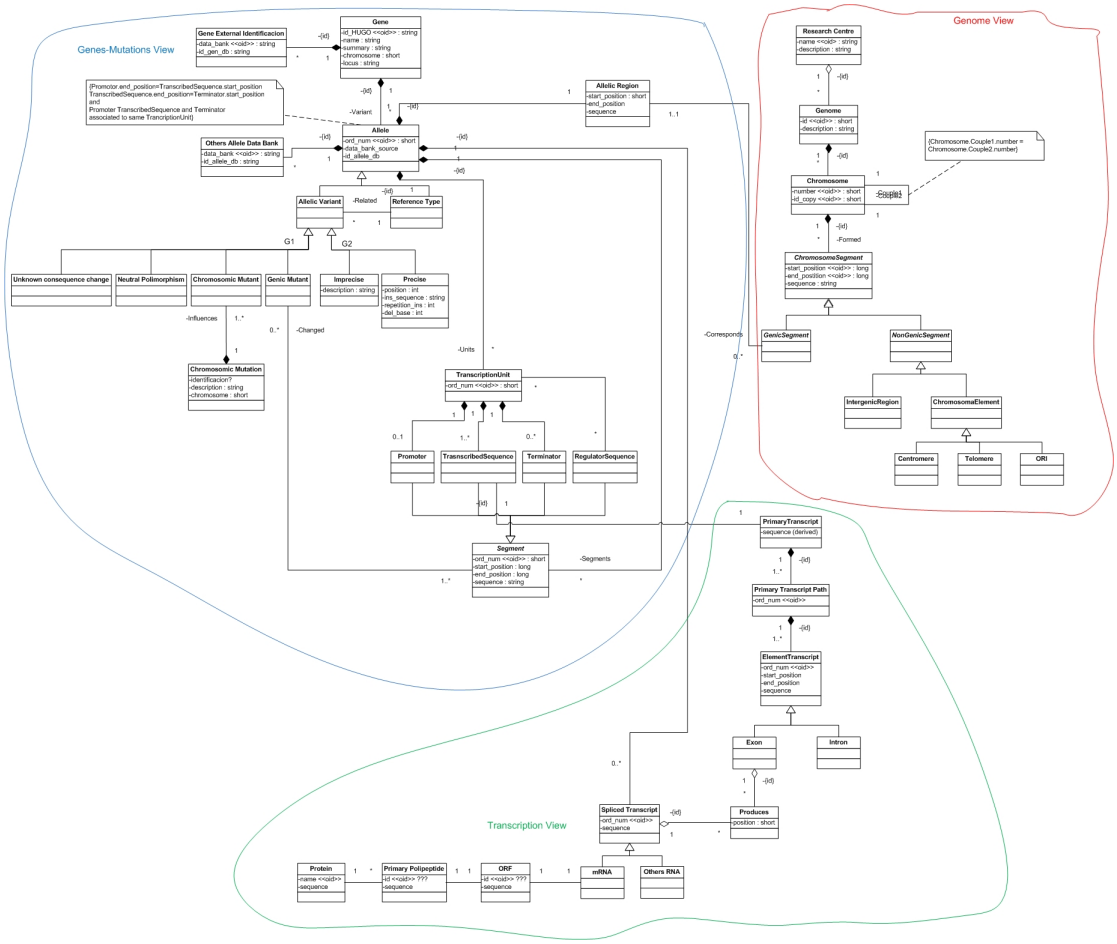


Fig. 6. General view of Conceptual Schema of Human Genome.

The *Gene* class models the concept of generic gene independently of the samples registered in the databases. In this class, we find attributes as *Id_ Hugo*, a numeric code that represents the universal code for the gene according to HGNC (Human Genome Nomenclature Committee) [12]. The *name* attribute stores the common name of the gene. *Chromosome* attribute represents the number of the chromosome where the gene is located and the locus attribute represents a gene location into the chromosome according to NCBI information [13]. Finally, a summary of the gene is extracted from NCBI database, and it is stored in the *summary* attribute.

Other main schema class is *Allele* class, which represents the instances of a generic gene and is the most important class of the model since all the information depends on it. This class contains all the relevant information about alleles such as allele databases, references, variants, generic genes and mRNA and DNA sequences. In the case of *Allelicvariant* class, all the information about the variation is represented as well as some *identification* and *descriptive* attributes. The *ord_ num* attribute is the internal identification number of the allele in our database. *Data_ bank_ source* attribute is the external code of the database source. Another important class is *Allelic Region*; this class represents the chromosomal region where the allele is, and contains an important attribute: *sequence*, which will store the complete DNA sequence of the allele. The attributes *start_ position* and *end_ position* will describe the beginning and the end of the allele in reference to the chromosome. The relation between *Gene* class and *Allele* class helps to identify any allele of a gene in the information system. The cardinality of this relation (1..1:0..N) allows to represent a gene with no allelic information.

To represent external references for gene and allele information, we created certain classes in the conceptual schema. The *GeneExternalIdentification* class represents the identification of a gene in different public databases. The *OthersAlleleDataBank* class, is the same representation but for alleles.

As far as we know, an allele might be considered reference or variant. For this reason, the *AllelicVariant* class and *ReferenceType* class are specialized classes from *Allele* class. The *ReferenceType* class represents the alleles used as references in the existing databases, and the *AllelicVariant* class represents allelic variations of a reference allele in a database. There is a related association between *ReferenceType* class and *AllelicVariant* class, this association allows us to represent a relation between a reference allele and its variations. It is important to note that the *WildType* class has been replaced by the *ReferenceType* class, a more suitable term for bioinformatic purposes since it does not have genetical meaning.

Once we propose an allelic variant, we determine a specialization hierarchy from the *AllelicVariant* class. This lead to the classification of allelic variants in two specializations. In G1, four different situations are considered: the allelic variation is specialized in *GenicMutant* when this variation affects a gene and it is specialized in *ChromosomicMutant* when the variation affects parts of the chromosome. The *ChromosomicMutation* class describes the chromosomic variation

that affects one or many genes in the same chromosome. *NeutralPolimorphisme* class is a variation that does not affect the phenotype. And finally the *Unknown-Consequence* class is used to represent the case when the variation consequences are not known yet.

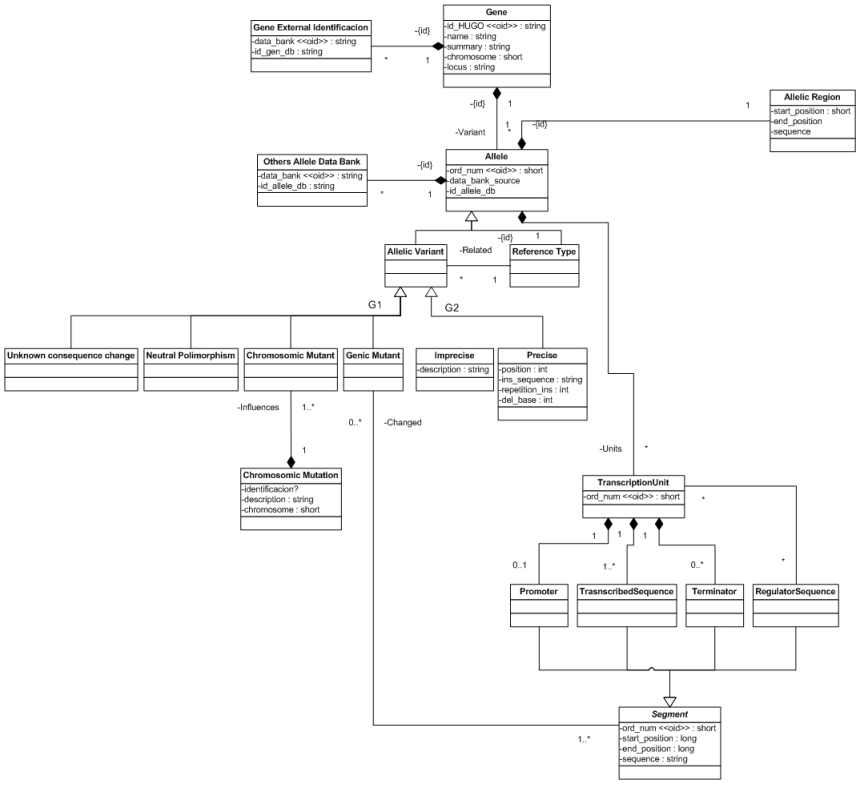


Fig. 7. Gene-Mutation View.

In G2, we classified the variation in two types: imprecise and precise. When we do not know details about the variation, we classify it as imprecise. There is a *description* attribute in the *Imprecise* class that stores information about the variation in text format. Opposite to that, when a variation is precise we can represent the position where the variation occurred.

In this view we also modeled the allele segmentation and structure. We considered that an allele segment has a significant and indivisible sequence of DNA. For this reason we implemented the *Segment* class, with attributes such as *ord_num*, (that identifies a segment between all the allele segments), *start_position* (initial position of the segment in a chromosome) or *end_position* (the end of the segment). The *sequence* attribute stores the sequence between *start_position* and *end_position*. This class has four specialized entities classi-

fied by their function in the transcription process. The first of these specialized classes is *Promoter*, which represents the region of the DNA sequence that facilitates the transcription of the transcription unit; *TranscribedSequence* is the sequence of DNA transcribed by the ARN polymerase; *Terminator* is a fragment of DNA that marks the end of the transcription process. Finally, the *RegulatorSequence* class is a segment with regulatory functions that regulates one or many transcription units.

After the definition of these specialized classes, a new class was created to model the concept of transcription unit. This class – *TranscriptionUnit* – has one attribute *ord_ num*, which is used for internal identification between all the transcription units of the same allele. The relations between this class and the specialized classes have different means. The relation between *TranscriptionUnit* class and *Promoter* class means that a Transcription Unit has a unique promoter; since this one may be unknown, this relation has cardinality 1..1:0..1. The relation between *TranscriptionUnit* class and *TranscribedSequence* class has cardinality 1..1:1..*. This means that many transcript sequences may exist in the same transcription unit starting all them at the same position. The relation between *TranscriptionUnit* class and *Terminator* class means that a transcription unit may have more than one terminator segment that can be unknown. The relation between *TranscriptionUnit* class and *RegulatorSequence* class means that a transcription unit may have many regulator segments, shared by different transcription units belonging to several genes in the most general case.

Regarding the precision of the DNA sequence in the schema, it is interesting to remark that initially the concept of allelic region was introduced to alleviate the inaccuracy of the DNA sequences that represent copies of the same gene in different databases. On the other hand, when the same databases were browsed to look for data regarding the fragmentation in different components of a gene (promoter, transcribed sequence and terminator) those data were precise and detailed. The *AlellicRegion* class indicates the approximate region where a gene is contained, whereas the *sequence* attribute of the *Segment* class gives clear and precise data information of the beginning and the end of the transcribed region.

5.2 Transcription View

Other important view of our model is the Transcription view that is showed in Fig. 8. In this view the basic components related to the protein synthesis process are modeled. This is a very important part of the schema, as it makes possible to link the genotype-oriented features modeled above with the phenotype-oriented manifestations that can be perceived externally. Consequently, it requires to explain the relationships between genes and alleles with their corresponding external expression in terms of behavior.

The first class found in this view is *Primary Transcript* class, which represents the RNA copy from the DNA of the transcribed sequence (*TranscribedSequence* class). This class has an attribute *sequence*, which is a derived attribute from the *Segment* class. We implemented the *PrimaryTranscriptPath* class, in order to model the different splicing factor-driven partitions of the *Primary Transcript*.

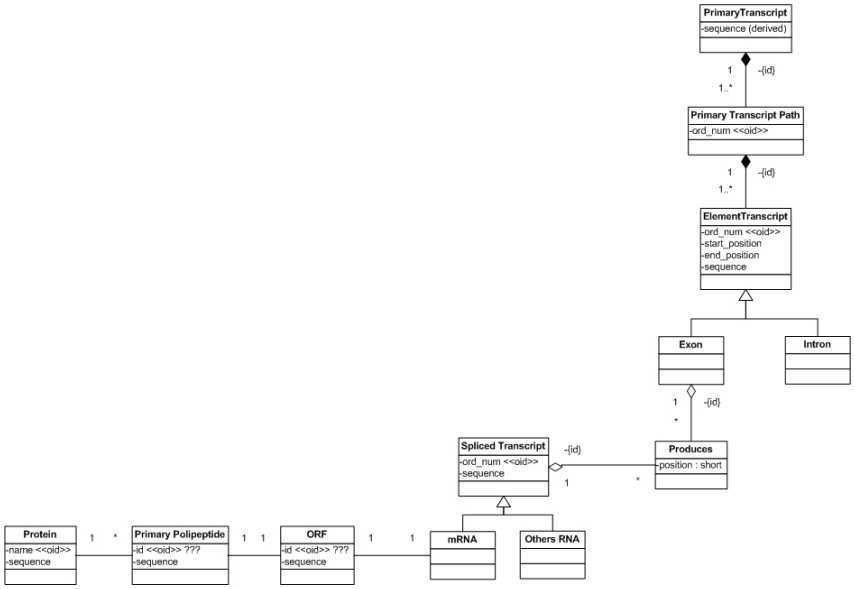


Fig. 8. Transcription View.

This class has an attribute *ord_num* used to identify a partition from all the partitions of a primary transcript.

The *ElementTranscript* class has an *ord_num* attribute which help us to identify a specific fragment from all of the partition fragments. The specialized *Exon* and *Intron* classes represent the types of the partition fragments.

The *Produces* class represents different exon combinations from a Primary Transcript that produces a Spliced Transcript. The result of these combinations is the specialized classes *mRNA* and *othersRNA* types. The Spliced Transcript class has the resultant sequence of the exon combinations in the *sequence* attribute and *ord_num* attribute for its identification between all the allele spliced transcripts.

The mRNA contains nucleotide sequences that could potentially encode a protein; this is what we know as ORF (Open Reading Frame). In our model, the *ORF* class has an *id* attribute as internal identification and a sequence attribute that stores codifying sequence. Then we have the *Primary Polypeptide* class, which describes the protein primary structure: the amino acid chain obtained after the translation of an ORF. This amino acid chain suffers some chemical transformations and the final result is a functional protein, which is represented in our model as *Protein* class. A protein could be formed by one o more Primary Polypeptides. In the *Protein* class we find a *name* attribute which represents the name of the resulting protein and its amino acid sequence in the *sequence* attribute.

5.3 Genome view

In Fig. 9, a conceptual schema of a complete individual genome is presented. This view is especially interesting for future applications, since massive parallel sequencing technologies will allow the complete sequencing of individual genomes at a very low price in the near future [8,9]. When this information becomes available we will be able to store it in our database. First we have a *Research Centre* class which represents the labs or research centres where a specific human genome was sequenced. We have a *name* and *description* attributes to record the genome source. Then *Genome* and *Chromosome* classes represent a complete genome sequenced by a specific research centre. A Genome is a set of chromosomes, this is represented by cardinality 1..* relation between *Genome* class and *Chromosomes* class. The *number* attribute will determine a specific chromosome on a specific genome. The *"couple"* relation on the *Chromosome* class represents the concept of homologue pair, which means that every human cell will carry two equivalent chromosomes (one from the father and one from the mother) with different alleles for each gene.

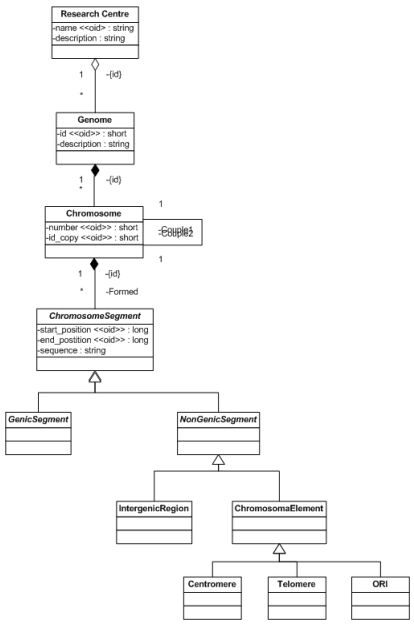


Fig. 9. Genome View.

The *ChromosomeSegment* class represents all the segments that form a complete chromosome. This class has as *sequence* attribute to store the corresponding DNA sequence delimited by *start_position* and *end_position* attributes. Other important concepts that we have represented in our model is the struc-

ture of a chromosome. We identified that a chromosome has two main types of segments: coding-related segments (represented by the *GenicSegment* class) and not coding-related segments (represented by the *NonGenicSegment* class); two classes are specialized from *NonGenicSegment* class: *IntergenicRegion* class, which represents the space between genes and *ChromosomalElement* class. The last one has three specialized classes that describe other elements of the chromosomes (*Centromere* class, *Telomere* class, and *ORI* class) whose function is to keep the chromosome functional and have nothing to do with protein production.

It is interesting to remark that, at the beginning of the modeling process we expected to have a unique, complete, reference description for each gene. This description should include DNA and RNA sequences, alleles, segmentations, mutations and every detail that could be potentially included in the schema. This is the main reason of the *GenericGenicSegment* class addition, which tries to manifest the uniqueness of the gene description. Later, conscious of the fact that different databases gave different definitions of the same gene, we chose to drop the idea of a "generic gene" to substitute it by the possibility to include different descriptions of the same gene offered by different databanks. As a result of this *OthersAlelleDatabank* class was included, as well as the *databank* attribute in the *Allele* class. This is another clear example of concept evolution driven by the learning process or what we previously called a Fuzzy Conceptual Modeling Strategy.

6 Conclusions

The way in which the knowledge of the human genome domain is captured and acquired by the IS experts is one of the richest part of this chapter. When biologists and software engineers interact, the underlied vocabularies and ontologies are too easily too far. To make them become as closer as possible is being a continuous challenge for the group of people working on this conceptual modeling of the human genome effort. In this chapter, we have tried to show how this evolution occurred, and how so basic notions as those of gene, genic segment, alleles, mutations or transcription, were refined step by step. Many times a definition accepted by everybody was changed after some time just because some vague detail was missed out in the first rounds. As at the end, a concrete proposal of conceptual schema is presented as a result of these discussions and interactions, we argue to provide with all the reported experience a very valid material for any other interdisciplinary group interested in facing the problem of understanding the human genome.

Another contribution of this chapter is directly related to the historical value recognized to conceptual modeling. The modeling benefits that biological systems research should get from IS theory and practice include at least:

– working at a higher abstraction level to specify systems easily and more precisely,

- making possible to reason about a biological system prior to its construction to foresee consequences in advance, and to facilitate simulation and validation,
- the real possibility of automating the subsequent systems development.

It is our belief that real practice in the genomic domain is also strongly requiring the IS experts participation. It makes sense from our IS perspective to talk about the chaos of genome data, meaning by it that there are currently tons of data from the genome publicly available, each one with its corresponding database that is defined with an specific schema, data format, identifications, etc.. and where the integration of the different sources is a very difficult task that sometimes is just not possible.

The proper management of this information ecosystem is an urgent need, because the available scientific literature and experimental data, the summaries of knowledge of gene products, the information about diseases and compounds, the informal scientific discourse and commentaries in a variety of forums, and many more genetic-related data is dramatically increasing day after day. To understand the relevant knowledge that is behind that huge amount of information, conceptual models should become the key software artifact. With our proposal of a Conceptual Schema for the Human Genome, including the way in which the model construction has evolved to reach the current situation, we believe to demonstrate that such a task is feasible and fully useful. This is why we have chosen Model Driven-based Engineering applied to the Understanding of the Human Genome as the chapter title: if models become the key artifact used to interpret and exploit the knowledge that is around the human genome domain, its understanding and subsequent number of successful practical applications will clearly increase.

Related with usefulness, we are now in the process of providing the adequate contents to the database that corresponds to the specified conceptual schema. We want to achieve a very simple but ambitious objective: if with Conceptual Models targeted at digital elements we can improve Information Systems Development, with Conceptual Models targeted at life we can directly improve our living. Our Conceptual Schema for the Human Genome is our modest contribution in that direction.

References

1. Olivé, A.: Conceptual Modelling of Information Systems. Springer-Verlag. Berlin-Heidelberg (2007)
2. Falkenberg, E., Hesse, W., Lindgrek.en, W., Nilsson, E., Han, J., Rolland, C., Stamper, R., Van Assche, F., Verrijn-Stuart, A., Voss, K.: A Framework Of Information System Concepts. IFIP. (1998)
3. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice. Springer-Verlag. Berlin-Heidelberg (2007)
4. Pastor, O.: Conceptual Modeling Meets the Human Genome. Conceptual Modeling - ER 2008. LNCS. vol.5231, p.1. Springer-Verlag. Berlin-Heidelberg (2008)

5. Paton, W.N., Khan, S., Hayes A., Moussouni, F., Brass, A., Eilbeck, K., Globe, C., Hubbard, S., Oliver, S.: Conceptual modeling of genomic information. Bioinformatics. 16, 6, 548–57 (2000)
6. Ram, S.:Toward Semantic Interoperability of Heterogenous Biological Data Sources. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS. vol.3520, p. 32. Springer. Heidelberg (2005)
7. Garwood, K., Garwood, C., Hedeler, C., Griffiths, T., Swainston, N., Oliver S., Paton, W.: Model-driven user interface for bioinformatics data resources: regenerating the wheel as an alternative to reinventing it. Bioinformatics. 7, 532, 1–14 (2006)
8. Bornberg-Bauer, E., Paton, N.: Conceptual data modelling for bioinformatics. Briefings in bioinformatics. 3, 2, 166–180 (2002)
9. Wheeler, D.A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A., He, W., Chen, Y., Makhijani, V., Roth, G.T., Gomes, X., Tartaro, K., Niazi, F., Turcotte, C.L., Irzyk, G.P., Lupski, J.R., Chinault, C., Song, X., Liu, Y., Yuan, Y., Nazareth, L., Qin, X., Muzny, D.M., Margulies, M., Weinstock, G.M., Gibbs, R.A., Rothberg, J.M.: The complete genome of an individual by massively parallel DNA sequencing. Nature. 452, 872–877 (2008)
10. Gerstein, M.B., Bruce, C., Rozowsky, J., Zheng, D., Du, J., Korbel, J., Emanuelsson, O., Zhang, Z., Weissman, S., Snyder, M.: What is a gene, post-ENCODE? History and updated definition. Genome Res. 17, 669–681 (2007)
11. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P.: Molecular Biology of the cell. Garland Science, New York (2002), http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=mboc4
12. Gene Nomenclature Committee, http://www.genenames.org
13. National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov
14. Hedeler, C., Wong, H.M., Cornell, M.J., Alam, I., Soanes, D., Rattray, M., Hubbrad, S.J., Talbot, N.J., Oliver, S.G., Paton, N.: e-Fungi: a data resource for comparative analysis of fungal genomes. BMC Genomics. 8, 426, 1–15 (2007)
15. e-fungi Project, http://www.cs.man.ac.uk/ cornell/eFungi/index.html