Estudio Preliminar del desarrollo de un Sistema de Información para el diagnóstico genético

UNIVERSIDAD POLITECNICA DE VALENCIA

PROS
Centro de Investigación en Métodos de Producción de Software

Ana María Martínez Ferrandis

Universidad Politécnica de Valencia

18/02/2011

Contenido

CONTENIDO **3**

ÍNDICE DE ILUSTRACIONES **5**

INTRODUCCIÓN **11**

1.1. MOTIVACIÓN 11
1.2. PLANTEAMIENTO DEL PROBLEMA 12
1.3. OBJETIVOS 13
1.4. SOLUCIÓN PROPUESTA 14
1.5. ESTRUCTURA DE LA TESINA 16

SISTEMAS DE INFORMACIÓN GENÓMICOS **17**

2.1. SISTEMAS DE INFORMACIÓN Y MODELADO CONCEPTUAL – GEIS 17
2.2. QUÉ ES UN GEIS 18
2.2.1. Necesidad de un esquema conceptual en un GeIS 19

ANTECEDENTES BIOLÓGICOS **23**

3.1. GENÓMICA 23
3.1.1. ADN y cromosomas 23
3.1.2. El gen y su estructura 25
3.1.3. El ARN 26
3.2. TRANSCRIPCIÓN DEL ADN 26

3.3. SÍNTESIS DE PROTEÍNAS, TRADUCCIÓN DEL ARN. **27**

3.4. MUTACIONES Y VARIACIONES **28**

3.5. RELACIÓN GENOTIPO – FENOTIPO **30**

ESTADO DEL ARTE **31**

4.1. ATLAS 31
4.1.1. Single record queries 33
4.1.2. Genome annotation 33
4.1.3. Inference of proteine-proteine interactions 33
4.1.4. Disease-gene associations 34
4.2. SIFT 34
4.2.1. SIFT Batch Protein 36
4.2.2. SIFT dbSNP 37
4.2.3. SIFT Blink 37
4.2.4. SIFT Sequence 37
4.2.5. SIFT Aligned Sequences 37
4.2.6. SIFT Genome 38
4.3. POLYPHEN – 2 38
4.4. ALAMUT 41
4.5. HOPE 46
4.6. MUTALYZER 51
4.6.1. Sequence Variant Description Generator Input 51
4.6.2. Sequence Variant Description Checker Input 52
4.6.3. SNP Converter Input 55
4.6.4. Batch Sequence Variant Description Checker Input 55
4.7. COMPARATIVA 55

EJEMPLO ILUSTRATIVO **59**

ALINEAMIENTO **67**

6.1. BLAST 70
6.2. CLUSTAL W 72
6.3. BLAT 73
6.3.1. Formato AXT *76*
6.3.2. Formato MAF *77*
6.3.3. Formato WU-BLAST *78*
6.3.4. Formato BLAST *78*
6.3.5. Formato PSL *78*
6.4. LECCIONES APRENDIDAS 81

BÚSQUEDA DE VARIACIONES **83**

7.1. VARIACIONES SINÓNIMAS 85
7.1.2. Solución planteada *87*
7.2. RELACIÓN GENOTIPO - FENOTIPO 91
7.3. LECCIONES APRENDIDAS 92

NOTACIÓN DE VARIACIONES **95**

8.1. HGVS 95
8.1.2. Recomendaciones generales *96*
8.2. TRADUCCIÓN 99
8.2.1. Algoritmo propuesto *102*
8.3. LECCIONES APRENDIDAS 113

PRUEBA DE CONCEPTOS **115**

9.1. PROCESO DE ANÁLISIS 115
9.2. ARQUITECTURA Y CLASES 117
9.3. ANÁLISIS DE SECUENCIAS 119
9.4. FILTRADO DE VARIACIONES 120
9.5. CONOCIMIENTO FENOTÍPICO Y FORMATO HGVS 121

CONCLUSIÓN **123**

BIBLIOGRAFÍA **127**

ANEXO: CÓDIGO DEL PROTOTIPO **129**

Índice de Ilustraciones

Ilustración 1 Componentes de un Sistema de Información 17
Ilustración 2 Esquema conceptual para representar las variaciones 20
Ilustración 3 De la célula al gen 23
Ilustración 4 Estructura cromosoma 24
Ilustración 5 Estructura de un gen y fases para la síntesis de la proteína 25
Ilustración 6 Proceso de transcripción (imagen modificada de Pearson Education) 26
Ilustración 7 Código genético, correspondencia entre codones y aminoacidos 28
Ilustración 8 Arquitectura Atlas 32
Ilustración 9 Algoritmo SIFT para la puntuación de AASs 35
Ilustración 10 Herramientas SIFT 36
Ilustración 11 Interfaz PolyPhen -2 38
Ilustración 12 Fases proceso PolyPhen-2 39
Ilustración 13 Listado de los estudios llevados a cabo 40
Ilustración 14 Resultado PolyPhen-2 40
Ilustración 15 Alamut, aplicación cliente-servidor 41
Ilustración 16 Selección del gen 42
Ilustración 17 Descripción de la interfaz de Alamut 43
Ilustración 18 Información detallada de SNP 43
Ilustración 19 Introducción de una mutación 44
Ilustración 20 Informe Alamut 45
Ilustración 21 Vista Splincing 46
Ilustración 22 Interfaz HOPE - Inserción secuencia proteínica 47
Ilustración 23 Interfaz HOPE - Selección de mutación 48
Ilustración 24 Interfaz HOPE - Informe resultado 49
Ilustración 25 Interfaz herramienta Sequence Variant Description Generator Input 52
Ilustración 26 Interfaz Sequence Variant Description Checker Input 53
Ilustración 27 Información transcritos y proteína 54
Ilustración 28 Información fenotípica 54
Ilustración 29 Traducción rs a formato HGVS 55
Ilustración 30 Fases de creación de un informe de diagnóstico génico 60
Ilustración 31 Investigación grupo Genoma 61
Ilustración 32 Funciones bioinformático 62
Ilustración 33 Introducción de datos 63
Ilustración 34 Fase de creación del informe 64
Ilustración 35 Diagrama de secuencia de la fase de creación del informe 65
Ilustración 36 Diagrama de secuencia para la fase de almacenamiento del informe 65
Ilustración 37 Alineamiento básico 67
Ilustración 38 Alineamiento global 68
Ilustración 39 Alineamiento local 69
Ilustración 40 Sección 1 de los resultados 71
Ilustración 41 Alineamiento BLAST parte gráfica 71
Ilustración 42 Alineamientos en BLAST, formato texto 72
Ilustración 43 Interfaz Clustal W 73

Ilustración 44 Interfaz web BLAT 74
Ilustración 45 Campos de la línea de resumen 77
Ilustración 46 Formato WU-BLAST 78
Ilustración 47 Formato PSL 79
Ilustración 48 Fórmulas de extracción de variaciones 79
Ilustración 49 Descarte de bloques de lectura 81
Ilustración 50 Esquema conceptual - Variaciones 84
Ilustración 51 Búsqueda de variaciones guiada por el modelo 85
Ilustración 52 Variaciones sinónimas 86
Ilustración 53 Borrado de un nucleótido 87
Ilustración 54 Posiciones derecha e izquierda a la repetición 87
Ilustración 55 Inserción de un nucleótido 88
Ilustración 56 Posiciones a derecha e izquierda de una inserción 88
Ilustración 57 Indel de un nucleótido 88
Ilustración 58 Eliminación de varios nucleótidos 89
Ilustración 59 Obtención de PD en el caso del borrado 89
Ilustración 60 Obtención de PI en el caso del borrado 89
Ilustración 61 Inserción de varios nucleótidos 90
Ilustración 62 Determinar posición derecha en el caso de la inserción 90
Ilustración 63 Determinar posición izquierda en el caso de la inserción 90
Ilustración 64 Indel de varios nucleótidos 91
Ilustración 65 Esquema Conceptual de la parte de Fenotipo 92
Ilustración 66 Secuencias de referencia (Fuente: página web de HGVS) 97
Ilustración 67 Esquema conceptual transcripción 100
Ilustración 68 Transformación de ADN hasta ADN codificante (imagen modificada de http://www.genome.gov) 101
Ilustración 69 Descripción a nivel de ARN 102
Ilustración 70 Indel en secuencia de aminoácidos 103
Ilustración 72 Cambio traducido a una inserción de un aminoácido 104
Ilustración 71 Inserción de 3 nucleótidos 104
Ilustración 73 Inserción con frameshift 105
Ilustración 74 Inserción con frameshift, caso 2 106
Ilustración 75 Borrado de un codón 107
Ilustración 76 Borrado parcial de codones 107
Ilustración 77 Borrado con frameshift 109
Ilustración 78 Indel sin frameshift 109
Ilustración 79 Indel sin frameshift 2 110
Ilustración 81 Indel sin frameshift - Paso 2 111
Ilustración 82 Indel sin frameshift - Paso 3 111
Ilustración 83 Indel con frameshift 112
Ilustración 84 Indel con frameshift, caso especial 112
Ilustración 85 Herramienta de análisis de mutaciones basada en CSHG 116
Ilustración 86 Arquitectura del prototipo 118
Ilustración 87 Resumen del diagrama de clases 118
Ilustración 88 Proceso de alineación 120

Ilustración 89 Proceso de clasificación de variaciones 121
Ilustración 90 Proceso de introducción del fenotipo y descripción en formato de HGVS..... 121

Agradecimientos

He de agradecer al Centro de Investigación en Métodos de Producción de Software (ProS) y a todos los miembros del proyecto Genoma por el buen ambiente de trabajo que hemos compartido, además del apoyo y colaboración ofrecidos para la realización de esta tesis.

Debo hacer una mención especial al Director del Centro ProS y mi Director de Tesis, Óscar Pastor, por darme la oportunidad de trabajar en este proyecto y por ayudarme en la realización de mi trabajo durante este tiempo.

Me gustaría hacer un agradecimiento especial a Francisco Valverde por todo su tiempo dedicado a trabajar conmigo en esta tesis y por sus múltiples y valiosas lecturas y correcciones, que me han ayudado a la realización de esta tesis de la mejor manera posible.

También querría agradecer a Ana Levin por ofrecerme su visión biológica que ha sido de gran ayuda en la realización de esta tesis y su ayuda en la corrección de los aspectos biológicos de la misma.

Quiero agradecer también a Verónica y Ainoha por sus consejos, tardes de confidencias, risas y lloros compartidos.

Mi mayor agradecimiento a mis padres y hermana, por haberme apoyado en todas mis decisiones, en toda mi trayectoria académica y durante toda mi vida. Sin su fuerza, cariño y apoyo nada de esto hubiese sido posible.

A Carlos, por estar conmigo durante todo este periodo de tesis y prestarme todo su apoyo. Por ofrecerme su hombro, su cariño y hacer que todo sea más fácil.

Al resto de mi familia y amigos por haber estado siempre que los he necesitado.

Por último, a mis profesores del máster por haber hecho que esta chica de Informática Industrial redescubriera la Ingeniería del Software y los Sistemas de Información.

Y, cómo no, a mis compañeros del máster, por haber compartido conmigo tantos inolvidables momentos durante el periodo que termina con esta tesis.

Introducción

1.1. Motivación

Actualmente vivimos en la sociedad de la información en donde la percepción del ser humano no se limita a su circunstancia en términos del bloque, barrio o ciudad en donde vive sino que se puede hablar de horizontes como mínimo a nivel planetario. Todo esto conlleva un flujo de información de carácter desorbitado.

En el mundo científico tanto flujo de información actúa como un arma de doble filo. La cantidad de información en la que buscar es inmensa, lo cual ayuda al investigador en su campo. Sin embargo, el problema reside en la criba y búsqueda de datos de calidad. Este hecho se complica cuando las cantidades de información de las que disponemos son tan grandes. Centrándonos en el ámbito de la bioinformática se puede observar cómo, debido a la secuenciación completa del Genoma Humano [1, 2] y al crecimiento explosivo de Internet, aparecen portales Web que pretenden facilitar a los investigadores el acceso a los datos genómicos y a herramientas bioinformáticas. De este modo se avanza con más velocidad en la generación de conocimiento.

A pesar de la gran cantidad de tareas en la bioinformática [3] para la mayor parte de éstas existen herramientas que proporcionan soluciones, lo cual es una gran ventaja. Sin embargo, el problema reside en que cada una de estas herramientas implementa su propio formato de datos lo que incrementa el trabajo del biólogo en el sentido de que habitualmente no ha de hacer una sola tarea, sino que debe procesar los resultados obtenidos para ser utilizados por otra herramienta. La mayoría de veces este traspaso de información se ha de hacer manualmente.

Por otra parte cada instituto de investigación genera resultados y los clasifica según sus necesidades y sus estándares, guardándolos en sus propios repositorios de información. Ejemplos muy conocidos de estos almacenes de datos son OMIM [4] y *Human Genome Mutation DataBase* (HGMD)[5] . Muchos de estos repositorios son de acceso libre, lo cual produce una gran oferta de información aunque sin un estándar asociado de acceso e interpretación.

En el trabajo de Stevens et al. [6] se definen cuáles son las tareas más frecuentes en bioinformática concluyendo que algunas de éstas son el alineamiento de secuencias, la búsqueda de coincidencias entre ellas y la búsqueda de bibliografía. Se destaca también el descontento que existe ente los investigadores en lo que a interoperabilidad en el manejo de datos se refiere, que es precisamente lo se ha comentado con anterioridad.

Teniendo en cuenta todos estos puntos se puede concluir que existen dos grandes problemas dentro del marco de la bioinformática: (1) la desestructuración de la información y (2) la utilización manual de diferentes herramientas para llevar a cabo un estudio.

Centrándonos en el primer punto, el de la desestructuración de la información, y llevándolo a otros ámbitos, como podría ser el de la administración pública, podemos encontrar ciertas similitudes. En ambos dominios nos encontramos con cantidades ingentes de datos, por ejemplo. Es por ello que obtener información relevante a partir del acceso, relación e

interpretación de dichos datos no es tarea fácil. Sin embargo, en el ámbito de la administración es frecuente la utilización de sistemas de información para manejar y trasformar estos datos en información útil. Un ejemplo muy conocido de su uso exitoso es el relacionado con sistemas organizacionales [7]. En consecuencia, es inevitable preguntarse el porqué del casi inexistente uso de los sistemas de información en un campo como el de la bioinformática. Lo cierto es que en un ámbito donde las ciencias de la computación se utilizan principalmente como mera herramienta ad-hoc para problemas determinados o puntuales, la idea de poder obtener un mayor beneficio a través de la utilización de las diversas técnicas disponibles en el campo de la informática ha pasado desapercibida. Es aquí donde encontramos la principal motivación de esta tesis de máster: abordar la integración de los sistemas de información dentro del campo de la bioinformática, específicamente en el área de la genética. Nuestra idea es describir lo que es un GeIS (*Genome Information System*) como un sistema de información diseñado específicamente para el ámbito de la genética capaz de manejar una gran cantidad de datos genómicos. Una vez descrito lo que es un GeIS, el segundo punto a tratar sería el de aplicarlo a un problema determinado, en este caso el diagnóstico génico a partir de la secuencia de ADN de un individuo, mediante técnicas avanzadas de Ingeniería del Software (IS).

1.2. Planteamiento del problema

Gracias a los avances llevados en el campo de la genómica y, en concreto, a los adelantos en la secuenciación del ADN, se ha llevado a cabo un aumento exponencial en la cantidad de datos genómicos generados por los investigadores. Muchos de estos experimentos se centran en intentar comprender la relación existente entre el genotipo (combinación y configuración de genes de un individuo) y el fenotipo (cómo se expresan dichos genes en una característica humana específica) de un individuo. Como consecuencia, la creación de bases de datos biológicas y de herramientas para la explotación de los datos producidos ha crecido drásticamente. Sin embargo, normalmente estas herramientas y bases de datos son definidas para una determinada área de investigación o un experimento concreto. Es por ello que lo más normal es que cuando un biólogo quiere utilizar dicha información, para llevar a cabo un experimento, éste se encuentre con que los requisitos de las herramientas que ha de utilizar o de las bases de datos que debe consultar no son los mismos que él necesita. Por tanto, el investigador debe dedicar mucho tiempo y esfuerzo para llevar a cabo un simple análisis. Esto es debido a que las herramientas bioinformáticas no se desarrollan siguiendo los principios de la Ingeniería del Software (IS), y por tanto dichas herramientas no se encuentran alineadas con los requisitos del usuario. Las consecuencias principales de esto son:

- Algunas de las bases de datos biológicas son tan sólo legibles por humanos, y por tanto no pueden ser procesadas adecuadamente de forma automática.
- La extracción de datos relevantes se complica ya que se encuentran repartidos en diferentes bases de datos.
- La integración y especificación de herramientas de *workflow* no es trivial, ya que es necesaria la utilización de diversas herramientas para llevar a cabo el análisis de los datos.
- La inclusión de nuevos estudios o de nueva bibliografía en las herramientas ya disponibles se convierte por tanto en una tarea difícil.

Por otro lado nos encontramos con que actualmente disponemos de herramientas hardware muy avanzadas, por ejemplo en la secuenciación de cadenas de ADN. Cuando el proyecto del Genoma Humano comenzó, la secuenciación de ADN implicaba laboriosos métodos. Hoy en día las tecnologías de secuenciación son mucho más fáciles, más rápidas y sobretodo mucho más baratas. El doctor Julian Parkhill, director de secuenciación en el instituto *Wellcome Trust Sanger*, expone que actualmente los investigadores son capaces de, mediante la utilización del nuevo hardware disponible, hacer cosas que hubieran sido inconcebibles hace dos o tres años. En la Tabla 1 podemos ver una comparativa de cómo el precio y el tiempo de la secuenciación ha ido disminuyendo según el hardware ha ido evolucionando. Sin embargo, a medida que esto sucede vemos como del mismo modo la longitud de la cadena que se lee se ve, lo cual puede conllevar un esfuerzo mayor a la hora de ensamblar los trozos de una secuencia.

En los últimos años han aparecido las secuenciadoras de tercera generación, también conocidas como *next generation sequencing* (NGS)[8], que son capaces de leer un conjunto de bases por segundo en lugar de leer una base cada veinte minutos. Se dice que el nuevo producto de *Pacific Biosciences*, *PacBio*, será capaz de leer secuencias de ADN de entre 1000 y 1200 bases a grandes velocidades, lo cual es un gran avance si observamos la relación entre los tiempos y las longitudes de lectura de la Tabla 1.

	Longitud de lectura (bases)	Tiempo de ejecución (días por gigabase)	Coste ($ por 1000 bases)
Capillary	1000	500	0,10$
454	454	2	0,02$
Illumina	75	0.5	0,001$
SOLiD	50	0.5	0,001$

Tabla 1Evolución de los secuenciadores

De lo anterior se extrae que en lo referente al hardware, en el campo de la bioinformática existe una velocidad de avance gigante. Sin embargo, por lo que respecta al software parece que esta existe una evolución, ésta es mucho más lenta. Es por ello que se precisa de la generación de nuevas herramientas software que sean capaces de lidiar con la gran cantidad de datos que las nuevas tecnologías producen.

1.3. Objetivos

El objetivo general de esta tesis es estudiar cómo debe desarrollarse un sistema de información dentro de un dominio tan actual y relevante como lo es la genómica. Dicho campo es excesivamente amplio y es por ello que este trabajo se centra en el estudio del campo de la genética, concretamente en la parte referida a las variaciones genéticas y sus efectos en el organismo humano. Para llevar a cabo dicho estudio se desarrollará un prototipo para el diagnóstico genético mediante el cual se obtendrá la problemática existente en el campo, pudiendo estudiar las diferentes formas de solucionarla mediante técnicas de Ingeniería del Software. Esta tesis justifica la necesidad de introducir tanto las técnicas de ingeniería del software como las de modelado conceptual en el campo de la genómica. De estos objetivos generales se extraen las siguientes preguntas, que son objeto de investigación:

1. ¿Cuáles son las características diferenciadoras de un Sistema de Información para el diagnóstico genético?
2. ¿Cómo seleccionar qué algoritmo de alineamiento es mejor para llevar a cabo un diagnóstico genético?
3. ¿Cómo deben localizarse las variaciones de forma precisa en una muestra de ADN?
4. ¿Cómo deben estandarizarse las variaciones detectadas para su posterior integración?

1.4. Solución propuesta

Las contribuciones principales de esta tesis responden respectivamente a cada una de las preguntas investigativas planteadas en el apartado de objetivos:

1. Los Sistemas de Información Genómicos se diferencian de los Sistemas de Información normales en varios puntos: (1) la necesidad de guardar datos extremadamente grandes, como pueden ser por ejemplo cromosomas, o secuencias de ADN, (2) la gran desestructuración de información que se encuentra en este ámbito, y (3) la continua ampliación del conocimiento del dominio, pues cada vez existen más herramientas que permiten llevar a cabo nuevos estudios y descubrir o redefinir conceptos. Algunos investigadores han propuesto [9] el desarrollo de sistemas de información genómicos (GeIS) para la resolución de las cuestiones descritas en el planteamiento del problema. En este trabajo se presenta un nuevo enfoque para el desarrollo de GeIS: el uso de modelos conceptuales que definan y organicen los datos genómicos de manera formal. El modelo conceptual expuesto en esta tesis ha sido resultado de la colaboración interdisciplinar llevaba a cabo con biólogos expertos en el campo y tiene como objetivo guiar el alineamiento de los conceptos de ambos campos.
2. A la hora de seleccionar un algoritmo de alineamiento de secuencias se deben tener en cuenta varios puntos: (1) el tipo de búsqueda que queremos realizar, (2) si es necesaria la obtención rápida de resultados y (3) los formatos de salida que queremos tratar. Uno de los principales problemas de las herramientas que proporcionan alineamiento de secuencias es que están diseñadas desde el punto de vista del problema, obviando las necesidades de los biólogos. Debido a esto, dichos algoritmos ofrecen soluciones al problema deseado, sin embargo no son óptimos en cuanto a la extracción de la información que ofrecen. Existen dos categorías de alineamiento de secuencias, la global y la local. Dependiendo de lo que se desee analizar en una secuencia se deberá optar por una categoría u otra. Un alineamiento global fuerza al alineamiento a ocupar la longitud total de todas las secuencias introducidas, mientras que un alineamiento local identifica regiones similares dentro de largas secuencias que normalmente son muy divergentes entre sí. Dentro de los alineamientos locales existen algoritmos como FASTA, BLAST o BLAT que utilizan bases heurísticas proporcionando soluciones rápidas y buenos resultados. Para lidiar con este problema se han estudiado las diferentes herramientas de alineamiento utilizadas en el campo del diagnóstico génico identificando sus pros y sus contras, así como ofreciendo una solución de cara a la implementación de GeIS.
3. Existen tantas fuentes diversas que contienen variaciones, que a la hora de realizar una búsqueda puntual la tarea se vuelve costosa debido a que cada fuente está estructurada de una manera y posee unos estándares diferentes de representación de

la información. Así pues la desestructuración consecuente de la información conlleva un problema importante a solventar. La necesidad de un modelo conceptual global, que unifique conceptos en este ámbito, es imprescindible. Por otra parte, la heterogeneidad en el mundo de la genómica no se da sólo en cuanto a las fuentes de datos de las que se disponen y la información disponible, sino que también nos topamos con cambios diferentes que tienen un mismo resultado. Nos referimos a cambios de nucleótidos que referencian a variaciones diferentes pero que tienen como resultado una misma mutación, lo que denominamos, en el contexto de esta tesis de máster, como variaciones sinónimas. Esto es debido a las repeticiones nucleotídicas en el ADN, y por tanto a la baja variabilidad en su población, así como a las diferentes técnicas de alineamiento disponibles para la extracción de variaciones genéticas. El caso de las variaciones sinónimas repercute tanto a la hora de documentar una variación como a la hora de buscarla, pues es posible que teniendo el mismo resultado que alguna otra variación documentada, la variación que se está estudiando en ese momento no corresponda con ninguna variación en el repositorio de información. Por tanto, puede que se esté o bien duplicando información, debido a la inserción de un mismo resultado, o perdiendo información, no devolviendo en un informe genético la información que sí se dispone. Para solventar el problema de la búsqueda de variaciones en el capítulo de variaciones de este trabajo se referencia la parte del modelo conceptual que casa con dicho punto, ofreciendo además una solución práctica al problema de las variaciones sinónimas.

4. En lo referente a la documentación de las variaciones, cada biólogo describe las variaciones encontradas en su estudio como cree conveniente. Esto origina una desestructuración en la información disponible, lo cual dificulta la búsqueda y extracción de datos de diferentes fuentes. Actualmente la sociedad *Human Genome Variation Society* (HGVS) ha redactado una serie de recomendaciones tanto para la nomenclatura de variaciones como para el contenido de las bases de datos de mutaciones. La discusión acerca del tema de la uniformidad en la manera de describir las variaciones genómicas data del año 1993, en la que se publicaron dos artículos [10, 11] que abordaban dicho tema. La documentación de variaciones a nivel genotípico o a nivel de ADN codificantes es trivial, sin embargo la traducción de dicha nomenclatura a nivel proteínico es compleja. Dentro de este punto podemos encontrarnos con casos como los *frameshifts*, que son variaciones que produce un cambio en la pauta de lectura de la secuencia. Por la naturaleza de tripleta de los codones la inserción o eliminación de nucleótidos puede cambiar lo que serían los grupos siguientes de codones que preceden al cambio (incluido él mismo), lo que produciría una traducción completamente diferente a la original. Pese a que el frameshift en general produce que la lectura de los codones después de una mutación se codifique de forma diferente, puede haber excepciones debido a la redundancia en el código genético. Además el codón de parada original no se leerá, pudiéndose adelantarse o atrasarse. La proteína que se cree podría ser anormalmente larga o anormalmente corta, y lo más probable es que no sea funcional. Como aportación se ha llevado a cabo un estudio de los problemas encontrados a la hora de formalizar la descripción de variaciones siguiendo las guías de HGVS, tratando también el problema de los frameshifts, y se ofrece una implementación práctica de la traducción automática de variaciones.

Como solución transversal se ha implementado un prototipo para validar las soluciones propuestas en este apartado.

1.5. Estructura de la tesina

A fin de cumplir con los objetivos descritos, la tesina se compone como se detalla a continuación. En primer lugar se especifica qué es un Sistema de Información Genómico y se ofrece un capítulo de antecedentes biológicos, necesario para entender los términos sobre los que se discute en este trabajo. A continuación se realiza un estado del arte de las herramientas utilizadas para el estudio de los diagnósticos genéticos y la explotación de datos de diferentes fuentes de datos. El quinto punto introduce un ejemplo ilustrativo del problema planteado, explicando las dificultades encontradas y ofreciendo una visión real de un ejemplo de tarea biológica: la creación de un diagnóstico genético. A continuación se pasan a comentar uno a uno los principales problemas encontrados a la hora de integrar un Sistema de Información Genómico. Los tres capítulos que van a continuación, siguen una estructura: estudio – problema – lecciones aprendidas. El sexto punto introduce el problema del alineamiento de secuencias, realizando un estado del arte de las herramientas actualmente empleadas para llevar a cabo dicha tarea. Se explica la problemática existente, ofreciendo una posible solución, y se determinan las lecciones aprendidas a partir de este punto. El séptimo punto comenta la problemática relacionada con la búsqueda de variaciones y más concretamente con lo que en este trabajo se denomina como variaciones sinónimas. En este capítulo se ofrece una solución mediante la utilización de modelos conceptuales y se añade un posible algoritmo para solucionar el caso concreto de las variaciones sinónimas. Se añade también la relación existente entre genotipos y fenotipos, y como a través de los modelos conceptuales esta asociación puede encontrarse fácilmente. Este punto finaliza mediante la explicación de las lecciones aprendidas en esta fase. En el capítulo octavo se habla del problema de la estandarización de la nomenclatura a la hora de describir variaciones. Se estudia la guía actual HGVS para este propósito, ofreciendo una explicación de la problemática encontrada al ponerla en práctica y a su vez se añade un estudio sobre cómo solventar los problemas encontrados, así como las lecciones aprendidas en este apartado. En el punto noveno de esta tesis se comenta la puesta en práctica de las lecciones aprendidas en los apartados anteriores mediante la implementación de un prototipo. Por último se detallan las contribuciones de este trabajo, las conclusiones, publicaciones realizadas y las líneas futuras de trabajo.

Sistemas de Información Genómicos

2.1. Sistemas de Información y Modelado Conceptual – GeIS

En todo sistema de información es condición necesaria aplicar la Ingeniería del Software según [12]. Uno de los aspectos más importantes de los que se encarga la Ingeniería del Software es el del modelado conceptual de sistemas de información, lo que podría denominarse como el diseño de estos sistemas. El concepto de sistemas de información apareció sobre los años 60. Pese a que puede considerarse un concepto bien establecido todavía es difícil ofrecer una definición precisa. Según [12] un sistema de información es un sistema que colecta, almacena, procesa y distribuye información, estando dicha definición constreñida a los denominados sistemas diseñados, es decir, sistemas que son diseñados y construidos por un ingeniero y nunca por un programador. Esta definición debe fijar también el tipo de información manejada por estos sistemas, haciendo referencia dicha información a un estado de un dominio, también llamado objeto del sistema o universo del discurso.

Los componentes de los sistemas de información son tema de debate. Asumiendo una visión de base de datos se podría decir que todo sistema de información se divide en una capa de aplicación y una base de datos (**¡Error! No se encuentra el origen de la referencia.**). La capa de

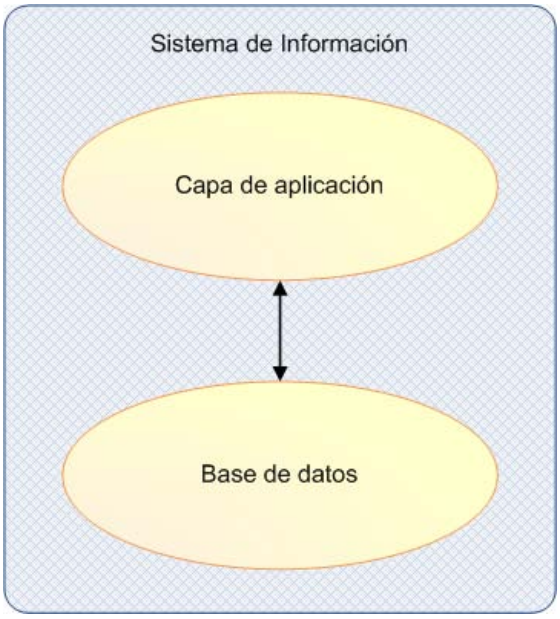

Ilustración 1 Componentes de un Sistema de Información

aplicación se compone de interfaces que permiten a las aplicaciones y al usuario interactuar con los datos. La base de datos es la encargada de almacenar dichos datos.

Las funciones que tiene todo sistema de información, según [12] son 3: (1) memoria, (2) informativa y (3) activa.

La función de memoria tiene por objetivo mantener una representación interna del estado de un dominio. Esto correspondería a la base de datos o base de información, pudiendo ser esta temporal o no temporal.

El objetivo de la función informativa sin embargo radica en llevar a cabo acciones que cambien dicho estado del dominio, pudiéndose ejecutar como petición del usuario o de una forma autónoma.

Por último, la función informativa es la encargada de proporcionar a los usuarios información acerca del estado del dominio. Normalmente el usuario realiza una consulta y el sistema la responde.

De todo esto se concluye que para que cualquier sistema de información funcione como debe, se ha de tener conocimiento sobre sus componentes, su dominio y las funciones que éste debe llevar a cabo. En el campo de los sistemas de información esto es lo que se denomina esquema conceptual, y es el producto final de todo modelo conceptual.

2.2. Qué es un GeIS

Podemos definir un Sistema de Información Genómico (SIGe o GeIS en inglés [*Genomic Information System*]) como todo aquel sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar información genómica. Esto se lleva a cabo mediante la integración organizada de hardware, software y datos genómico diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas dicha información genómica. Las diferencias existentes con un Sistema de Información normal son:

- La necesidad de almacenar datos extremadamente grandes, como lo pueden ser las secuencias genómicas.
- La desestructuración de información existente en el dominio de la genómica.
- La continua evolución del dominio.

Todo GeIS debe disponer de un sistema automático de extracción y actualización de información de diferentes fuentes de datos. Del mismo modo, en todo GeIS los métodos de búsqueda y recuperación de datos han de ser eficientes y fáciles de usar, permitiendo la realización de preguntas combinadas que de otra forma serían difíciles de ejecutar y contestar.

Todo GeIS bien formado debe definir y contener la estructura básica de un gen y sus características con el mayor detalle posible, dentro de estos elementos caben destacar: los alelos que lo definen, los intrones y exones de cada uno de sus transcritos, así como las variaciones que pueden estar asociadas a cada uno de ellos, su fenotipo asociado y la bibliografía en la que se registraron dichas variaciones.

El punto clave en un GeIS es la integración de diversa información a un formato común, concordante con la estructura comentada en el párrafo anterior, válido para ser utilizado en otros proyectos incrementando la utilidad de las bases de datos de estos sistemas de información comparándola con la utilidad ofrecida por los buscadores web.

Los GeIS son útiles tanto para bioinformáticos como para estudiantes que busquen relaciones entre datos genómicos, como por ejemplo relaciones del tipo genotipo-fenotipo, de modo que dichos estudios se pueden llevar a cabo de manera más sencilla.

2.2.1. Necesidad de un esquema conceptual en un GeIS

El esquema conceptual es la representación de los requisitos informales de un sistema de información en términos de una descripción formal y completa, pero independiente de todo criterio de implementación. Esta descripción formal se obtiene utilizando un modelo conceptual cuyo resultado es el esquema conceptual.

El esquema conceptual siempre ha de existir en el desarrollo de sistemas de información, en lo único en que puede variar es en su forma [12]. Un esquema conceptual puede por ejemplo ser mental y existir tan sólo en la mente del diseñador, o puede ser explícito. Este último se construye mediante un lenguaje de modelado como puede ser UML [13] o el modelo entidad relación [14].

El esquema conceptual proporciona la integración necesaria entre los componentes del sistema de información, sin esta integración el sistema de información se vería fragmentado y el reto de mantener la consistencia del sistema se dejaría a cargo de los actores humanos. Desafortunadamente es bien conocido que la intervención humana es costosa, se convierte en una fuente continua de errores y lleva a la creación de sistemas de baja calidad.

Tanto la modularidad como la evolución son dos de las propiedades básicas de un sistema de información, y la existencia de un esquema conceptual subyacente a este sistema ayuda a incrementar eficientemente su gestión. Ignorar las técnicas de modelado conceptual a la hora de diseñar un sistema de información genómico provocaría que todos los problemas, que se producen en los sistemas de información convencionales, se trasladasen a este dominio.

Es por ello que en este capítulo se presenta el esquema CSHG [15, 16], comentado en el capítulo de especificación de requisitos más adelante. El principal objetivo de este esquema conceptual es el de establecer una conexión entre el campo genómico y el dominio del desarrollo de sistemas de información.

Una de las características principales del dominio genómico es su heterogeneidad. Unificar los conceptos principales de este campo no es una tarea sencilla. Además existe la complicación de que el conocimiento en este campo se encuentra todavía en desarrollo y sus conceptos están continuamente evolucionando, lo cual complica la organización de toda la información genómica disponible.

Las bases de datos genómicas se ven afectadas por el problema de la heterogeneidad. En este campo, cada base de datos captura los conceptos según la interpretación y la terminología de un biólogo. Sin embargo, existen diferentes definiciones para el mismo concepto; por ejemplo, una variación en una secuencia de ADN puede ser referenciada bajo los términos: variación, mutación, polimorfismo o SNP [17]. Pese a que todos estos términos representan más o menos el mismo concepto, existen ligeras diferencias entre ellos. El problema de la representación heterogénea de datos puede solucionarse mediante el uso de modelos conceptuales, como algunos trabajos proponen [18]. El desarrollo de un modelo conceptual para representar el genoma humano es una aproximación útil para entender un dominio tan complicado como el de la genómica, ya que se definen conceptos precisos y se relacionan entre sí. Si en un futuro se descubren nuevos conceptos, relaciones o cambios, éstos podrán incorporarse al modelo fácilmente.

El esquema conceptual que se presenta aquí es preciso con los conceptos genómicos y los principios de los sistemas de información, ya que ha sido desarrollado por ingenieros de software y biólogos especializados en el campo genómico. El esquema presentado en esta sección se centra en la descripción de variaciones genómicas, siendo un extracto del esquema conceptual CSHG.

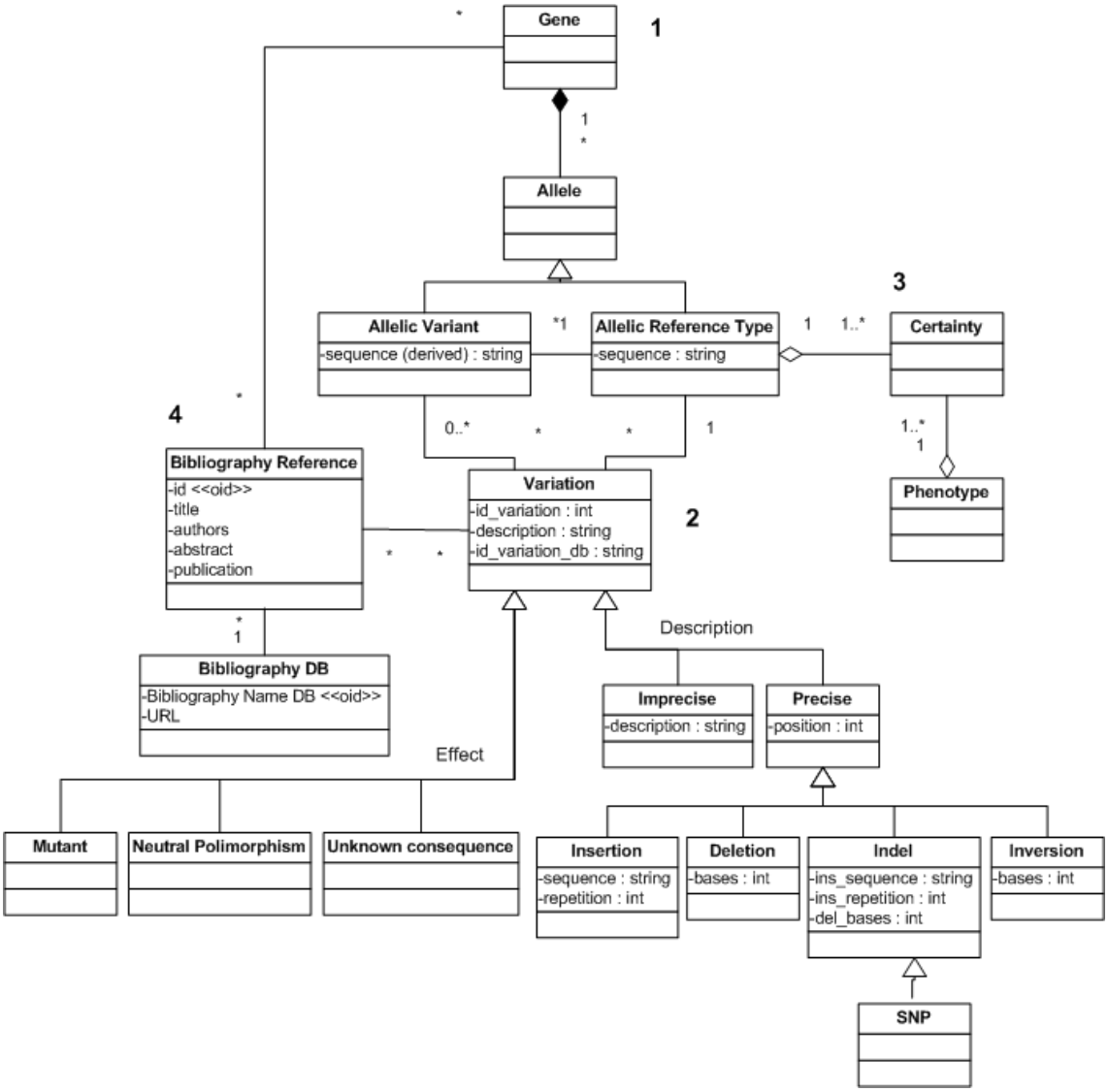

Ilustración 2 Esquema conceptual para representar las variaciones

En la Ilustración 2 se muestra el esquema conceptual propuesto. En la parte superior de la figura se definen las clases (1) *Gene* y *Allele*. La entidad Gene modela el concepto genérico de gen mientras que la entidad Allele representa las instancias individuales de un gen. La clase *Allele* tiene dos especializaciones: *Allelic Reference Type* y *Allelic Variant*. La primera especialización modela la secuencia de referencia que deifene un gen universal que será utilizado para propósitos comparativos. Estas secuencias de referencia son extraídas a partir de fuentes de datos confiables como RefSeqGene. La segunda especialización representa una secuencia de ADN de un individuo que contendrá multitud de variaciones con respecto al alelo de referencia.

Cada variación encontrada a partir del proceso de comparación llevado a cabo sobre una secuencia, es modelada mediante la entidad *Variation*. Esta entidad almacena todas las variaciones documentadas en la literatura genómica que están asociadas a alguna enfermedad o a cambios normales debidos a la naturaleza intrínseca de un individuo. A su vez, dicha entidad tiene asignados dos grupos diferentes de especialización. El primero corresponde a la descripción de la variación y se compone de dos especializaciones: la entidad *Precise*, que definen aquellas variaciones de las que se puede representar la posición donde ocurrieron, y la entidad Imprecise, que hace referencia a aquellas variaciones cuya posición no está especificada y de las que no tenemos detalles. Las variaciones precisas se clasifican a su vez en cuatro entidades, dependiendo del tipo de cambio que se ha originado: a) *Insertion*, son aquellas variaciones en las que ha habido una inserción de nucleótidos con respecto a la secuencia de referencia, b) *Deletion*, hace referencia a los cambios compuestos por la eliminación de algún nucleótido con respecto al alelo de referencia, c) *Indel*, son aquellas variaciones en las que se ha producido una inserción y a su vez una eliminación de nucleótidos, y d) *Inversion*, que vienen dadas por aquellas variaciones en las que ha habido una inversión de posición de nucleótidos. Un indel puede ser categorizado como un SNP si ocurre al menos en un 1% de la población. El segundo grupo consiste en tres especializaciones: *Mutant*, que representa a las variaciones que están relacionadas con alguna enfermedad, *Unknonw Consequence*, hace referencia a aquellas variaciones cuyo efecto no ha sido todavía descubierto, y *Neutral* polimorphism, que son las variaciones de las que se conocen sus consecuencias pero no están asociadas a ninguna enfermedad.

Una variación especificada en nuestro esquema tiene siempre asociado un fenotipo, que está modelado por la entidad *Phenotype* (3). La entidad *Certainty* especifica la probabilidad de que un fenotipo pueda mostrarse a causa de una variación concreta en un genotipo. En caso de que se encuentre una asociación entre un genotipo y un fenotipo, es esencial saber información sobre la referencia bibliográfica y la base de datos original en donde el descubrimiento fue descrito. Esta información se representa a través de las entidades *Bibliography Reference* y *Bibliography DB* respectivamente (4).

Así pues el esquema aquí presentado mejora la descripción del dominio, en este caso centrado en el proceso que va desde la secuencia de ADN a las variaciones que esta secuencia pueda tener y las implicaciones que éstas pueden tener en el fenotipo expresado a partir de ese genotipo.

Antecedentes biológicos

Dado que este trabajo se encuentra entre dos disciplinas tan diferentes como la informática y la biología, es innegable la necesidad de la existencia de un capítulo que aborde los temas biológicos más relevantes a la hora de conseguir una comprensión completa de los temas aquí tratados.

3.1. Genómica

Todos los organismos están compuestos de una o más células, en concreto el cuerpo humano está formado por 10 billones de éstas. La célula es la unidad morfológica elemental y funcional de los seres vivos. Las células desarrollan funciones específicas que en conjunto hacen posible que el organismo del individuo que constituyen funcione correctamente [19]. Cada célula posee una zona llamada núcleo en donde se almacena toda la información genética de un organismo, dicha información se encuentra en el genoma.

El genoma (Ilustración 3) se compone del conjunto completo de moléculas de ADN que componen un organismo y su tamaño puede variar de un organismo a otro. El genoma más pequeño conocido de un organismo de vida libre (una bacteria) contiene alrededor de 600.000 pares de bases. El genoma humano, ese gran libro de la vida que contiene las instrucciones que determinan las características físicas y en parte psicológicas e intelectuales del individuo, tiene alrededor de 3 millones de bases.

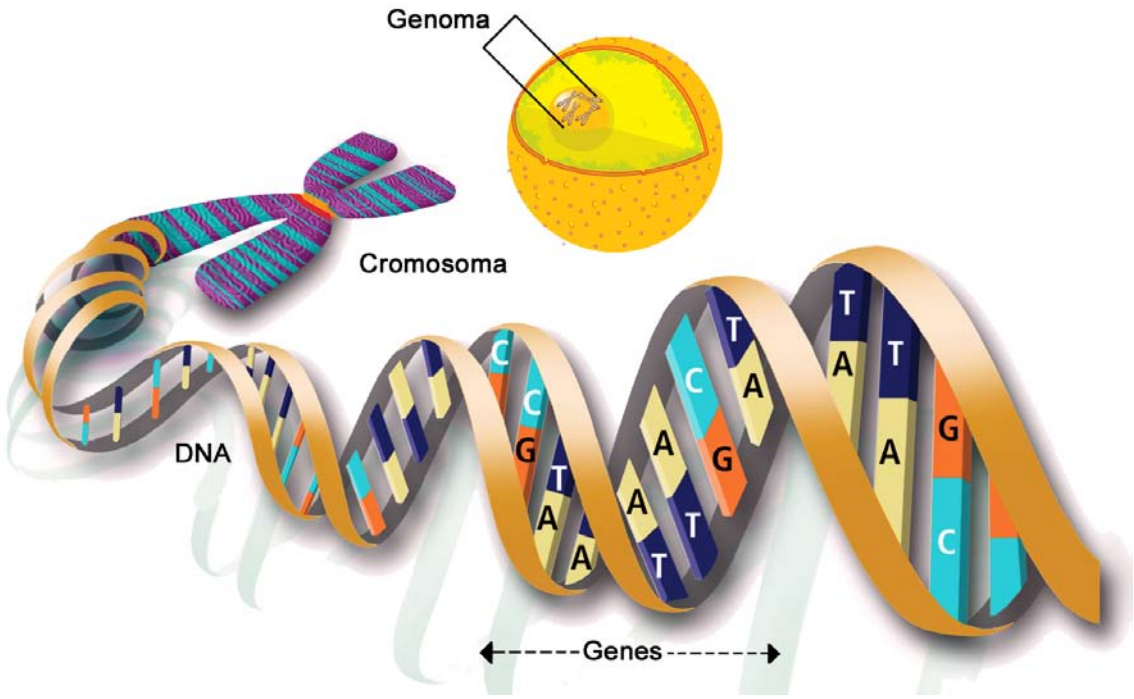

Ilustración 3 De la célula al gen

3.1.1. ADN y cromosomas

El material hereditario no se encuentra disperso por todo el núcleo celular sino que está agrupado en una macromolécula que recibe el nombre de ácido desoxirribonucleico (ADN). En el ser humano, todas sus células, contienen en el núcleo este acido en forma de 23 pares de cromosomas que constituyen su cariotipo. Los cromosomas se agrupan por pares debido a que el ser humano es un organismo diploide y cada uno de los cromosomas homólogos son provenientes de uno de los progenitores.

El ADN es el encargado de controlar todos los procesos celulares, tales como la alimentación, la reproducción celular o la transmisión de caracteres de padres a hijos.

En el núcleo de cada célula humana, hay 46 cromosomas agrupados en 23 pares de homólogos. En cada cromosoma hay un número determinado de genes. Cada gen contiene información para sintetizar una o más proteínas.

Cada uno de los cromosomas, están formados por cromatina, compuesto que incluye al ADN y proteínas histonas (Ilustración 4). En su estado natural el cromosoma está dispuesto como una fina hebra dentro del núcleo celular. Durante la mitosis, los cromosomas pliegan dicha hebra formando la estructura necesaria para la replicación. En esta nueva configuración se reconoce una zona más compacta que recibe el nombre de centrómero. Mediante este centrómero, y estableciendo la disposición del mismo a lo largo de la estructura, es como se puede identificar cada pareja de cromosomas. A las terminaciones finales de los cromosomas se les denomina telómeros.

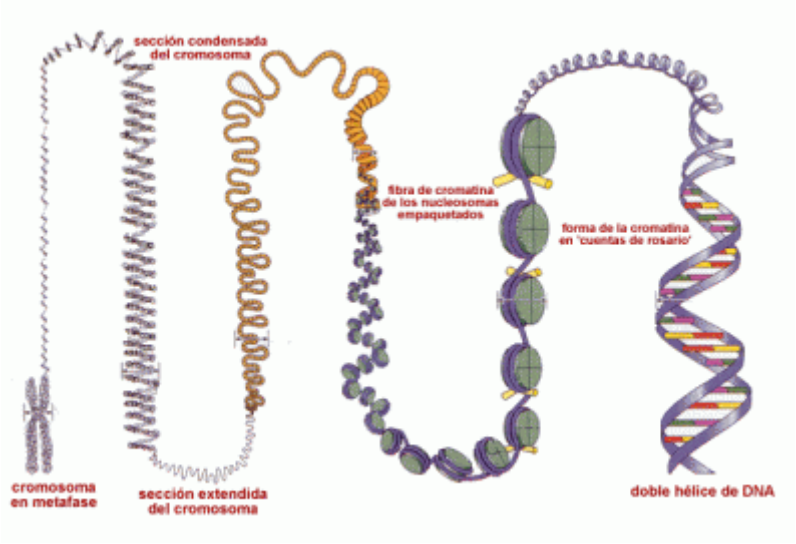

Ilustración 4 Estructura cromosoma

Cada cromosoma es una única molécula de ADN que, a su vez, está formado por millares de nucleótidos. El ADN altamente plegado es lo que forma la estructura del cromosoma.

La molécula de ADN está formada por dos largos filamentos, cada uno de ellos es una secuencia de nucleótidos, que se enrollan entre sí para dar lugar a una doble hélice 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. Las cuatro bases nitrogenadas son: la adenina (A), la guanina (G), la citosina (C) y la timina (T). Para conseguir tener esta estructura de doble hélice, la adenina se empareja siempre con la timina (A-T, T-A) y la citosina se empareja siempre con la guanina (C-G, G-C). Ya que el esqueleto azúcar-fosfato

es siempre igual, el tipo de nucleótidos y el orden en que se disponen, denominado secuencia, es la manera de escribir la información genética con un alfabeto de 4 letras. Este hecho permite representar el ADN de un cromosoma como una doble secuencia de caracteres formada por estas cuatro letras A, G, C, T. Así, cada cromosoma es un par de secuencias complementarias de nucleótidos: la cadena con sentido o codificadora que se lee de izquierda a derecha en dirección de 5' a 3'; y la cadena antisentido o anticodificadora que dirige la síntesis del ARN y se lee de derecha a izquierda, en dirección de 3' a 5'. Como las secuencias son complementarias, si se tiene una de las dos secuencias se puede obtener la otra.

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.

3.1.2. El gen y su estructura

Un gen es una secuencia lineal de nucleótidos de ADN, siendo ésta un fragmento de la secuencia total, que es fundamental para una función específica, pudiendo ser por ejemplo el desarrollo o el mantenimiento de una función fisiológica normal. La secuencia en que se hallan los nucleótidos es lo que diferencia unos genes de otros.

En un cromosoma se encuentran alineados muchos genes. Cada uno ocupa una posición determinada en el cromosoma denominada locus. En general, se llama gen al fragmento de ADN que lleva la información necesaria para la síntesis de una proteína. Es importante resaltar que, si bien el ADN es donde se almacena la información genética de un organismo, las proteínas son las que ejecutan dicha información ya que son las moléculas fundamentales para todos los aspectos estructurales y de actividad celular.

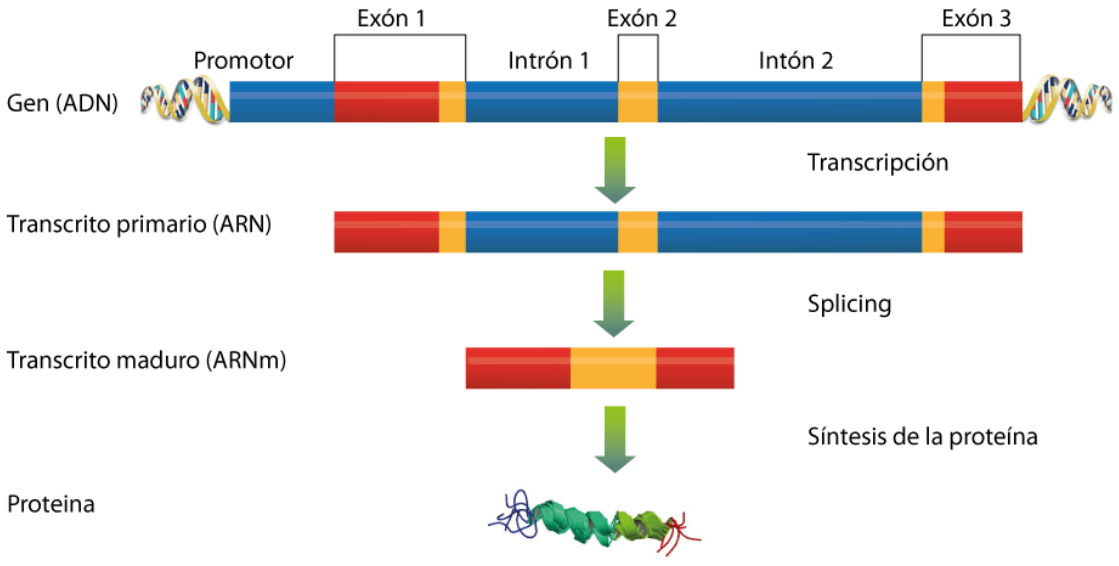

Ilustración 5 Estructura de un gen y fases para la síntesis de la proteína

En general, los genes son las secuencias que serán transcritas en proteínas (Ilustración 5). Los genes poseen diferentes números de nucleótidos en sus secuencias y, por lo tanto, cada gen debe ser localizable dentro del cromosoma al que pertenece para poder utilizar la información

genética que posee. El inicio de un gen viene marcado por su promotor, que es aquella secuencia de ADN que indica a las enzimas de transcripción dónde se debe empezar a traducir dicho gen. Del mismo modo, existe una secuencia de ADN que indica donde termina la secuencia transcribible, el terminador. Una unidad de transcripción es por tanto el conjunto formado por un gen, su promotor y su terminador.

La información genética constituye un mosaico en los que la información útil contenida en los exones es interrumpida por secuencias no codificantes, aparentemente inútiles, llamadas intrones.

3.1.3. El ARN

El Ácido Ribonucleico o ARN es la molécula que dirige las etapas intermedias de la síntesis de las proteínas. Esta molécula se encarga de llevar la información que determina la estructura de las proteínas. El ARN está formado por una cadena de nucleótidos, como ocurre con el ADN, sin embargo la molécula de azúcar del ARN contiene un átomo de oxígeno y la timina (T) que aparecía en el ADN desaparece, dejando paso a la base uracilo (U).

Existen tres tipos de ARN: (1) el ARN ribosómico (ARNr) se encuentra en los ribosomas celulares (estructuras especializadas situadas en los puntos de síntesis de proteínas), (2) el ARN de transferencia (ARNt) que es el encargado de llevar los aminoácidos a los ribosomas para incorporarlos a las proteínas, y (3) el ARN mensajero (ARNm) que lleva una copia del código genético obtenida a partir de la secuencia de bases del ADN celular, especificando la secuencia de aminoácidos de las proteínas. Los tres tipos de ARN se forman a medida que son necesarios, utilizando como plantilla secciones determinadas del ADN celular.

3.2. Transcripción del ADN

La transcripción del ADN es el proceso mediante el cual se transfiere la información contenida en la secuencia del ADN hacia la secuencia de proteína utilizando diversos ARN como intermediarios. Durante la transcripción, las secuencias de ADN son copiadas a ARN mediante una enzima llamada ARN-polimerasa que sintetiza un ARN mensajero manteniendo la información de la secuencia del ADN. Este proceso también se llama síntesis del ARNm.

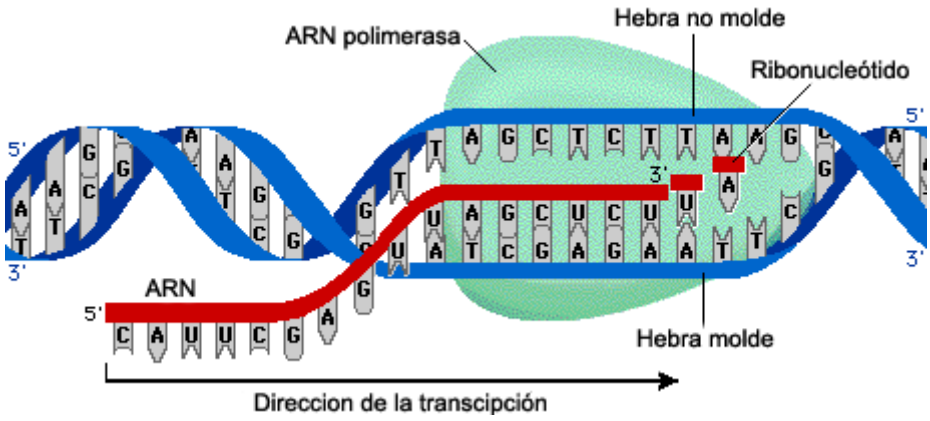

Ilustración 6 Proceso de transcripción (imagen modificada de Pearson Education)

En el proceso de la transcripción intervienen (1) una secuencia de ADN, que actúa como molde, (2) los ribonucleótidos trifosfato, que son las moléculas que forman el ARN (A,G,C y U) y (3) la ARN-polimerasa, tal y como se ve en la Ilustración 6.

La transcripción del ADN consta de varias fases:

- **Iniciación**. La ARN-polimerasa se une a una zona determinada delante del gen que se quiere transcribir, lo que denominábamos como promotor. A continuación se separan las dos hebras del ADN, iniciándose el proceso de copia del ADN a ARNm.
- **Elongación**. Se añaden los ribonucleótidos complementarios al ADN en sentido de 3' a 5'. Se selecciona el ribonucleótido complementario a la base de la hebras molde.
- **Terminación**. La ARN-polimerasa llega a la región terminadora que indica el final de la transcripción. Por tanto la ARN-polimerasa se separa del ARN transcrito y se cierra la doble hélice. Una vez finaliza la transcripción, se añade al ARN resultante una cola de unos 200 nucleótidos de adenina, la cola de poli-A, con lo que queda formado el ARN precursor del ARNm o transcrito primario.
- **Maduración del ARN**. En esta fase se eliminan los intrones, y se unen entre si los exones a través de unas enzimas llamadas ARN-ligasas. Este proceso de eliminación de intrones y unión de exones se denomina s*plicing*. A veces un mismo transcrito primario o pre-ARNm puede madurar de diversas maneras, permitiendo que con un solo gen se obtengan varias proteínas diferentes; a este fenómeno se le llama *splicing alternativo*.

El resultado final de la ejecución de estas fases es la construcción de la molécula de ARNm. Esta molécula es capaz de traspasar la membrana nuclear y viajar por el citoplasma hasta llegar al ribosoma celular en donde participará en el proceso de traducción o síntesis de proteínas.

3.3. Síntesis de proteínas, traducción del ARN.

Las proteínas están formadas por una secuencia de aminoácidos. Los aminoácidos son macromoléculas constituidas por un grupo amino y un grupo carboxilo. La secuencia de aminoácidos que constituyen una proteína es importante ya que determinan la estructura tridimensional de ésta y se determina, por tanto, su función.

Para que la proteína cumpla su función ha de ser sintetizada. Para ello, una vez el ARNm ha madurado y alcanzado a uno de los ribosomas, está preparado para comenzar el proceso de traducción del ARN o síntesis proteica. Este proceso toma como entrada el ARNm y genera como resultado una proteína funcional.

Para llevar a cabo el proceso de traducción es necesaria la existencia de una correspondencia entre los nucleótidos que constituyen el ARNm y los aminoácidos que forman una proteína. Dicha correspondencia se conoce como código genético y viene el ARNt es el encargado de llevarla a cabo. El código genético establece la correspondencia entre un aminoácido y la tripleta de nucleótidos adyacentes que lo formarán. Esta tripleta de nucleótidos recibe el nombre de codón. Como consecuencia de esta correspondencia y dado que cada nucleótido tiene cuatro posibles valores, podrían llegar a existir 64 (4^3) posibles aminoácidos, sin embargo

en las proteínas tan sólo se encuentran 20 aminoácidos diferentes (Ilustración 7). Es por esto que se dice que el código genético es degenerado, ya que un mismo aminoácido puede ser codificado por diferentes codones. Además, a esto hay que añadir que tres de las combinaciones no representan un aminoácido ya que son la marca o codón de parada, que indica donde debe parar la traducción. El codón que representa la Metionina a su vez representa el codón de inicio o punto en que debe de comenzar la traducción de la secuencia.

1ª Letra	U	C	A	G	3ª Letra
U	UUU Phe UUC UUA Leu UUG	UCU UCC UCA Ser UCG	UAU Try UAC UAA Stop UAG Stop	UGU Cys UGC UGA Stop UGG Trp	U C A G
C	CUU CUC CUA Leu CUG	CCU CCC CCA Pro CCG	CAU His CAC CAA Gln CAG	CGU CGC CGA Arg CGG	U C A G
A	AUU AUC AUA Ile AUG Met	ACU ACC ACC Thr ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG	U C A G
G	GUU GUC GUA Val GUG	GCI GCC GCA Ala GCG	GAU Asp GAC GAA Glu GAG	GGU GGC GGA Gly GGG	U C A G

2ª Letra

Ilustración 7 Código genético, correspondencia entre codones y aminoacidos

3.4. Mutaciones y variaciones

Las células poseen una maquinaria muy sofisticada y precisa que permiten realizar copias perfectas de una molécula de ADN, existen incluso diversos sistemas que desechan aquellas copias que hayan sido finalizadas correctamente. No obstante, en ocasiones ocurren ciertos fallos que son desapercibidos por dichos mecanismos de reparación y estos cambios no son eliminados pudiendo llegar a cambiar la información que se transmite a la molécula de ARN.

El término mutación apareció en el año 1901 cuando Hugo Marie de Vries lo definió como un cambio que no se ha producido en generaciones precedentes en la información biológica. Sin embargo, actualmente con el conocimiento del que se dispone éste término ha sido redefinido ya que no todos los cambios producidos en la información biológica afectan igual al individuo que los presenta. De hecho, los individuos de una especie no poseen exactamente el mismo genoma y por tanto su información biológica es diferente. Es por ello que se ha definido el término variación como todo aquel cambio en una secuencia de ADN de un individuo con respecto a su especie. Estas variaciones son las que proporcionan la diversidad existente entre las diferentes especies, por ejemplo la existencia de diferentes tonos de piel en la especie humana.

El término mutación se define entonces como alteraciones heredables que cambian la función genética. Una variación no tiene por qué ser una mutación, sin embargo toda mutación es una variación.

Las mutaciones pueden clasificarse, según la alteración que provocan, de tres maneras diferentes, (1) estructurales, si la mutación se ha producido a nivel del cromosoma, (2) génicas, cuando la mutación se ha producido a nivel del gen, y (3) numéricas, cuando alteran el número de cromosomas.

Las mutaciones no pueden localizarse en cualquier parte del genoma, sino que están localizadas en las secuencias de los genes, ya que son éstos los que intervienen en el proceso de síntesis de proteínas.

Dentro de las mutaciones genéticas, localizadas en el ADN, si se tiene en cuenta el cambio que se ha producido éstas se pueden clasificar como:

- Sustituciones. En ocasiones también son llamadas mutaciones puntuales Normalmente, la variación se produce en un único nucleótido, reemplazándolo por otro.

- Inserciones. Son aquellas en las que una o varias bases adicionales se introducen en la secuencia de ADN.

Deleciones o borrados. Ocurren cuando una base o varias bases de la secuencia se eliminan.

- Inversiones. Cuando una sección del cromosoma se encuentra en la secuencia de forma invertida.

- Translocaciones. Se originan cuando un segmento de cromosoma se intercambia o se traspasa a otro cromosoma.

Indels. Este tipo de cambio se produce cuando un segmento de la secuencia se elimina y en su lugar se inserta una nueva cadena de nucleótidos.

Se debe indicar que en el ámbito de la biología se llama deleción a la eliminación de una o más bases de una secuencia, aunque en términos de castellano puro se debería hacer referencia a este cambio como borrado.

Normalmente cuando la variación ocurre dentro de un gen, tanto en las inserciones como en los borrados, se produce un cambio en la pauta de lectura. Durante la traducción se da la lectura del ARNm en grupos de tres nucleótidos (codón), por eso cuando una o dos bases se insertan o se borran se cambia el marco de lectura, lo que se conoce en biología como frameshift. Este cambio en la pauta de lectura tiene como consecuencia una interpretación totalmente distinta de lo que codifica el ARNm y como consecuencia se origina una secuencia de aminoácidos totalmente diferentes a los traducidos inicialmente.

3.5. Relación genotipo – fenotipo

Se define genotipo al conjunto de genes que presenta un individuo. Estos genes determinan en gran parte las características que tiene un ser vivo. Sin embargo el hecho de que un individuo tenga un determinado gen no implica que éste vaya a expresarse, esto dependerá de la característica recesiva o dominante que el gen en cuestión tenga.

Al conjunto de caracteres morfológicos, funcionales, bioquímicos, conductuales, etc., presentados en individuo que se comenta en el párrafo anterior es a lo que se le denomina como fenotipo. Gran parte del fenotipo es hereditario, correspondiendo a las características que un ser vivo recibe de sus progenitores. Sin embargo, no todo el fenotipo es hereditario, ya que depende en gran parte del ambiente donde el individuo vive. Por ejemplo, el que una persona tenga una cicatriz en la cara es obviamente una característica fenotípica, pero ésta no es hereditaria.

Por tanto queda latente la relación existente entre el genotipo que tiene un ser vivo y la forma en que éste se expresa, su fenotipo.

Estado del arte

Uno de los objetivos más importantes de la bioinformática es el de la explotación de datos provenientes de diversas fuentes dispares que contienen información biológica heterogénea. La integración de datos es el desencadenante que permitirá llevar a cabo el análisis de datos bioinformáticos y descubrir las relaciones científicas que pueda haber entre ellos.

Todo esto ha llevado a la creación de diferentes herramientas, tanto en entornos académicos como industriales, que tratan de cubrir en mayor o menor medida estos objetivos. A continuación se describen algunas de ellas. Éstas han sido elegidas debido a su proximidad con el área referida al diagnóstico génico así como, algunas de ellas, por sus aportaciones en cuanto a las relaciones fenotípicas y la bibliografía que aportan.

4.1. Atlas

El sistema Atlas [20] es un almacén de datos biológicos que integra y guarda localmente secuencias biológicas, interacciones moleculares, información de homólogos, anotaciones funcionales de genes, y ontologías biológicas. Atlas tiene como meta proporcionar tanto acceso a la información como una infraestructura software para la investigación en el campo de la bioinformática.

El sistema Atlas se basa en el modelo relacional. La información almacenada acorde a un conjunto de modelos relacionales, se gestiona a través de consultas SQL que son implementadas mediante varias APIs (*Application Programming Interface*). Los métodos de dichas librerías API son utilizados para construir un conjunto de aplicaciones, que analizan y cargan las fuentes de datos de origen en la base de datos de Atlas.

Atlas almacena e integra localmente instancias de diversas fuentes de datos heterogéneas: GenBank, RefSeq, UniProt, Human Protein Reference Database (HPRD), Biomolecular Interaction Network Database (BIND), Database of Interacting Proteins (DIP), Molecular INteractions Database (MINT), IntAct, NCBI Taxonomy, Gene Ontology (GO), Online Mendelian Inheritance in Man (OMIM), LocusLink, Entrez Gene y HomoloGene. La integración de datos se realiza a dos niveles. En el primero, Atlas almacena los datos de tipos similares utilizando para ello modelos de datos comunes y reforzando las relaciones entre tipos de datos. En el segundo nivel, la integración se realiza mediante la combinación de APIs, ontologías y herramientas.

Con todo esto se han desarrollado herramientas que aprovechan los métodos de las APIs, para facilitar las demandas de los usuarios finales que van desde consultas sencillas de tipos de datos hasta consultas complejas con la intención de relacionar interacciones moleculares entre especies. Así pues, Atlas está visto para ser utilizado tanto por biólogos como por desarrolladores software.

El sistema Atlas está compuesto por cinco partes importantes: (1) fuentes de datos, (2) sistemas ontológicos, (3) modelos de datos relacionales, (4) APIs y, (5) aplicaciones (Ilustración 8).

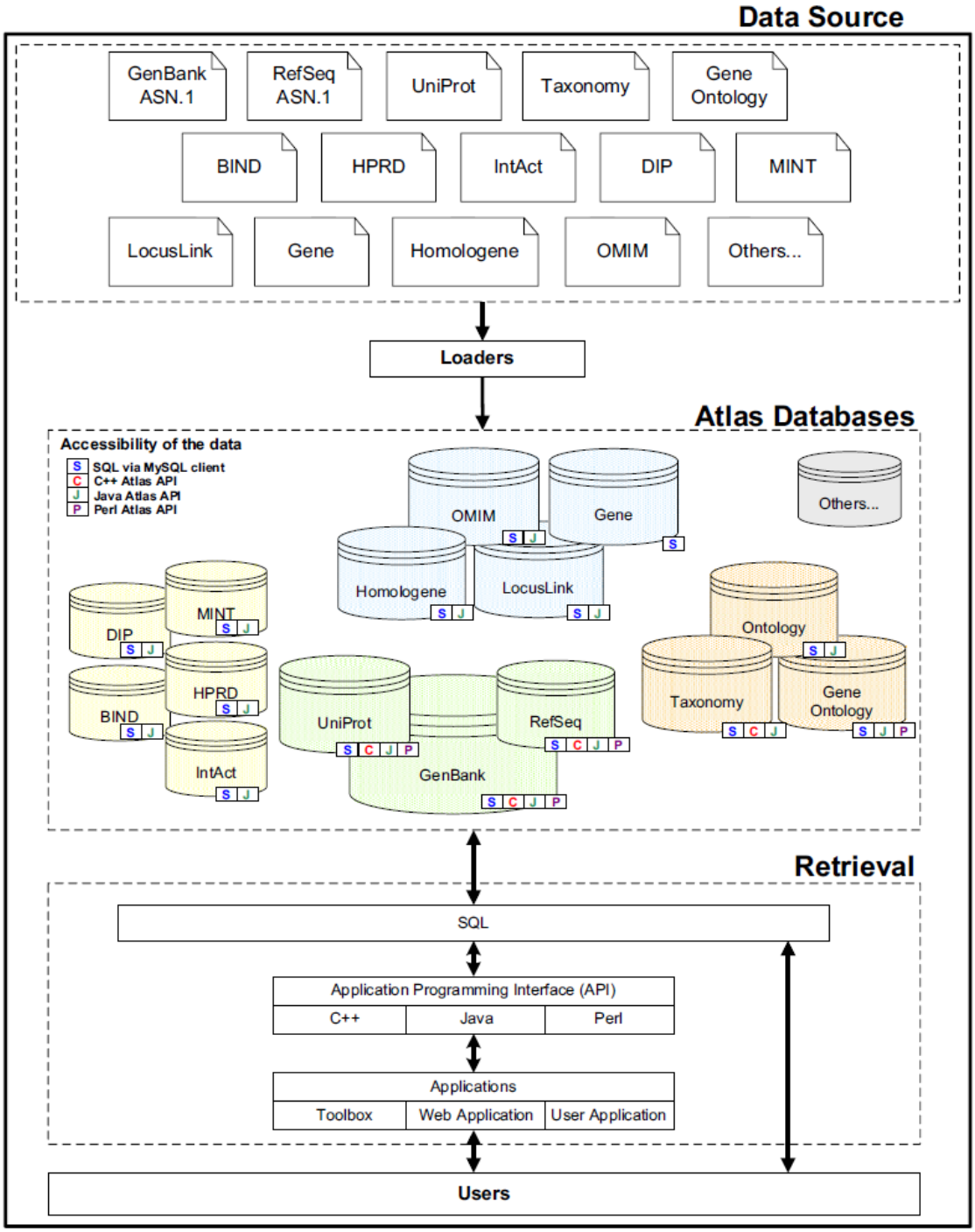

Ilustración 8 Arquitectura Atlas

- Los datos que se integran en el sistema, primero se descargan como archivos de datos de los repositorios fuente (Data Source).
- Los archivos son analizados y cargados en la base de datos relaciona MySQL de Atlas (Loaders).
- Las bases de datos de Atlas están agrupadas por temas biológicos (Atlas Data Bases).
- La capa de recuperación (Retrieval) muestra las diferentes interfaces de las bases de datos. Los datos pueden ser explotados tanto utilizando un cliente MySQL y consultas

SQL, como a través de los APIs y las aplicaciones implementadas en el Toolbox de Atlas.

En lo que respecta a las APIs existen dos tipos: (1) la de carga, utilizadas tanto para la creación de herramientas de carga de bases de datos como para poblarlas y, (2) la de recuperación, cuya función es la de recuperar la información de la base de datos y será utilizada en las aplicaciones de explotación de datos.

En lo referente a este proyecto, es la API de recuperación de datos la que nos interesa ya que es ésta la que se utiliza en las aplicaciones Toolbox de Atlas. Esta API está programada en tres lenguajes: C++, Perl y Java. Está construida mediante metodologías orientadas a objetos, empleando objetos para representar todo, desde las conexiones a la base de datos de bajo nivel hasta las estructuras de datos más complicadas. El código fuente se ofrece bajo la licencia GPL.

La Toolbox de Atlas es una colección de aplicaciones que utilizan el API implementado en C++ para llevar a cabo la explotación de la información de la base de datos. Las aplicaciones son las herramientas basadas en línea de comandos Unix. Son aplicaciones dirigidas al usuario final y por tanto no requieren de ningún conocimiento de programación para su utilización. Dichas aplicaciones permiten la recuperación de una secuencia a partir de su accesion number o su GI, la recuperación de secuencia de todos los organismos que se encuentren por debajo de un nodo dado en el árbol de la taxonomía de NCBI, la recuperación de características a partir del GI o el accesion number, etc.

Atlas ofrece flexibilidad en la integración y explotación de los datos. Los usuarios pueden acceder a los datos ya sea a través de SQL, las API o aplicaciones a nivel de usuario final. Algunas de las utilidades que ofrece el sistema Atlas son las que se describen en los siguientes puntos.

4.1.1. Single record queries

Lo que son las consultas de registro simple. Son las consultas más simples que se puedan hacer. El usuario puede introducir un número de acceso de GenBank o RefSeq y obtener la secuencia asociada a éste en diversos formatos: Fasta, GenBank o ASN.1. También se pueden obtener características con respecto a una secuencia del mismo modo.

4.1.2. Genome annotation

Ofrece herramientas para generar datos para el análisis del genoma, así como un modelo de datos para el almacenamiento de las características biológicas que se han anotado en las secuencias.

4.1.3. Inference of proteine-proteine interactions

Dadas las nuevas asociaciones de la información extraídas de Atlas, estas han suministran un particular uso para el prototipo de un sistema que infiere las interacciones a través de las

especies. Con Atlas es posible recuperar interacciones que se saben se han producido en una especie para otra especie, mediante la integración de grupos de datos que ocurren bajo una consulta y mediante la utilización de las APIs y herramientas que hacen estas consultas mucho más sencillas.

4.1.4. Disease-gene associations

El sistema Atlas está siendo utilizado en este sentido, por ejemplo, para determinar genes homólogos en distintas especies (ortólogos). Concretamente en los genes de la levadura que están implicados en enfermedades humanas. Lo que se está infiriendo es que los genes humanos para los cuales hay un ortólogo en la levadura representan los genes principales que son candidatos a ser agentes de enfermedades humanas.

4.2. SIFT

Sorting Intolerant From Tolerant (SIFT) [21] es un programa que predice, mediante el uso de secuencias homólogas, si una sustitución de un aminoácido (AAS: Amino Acid Substitution) afecta a la función de la proteína potencialmente provocando una alteración en el fenotipo. SIFT se ha aplicado a las bases de datos de variaciones humanas pudiendo diferenciar, en estudios de mutagénesis y en polimorfismos humanos, entre cambios de aminoácidos muy deletéreos y polimorfismos neutros.

La idea en la que se basa SIFT es que un nsSNP (non-synonymous single polymorphism), que ocurre en un determinado gen, puede causar una sustitución en un aminoácido en la correspondiente proteína que se produce, y por tanto afectará al fenotipo del organismo. Las variaciones no sinónimas constituyen más del 50% de las mutaciones conocidas que están involucradas en las enfermedades hereditarias en humanos.

Según los creadores de SIFT su uso debe ser como una guía a seguir para futuros experimentos, pero no para el uso clínico en el diagnóstico de enfermedades. Sin embargo, la habilidad de SIFT para distinguir entre sustituciones deletéreas y neutras hace que SIFT pueda utilizarse como una herramienta de predicción.

El funcionamiento de SIFT (Ilustración 9) se basa en la toma de una secuencia de estudio y el uso de múltiples alineamientos de información para predecir substituciones deletéreas o toleradas en cada una de las posiciones de la secuencia estudiada. SIFT lleva a cabo un procedimiento por pasos que: (1) busca secuencias similares a la dada, (2) elige las secuencias que más se acercan a ésta y que por tanto pueden compartir un funcionamiento similar, (3) obtiene el alineamiento de estas secuencias elegidas, y (4) calcula probabilidades normalizadas para todas las posibles sustituciones del alineamiento. Las posiciones cuyas probabilidades normalizada son menores que 0.05, son catalogadas como deletéreas o malignas. Las posiciones cuyo valor es mayor al 0.05 son predichas como toleradas. Los alineamientos y la búsqueda de secuencias parecidas se hacen mediante la utilización de la herramienta BLAST [22] de NCBI en su versión para proteínas.

Existen tres versiones diferentes de la herramienta SIFT:

- SIFT Genome Tools: Herramienta de predicciones para una serie de cromosomas con su posición y alelos asociados.
- SIFT Batch Tools: Permite realizar predicciones para múltiples proteínas y sus sustituciones.
- SIFT Single Protein Tools: Ofrece predicciones detalladas para una única proteína y sus sustituciones.

El uso de cada funcionalidad depende del tipo de análisis que se quiera llevar a cabo y del tipo de información de entrada que se desee utilizar.

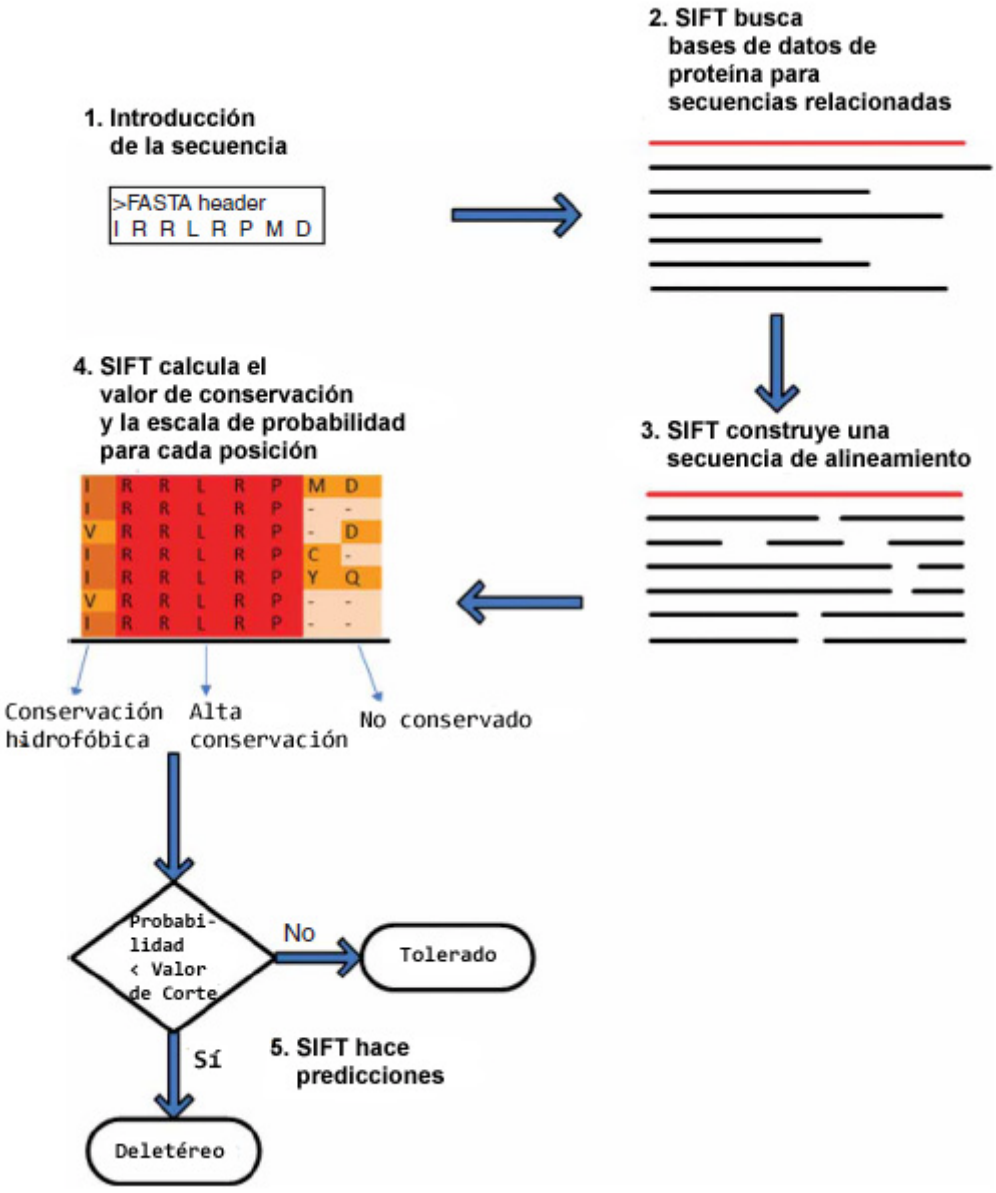

Ilustración 9 Algoritmo SIFT para la puntuación de AASs

En la Ilustración 10 se muestran gráficamente los tres tipos y las herramientas que se encuentran en cada uno de ellos.

A continuación se detallan las diferentes herramientas que se encuentran dentro de las de tipo de predicción de múltiples proteínas.

4.2.1. SIFT Batch Protein

Esta herramienta es utilizada cuando el usuario quiere realizar un análisis de múltiples proteínas con predicciones de nsSNPs (non-synonymous single nucleotide polymorphism) de la base de datos de dbSNP [23] (single nucleotide polymorphism database) o sustituciones elegidas por él mismo.

Como entrada de datos, esta herramienta soporta una lista de identificadores de proteínas (RefSeq o GI) y una lista opcional de sustituciones asociadas a cada una de las proteínas referenciadas. Esta entrada puede llevarse a cabo manualmente o a través de la carga de un fichero.

Existen dos parámetros que se pueden manipular por el usuario. El primero es la selección de las secuencias relacionadas con la proteína que SIFT obtiene a través de la búsqueda realizada por el algoritmo BLAST. Éste tiene dos opciones disponibles (1) *best hits*, que se refiere a las secuencias con mayor similitud y (2) *all hits,* opción que escoge los mejores 100 resultados ordenados por su puntuación. El segundo parámetro es el del porcentaje de similitud, que permite ocultar secuencias altamente parecidas a la de entradas. Su valor por defecto suele ser del 90%.

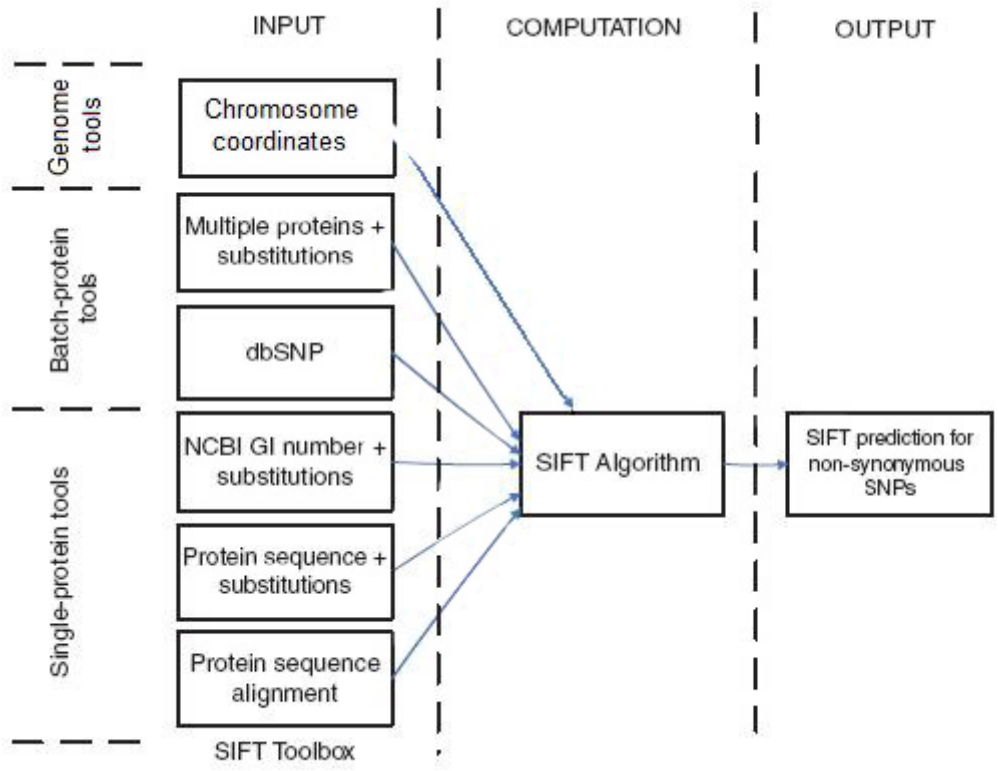

Ilustración 10 Herramientas SIFT

El tiempo estimado de respuesta, con una lista de unas 1000 proteínas, es de 5 a 10 minutos.

4.2.2. SIFT dbSNP

El uso de esta herramienta está destinado al conocimiento de predicciones conocidas de SNP que se encuentran en la base de datos de dbSNP. Esta herramienta tarda entre 2 y 3 minutos en ejecutarse para una entrada de una lista de 1000 rs IDs.

4.2.3. SIFT Blink

Esta herramienta se utiliza a fin de obtener predicciones para todos los posibles AASs de interés y para modificar parámetros como se explicaba en SIFT Batch Protein.

Esta herramienta se basa en el análisis de una sola proteína y acepta como entrada tanto el número GI como el identificador RefSeq, permitiendo al usuario obtener predicciones detalladas para la proteína que se quiere estudiar. SIFT obtiene proteínas relacionadas mediante la utilización de la herramienta de NCBI BLAST Link.

4.2.4. SIFT Sequence

Esta herramienta se encarga del análisis de una única secuencia proteínica que debe ser introducida en formato FASTA. Esta herramienta consume bastante tiempo, más de 20 minutos, debido a que requiere una búsqueda BLAST para compilar un grupo de proteínas similares. Se recomienda utilizar la herramienta anterior si se dispone del número GI o el identificador RefSeq de la proteína estudiada. Para obtener el formato FASTA de la proteína el usuario debe acceder y buscar manualmente dicha información, por ejemplo en las bases de datos de UniProt [24] o de NCBI.

En la aplicación existen tres parámetros que pueden ser configurados por el usuario.

- La base de datos donde buscar. Se ofrecen 3 bases de datos a elegir: UniProt SwissProt, UniProt TrEMBL y NCBI. La segunda es la que el programa usa por defecto, ya que es una gran base de datos de alta calidad. Pese a que utilizar una base de datos de mayor tamaño reduciría el coste de utilizar este programa, se recomienda utilizar bases de datos grandes como NCBI nr para asegurar la suficiente diversidad de grupos de secuencias relacionadas.
- La mediana de conservación de la secuencia. El rango de este valor va de 0, cuando los 20 aminoácidos se han visto en dicha posición como variación, y $\log_2 20$ (4.32) cuando tan sólo un aminoácido se ha encontrado en dicha posición. Se recomienda utilizar el valor por defecto, 3.0, puesto que se han llevado diversos estudios llegando a este valor de optimización.
- Porcentaje de similitud. Este valor ya ha sido comentado en la herramienta SIFT Batch Protein.

4.2.5. SIFT Aligned Sequences

Esta herramienta se debe usar cuando el usuario tenga un alineamiento múltiple de la secuencia que se quiere estudiar y las secuencias homólogas. La calidad del modelo de

sustitución depende de la calidad de las secuencias ortólogas relacionadas que se han obtenido, siendo las secuencias ortólogas aquellas que se encuentran en diferentes especies y que son altamente similares debido a que se han originado en un ancestro común. Por tanto, esta herramienta puede utilizarse por un usuario más avanzado que quiera usar un grupo de secuencias relacionadas que esté refinado manualmente.

Las secuencias alineadas pueden obtenerse mediante la búsqueda en la base de datos de proteínas de NCBI utilizando BLAST o mediante el uso de BLAST Link (BLink) [25], en el caso de que se disponga del identificador de la proteína además de la secuencia, para obtener los resultados BLAST pre-computados de NCBI.

Una vez que las secuencias homólogas se han recuperado ya sea por BLAST o Blink, el usuario puede utilizar herramientas en línea tales como ClustalW2 para alinear las secuencias. La entrada final a SIFT es una lista de secuencias en el formato multi-FASTA donde la secuencia a analizar estará al principio del archivo seguida de las secuencias homólogas alineadas.

4.2.6. SIFT Genome

Esta herramienta, a diferencia de las demás, ofrece predicciones en base a una serie de cromosomas con su posición y alelos asociados. Como parámetros de entrada, separados por comas, se utilizan el cromosoma, las coordenadas, la orientación y los alelos.

4.3. PolyPhen – 2

La herramienta Polymorphism Phenotyping v2 (PolyPhen-2) [26], que es libre y gratuita para fines académicos, predice el impacto que puede causar una sustitución de un aminoácido en la estructura y/o función de una proteína humana mediante consideraciones comparativas.

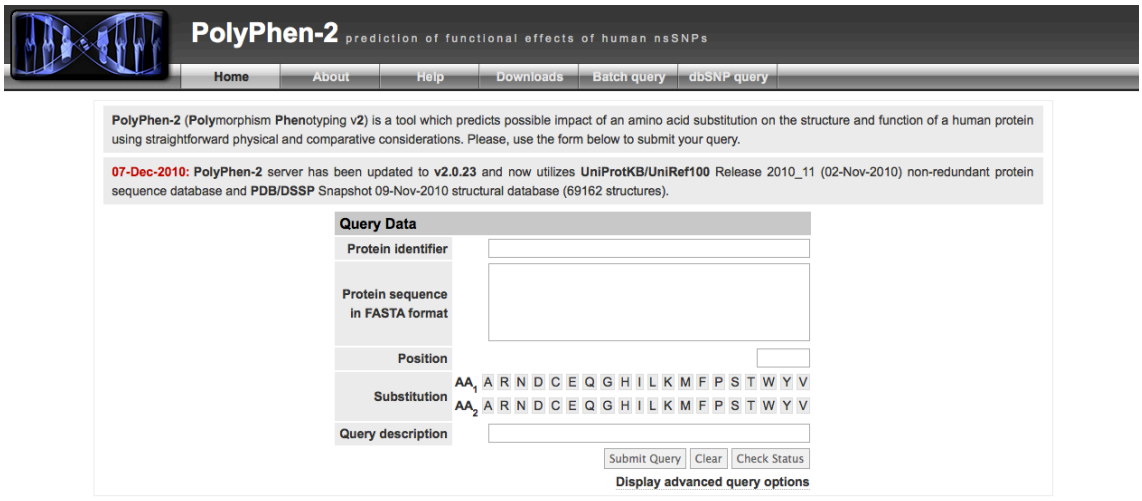

Ilustración 11 Interfaz PolyPhen -2

Dada una sustitución de un aminoácido en una proteína, PolyPhen-2 extrae varias características relacionadas con la secuencia y la estructura de donde se ha producido la sustitución e introduce estos datos en un clasificador probabilístico para obtener el grado de perjuicio que supone. Esta predicción se basa en reglas empíricas que se aplican a la secuencia, a la información filogenética y estructural que caracterizan la substitución.

PolyPhen-2 realiza un completo y automatizado proceso compuesto de varios pasos (Ilustración 12).

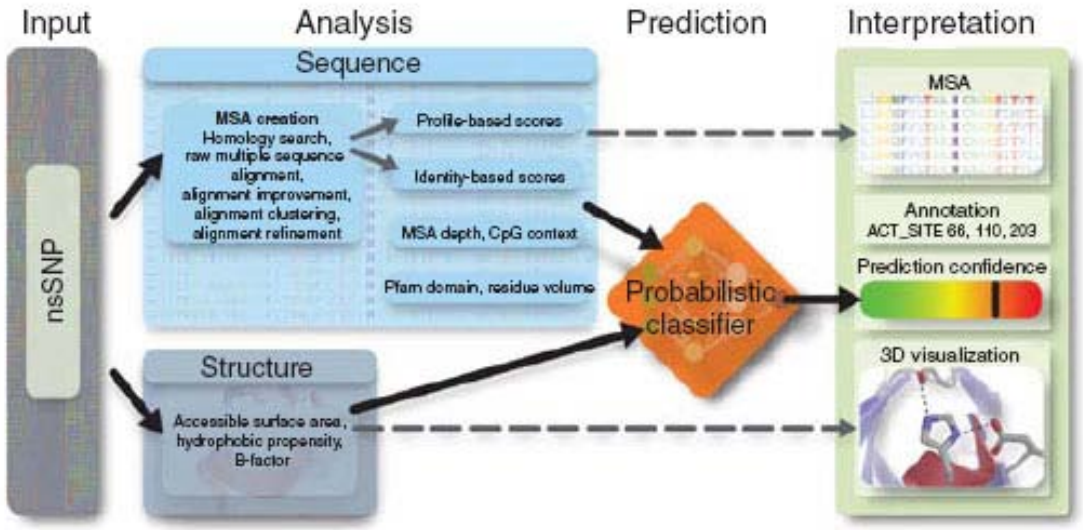

Ilustración 12 Fases proceso PolyPhen-2

En lo referente a la parte del análisis, cabe destacar dos partes: (1) el análisis de una secuencia, donde se identifica en qué sitio o región específica ocurre la sustitución y (2) el análisis de la estructura, donde se obtienen las características más importantes sobre a proteína entre las que se escogerán únicamente algunas de ellas como el B-facto para obtener los resultados finales.

Para llevar a cabo el estudio PolyPhen-2 necesita (Ilustración 11) de la introducción manual del identificador de la proteína que se desea estudiar o la secuencia de aminoácidos que forman la proteína en formato FASTA, la posición dónde se produce el cambio y, el cambio producido (qué aminoácido cambia por qué otro). Opcionalmente se puede introducir una descripción.

PolyPhen-2 predice el significado funcional de la variación mediante un clasificador Naïve Bayes. Para ello hace uso de dos pares de conjuntos de datos con los que se entrena el clasificador. Por una parte usa el conjunto de datos HumDiv que está formado por los alelos dañinos causantes de enfermedades Mendelianas que se encuentran en la base de datos UniProtKB junto con las diferencias entre las proteínas humanas y sus parientes mamíferos más cercanos. Por otra está el conjunto HumVar, formado por las mutaciones causantes de enfermedades humanas que se encuentran en a base de datos UniProtKB y los nsSNPs (nonsynonymous single-nucleotide polymorphisms) sin enfermedades dañinas asociadas. El clasificador Naïve Bayes obtiene la probabilidad de que una mutación sea dañina dando una tasa o estimación de falsos positivos (posibilidad de que una mutación sea clasificada como perjudicial cuando en realidad no lo es) y verdaderos positivos (una mutación es clasificada

como perjudicial y lo es). Las mutaciones son catalogadas como benignas, posiblemente dañinas o probablemente dañinas.

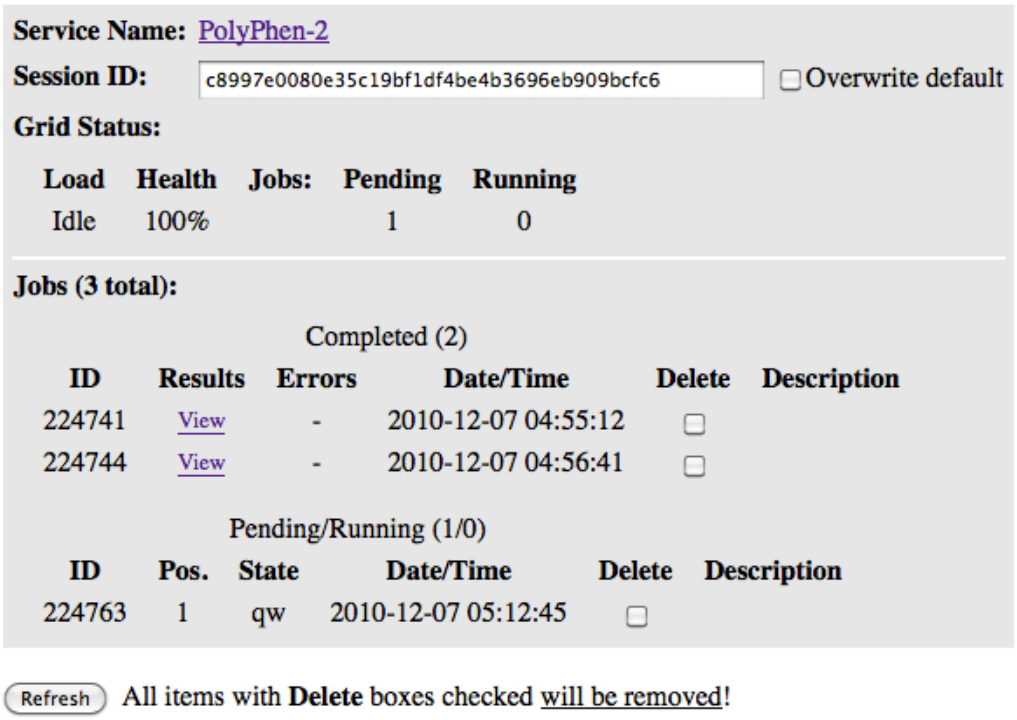

Ilustración 13 Listado de los estudios llevados a cabo

Una vez llevado a cabo el estudio PolyPhen-2 muestra una tabla dónde se puede acceder a los diferentes estudios llevados por el usuario, ordenados temporalmente en un listado (Ilustración 13). Esta interfaz no es demasiado intuitiva en cuanto a que es el usuario quien debe actualizar manualmente la página para ver cómo el estudio pasa del estado pendiente, al estado activo, y de éste al estado completado. Siendo en este último donde se podrá acceder a los resultados o a los errores obtenidos.

Una vez accedemos a la parte de los resultados, PolyPhen-2 muestra un informe donde se indica el grado de probabilidad de ser perjudicial de cada una de las variaciones tratadas para cada conjunto de datos (Ilustración 14).

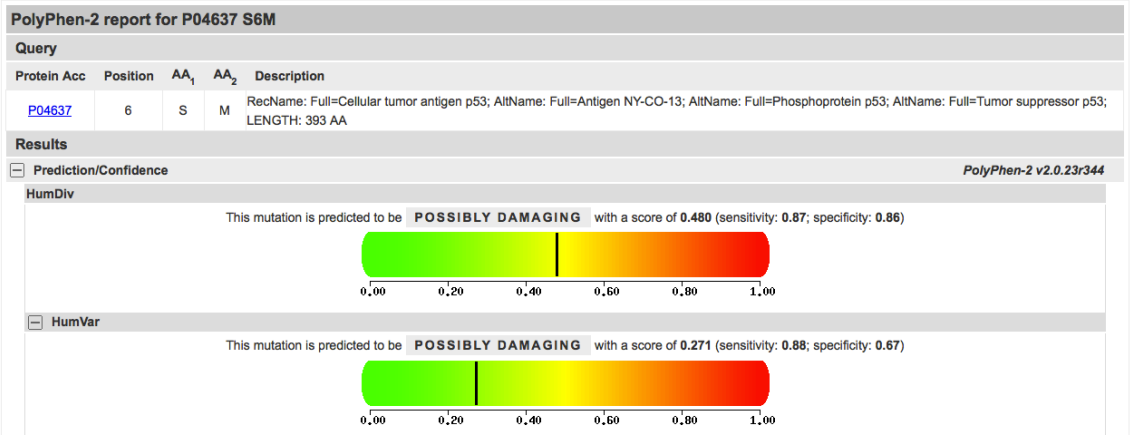

Ilustración 14 Resultado PolyPhen-2

El hecho de que la herramienta proporcione una predicción para cada uno de los conjuntos de datos con los que se entrena, hace difícil elegir qué diagnóstico es el más adecuado. Es por esto que Hum-Var se usa para realizar diagnósticos de enfermedades Mendelianas que requieren distinguir mutaciones con efectos drásticos a partir de otra variación humana. Sin embargo, Hum-Div se utiliza para evaluar alelos raros con una fuerte implicación en fenotipos complejos.

Por último, se debe hacer hincapié en la diferencia que existe entre las mutaciones totalmente penetrantes y los SNPs implicados en complejos fenotipos humanos. Las mutaciones penetrantes son aquellas en las que la frecuencia en la que un fenotipo específico se expresa en los individuos con un determinado genotipo, dadas unas determinadas condiciones ambientales, es alta. Éstas son las que causan enfermedades mendelianas. No obstante, que un SNP esté implicado en un fenotipo complejo no es condición necesaria y suficiente para definir dicho fenotipo, pero su efecto depende de muchos otros componentes genéticos y ambientales. En otras palabras, los SNPs pueden afectar a los factores de riesgo de tener un fenotipo específico en el sentido estadístico. Así pues, el efecto de un SNP particular sobre un fenotipo puede ser visto sólo como una diferencia de frecuencia entre los individuos que lo muestran. Por lo tanto, los resultados dados por PolyPhen-2 han de ser vistos en el sentido probabilístico y no como un diagnóstico.

4.4. Alamut

Alamut es una aplicación de software de apoyo a las decisiones desarrollada por Interactive Biosoftware, para el diagnóstico de mutaciones en la genética molecular en medicina. Es una aplicación cliente-servidor (Ilustración 15) que integra la información genética que proveniente de diferentes fuentes para describir variantes utilizando la nomenclatura HGVS [27] y ayuda a interpretar su condición patógena.

Ilustración 15 Alamut, aplicación cliente-servidor

Alamut obtiene su información genómica de Ensembl [28] y la integra con datos de otras fuentes como UniProt, dbSNP, PubMed, y UCSC [29].

El primer paso para la utilización de Alamut es el de introducir el gen que se desea estudiar (Ilustración 16). Existen diferentes formas de hacerlo: Introducción (1) del nombre del gen, (2) del símbolo del gen, (3) de una enfermedad o (4) de cualquier otra palabra relacionada con el gen. Una vez introducida esta información se ofrece una lista de genes que podrían coincidir con la información introducida, de modo que se facilite la entrada de dicho gen en el caso de, por ejemplo, que no se conozca su nombre o se quiera estudiar el gen por el fenotipo que ocasiona.

Alamut presenta la información del gen en una misma pantalla (Ilustración 17) en la que se puede encontrar la estructura del gen, su genoma, el grado de conservación de nucleótidos que tiene, su transcrito, los SNPs asociados, las variaciones almacenadas en las bases de datos asociadas que dan como resultado proteínas no funcionales, el alineamiento de múltiple proteínas.

En la parte referida a la estructura del gen se muestran los intrones y exones de éste. Siendo representados los intrones en color amarillo y los exones en color azul. Se puede ampliar la región que se desea estudiar para poder tener una visión más precisa de la zona.

Una vez ampliada la región de interés, se observa justo debajo su secuencia de ADN y su cadena complementaria. En la zona del transcrito se muestran en grupos de tres los nucleótidos de la cadena de ADN que forman los aminoácidos. En esta zona podemos seleccionar cualquier nucleótido o grupo de nucleótidos con el botón izquierdo del ratón e introducir manualmente la mutación que se desee.

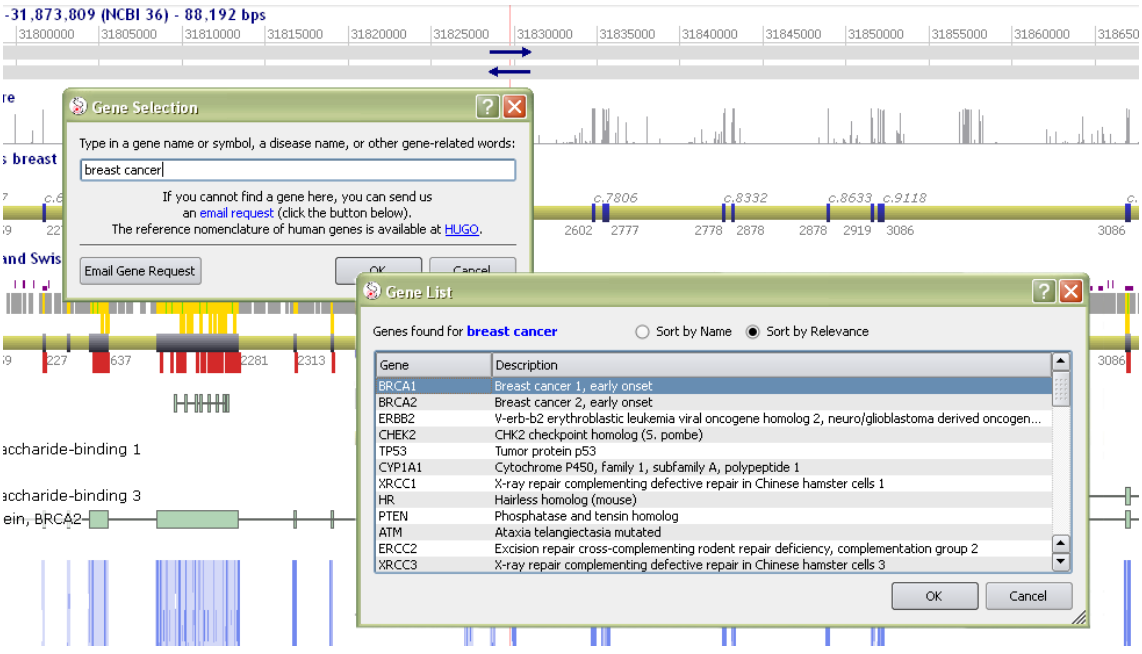

Ilustración 16 Selección del gen

Este proceso de introducción de mutaciones permite crear una base de datos de mutaciones propias, las cuales pueden exportarse en un fichero de texto siguiendo la nomenclatura de HGVS y por tanto importarse a otras investigaciones.

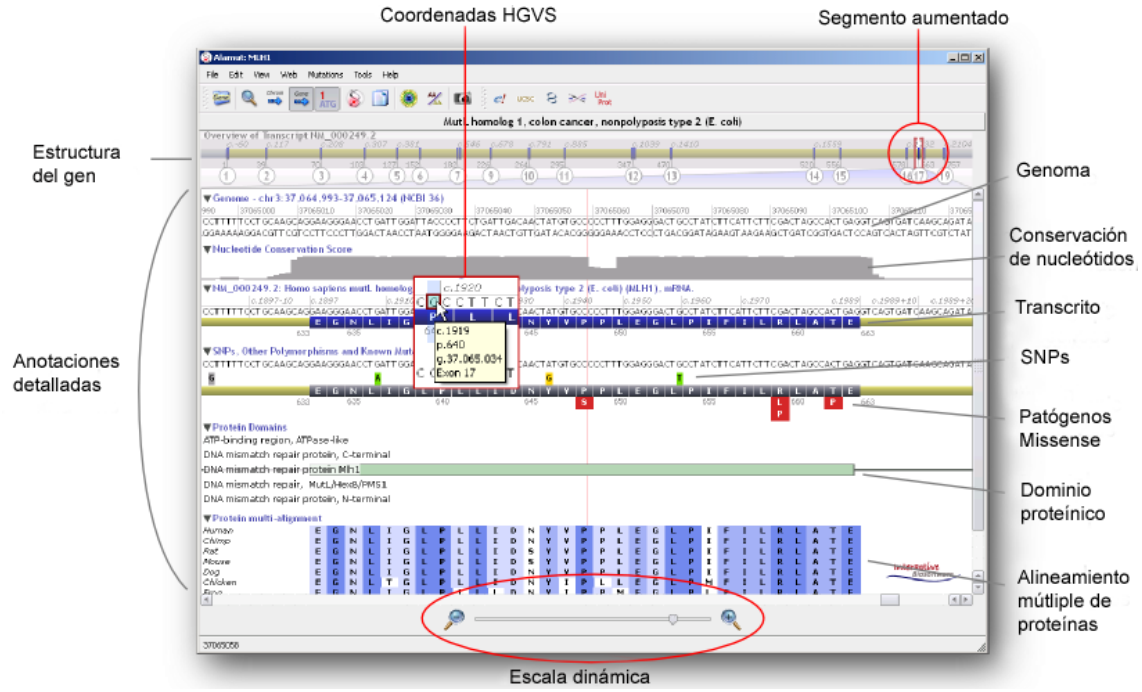

Ilustración 17 Descripción de la interfaz de Alamut

Por otra parte, en el apartado que se encuentra debajo del relacionado con el mRNA se encuentran accesibles los SNPs, otros polimorfismos y las variaciones que SwissProt tiene respecto al gen que se está estudiando. En verde se muestran los SNPs sinónimos, en amarillo los no sinónimos. Se proporciona la opción de ver las consecuencias que tiene dicho SNPs no sinónimo mediante el botón izquierdo del ratón. Para ambos tipos de SNPs es posible obtener más información detallada (Ilustración 18) mediante la utilización del botón derecho del ratón, teniendo así acceso a la información de NCBI mediante el link que se encuentra en el identificador del SNP.

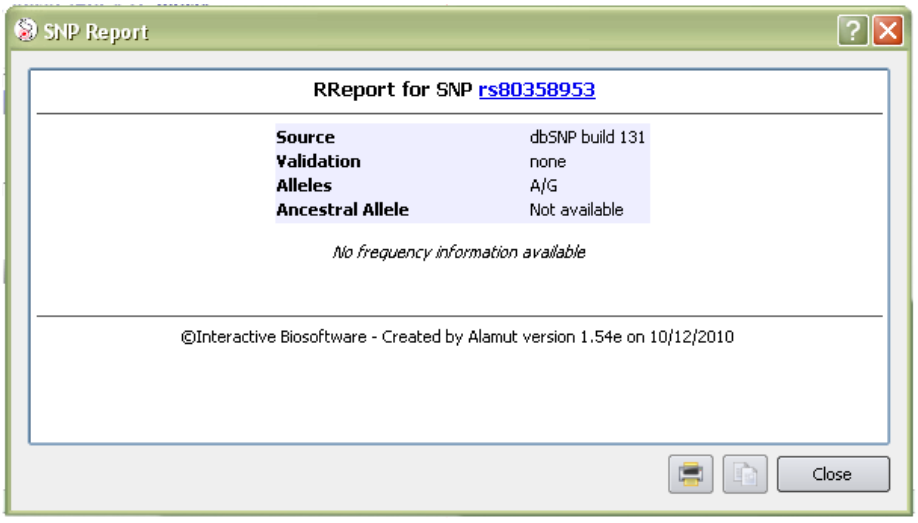

Ilustración 18 Información detallada de SNP

Se puede observar también que existen líneas moradas subrayando ciertos aminoácidos, estas líneas representan borrados o inserciones que se han producido en la cadena. Las letras que están en color rojo representan sustituciones. Los cuadrados rojo que se encuentran debajo

de los aminoácidos indican las variaciones de carácter patógeno que se encuentran en la base de datos de SwissProt, pulsando en éstas con el botón derecho del ratón se puede acceder a dicha base de datos y obtener más información.

Para añadir mutaciones en el programa se ha de seleccionar, mediante el botón derecho del ratón, los nucleótidos de la secuencia de referencia que se verán involucrados en dicho cambio. Una vez seleccionados, podemos añadir la mutación mediante el botón derecho del ratón tal y como se observa en la Ilustración 19. En la ventana que aparece se puede indicar de qué tipo de mutación se trata (substitución, eliminación, inserción, duplicado, indel) y dependiendo del tipo se añadirán el rango o la posición de la mutación y el cambio de nucleótidos que se haga originado. Una vez la mutación se guarda queda registrada en el gen como un cuadro amarillo que se puede encontrar en la parte superior de la sección del mRNA.

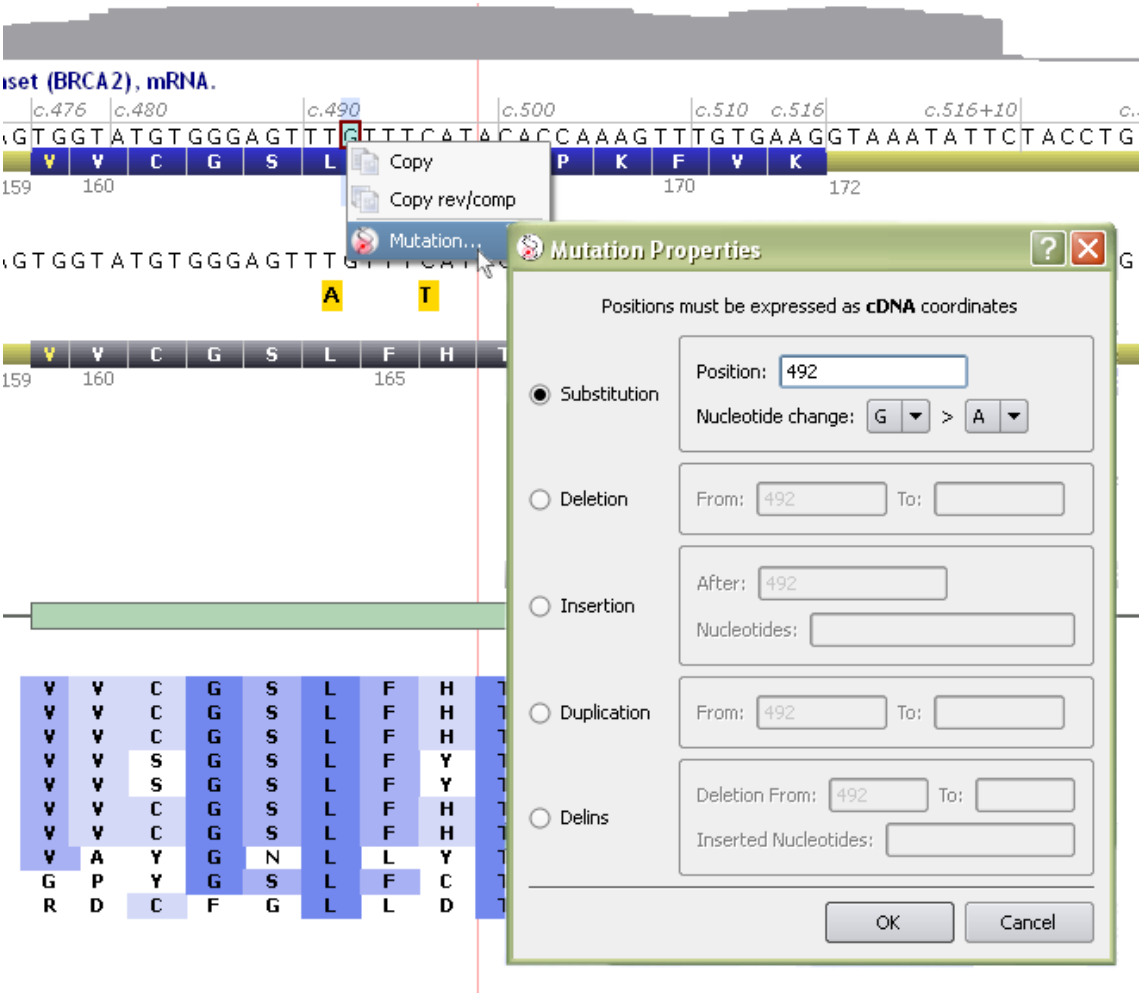

Ilustración 19 Introducción de una mutación

Finalizada la inserción de la mutación, Alamut permite generar un informe de la variación. En este informe el usuario deberá indicar qué consecuencias tiene dicha variación, existiendo tres o cuatro clases de consecuencias dependiendo del tipo de clasificación utilizada. Si se utiliza la clasificación simple podemos diferenciar entre patógeno, no patógenos y desconocidos. Si la clasificación elegida es la CMSG-VKGL entonces se puede diferenciar entre ciertamente no patógena, poco probable de ser patógena, probable de ser patógena, ciertamente patógena. Este último haría referencia a estudios que indican el grado de

probabilidad que tiene una variación de producir un efecto patógeno. Este dato se puede obtener mediante la utilización de herramientas como PolyPhen o SIFT, anteriormente explicadas. Alamut, contiene accesos directos a estas dos herramientas a través de la ventana del informe (Ilustración 20).

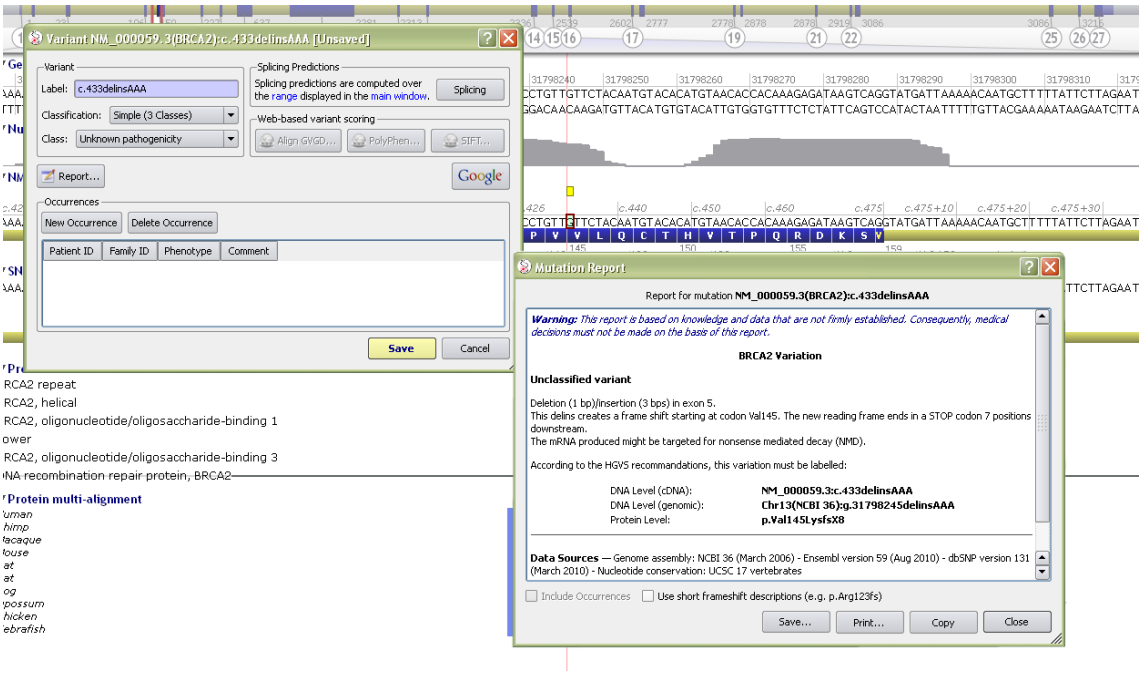

Ilustración 20 Informe Alamut

Esta ventana permite introducir también información sobre los pacientes en los que se ha encontrado dicha variación, pudiendo decir desde el identificador del usuario o el de su familia hasta el fenotipo que presenta. En esta ventana se puede a la búsqueda en Google de la variación. Existe un apartado para obtener la predicción de los splicings (Ilustración 21), dicha predicción se observa en una ventana a parte de la cual se puede extraer un informe en formato HTML.

Haciendo clic en el botón Report, Alamut muestra una ventana con el informe de la variación en la que se cita de qué tipo es la variación, qué efecto fenotípico produce si lo tiene, la variación en nomenclatura HGVS tanto a nivel genético, como a nivel de mRNA o a nivel proteínico, las bases de datos en las que se basa el estudio, etc.

Dicho informe puede guardarse en formato HTML comunicando al usuario que aunque este sea el formato en el que se ha guardado, el informe puede modificarse con alguna herramienta de procesado de texto como Microsoft Word u Open Office. También se puede copiar e imprimir directamente el informe mostrado.

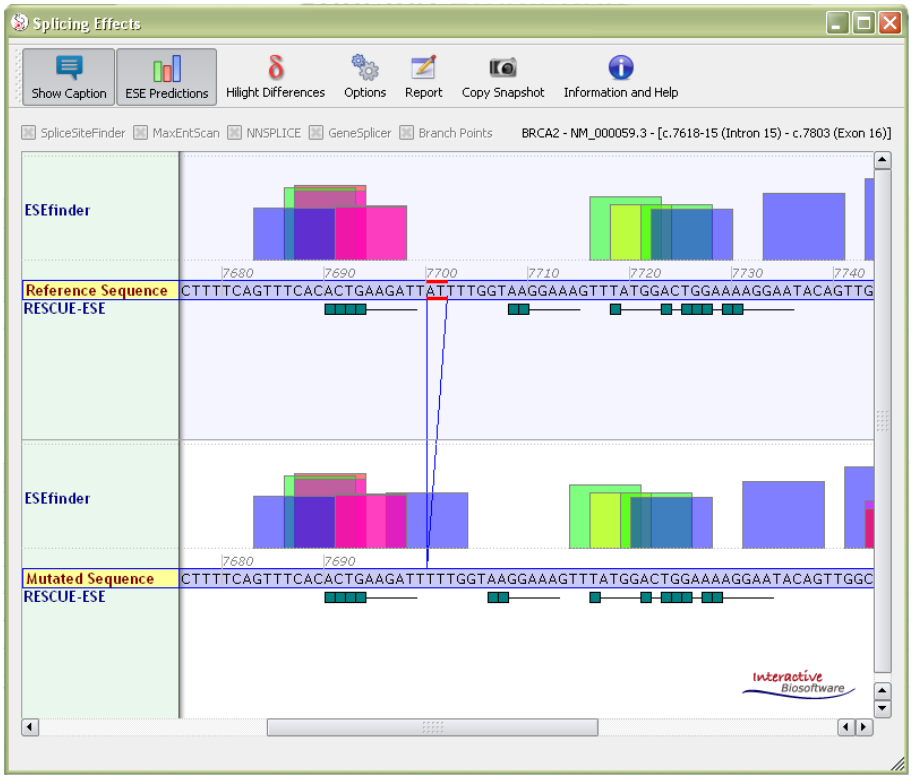

Ilustración 21 Vista Splincing

4.5. HOPE

Have yOur Protein Explained (Hope) [30] se basa en la idea que una parte considerable de las mutaciones causantes de enfermedades humanas se encuentran en la parte codificante de las proteínas del genoma y, por tanto pueden afectar a la estructura y el funcionamiento de estas proteínas provocando un efecto fenotípico.

Recolectar toda la información disponible que está relacionada con las proteínas que se quieren estudiar puede ser un reto y requiere mucho tiempo. La tarea de extraer la información para obtener una conclusión acerca de los efectos de una mutación es una tarea difícil.

Actualmente, existen muchos servicios en la red que ofrecen ayuda a la investigación biomédica en la predicción de los efectos que pueden producir las mutaciones. Estos servicios se nutren de la información que existe en diferentes fuentes, para llegar a las conclusiones sobre la posibilidad que tiene una mutación de producir una enfermedad.

HOPE se nutre de las ventajas de las nuevas herramientas de la era de la e-Science [31]. El desarrollo de servicios web facilita el uso de bases de datos y métodos externos en desarrollo de software de uso interno (in-house). También facilita el mantenimiento del software y el desarrollo de éste a partir del uso de la lógica suministrada por dichos servicios web. El uso de servicios web tiene tanto ventajas como desventajas. Las ventajas son:

- Ahorro de tiempo mediante la reutilización de código.
- Normalmente están actualizados.
- Ejecución remota, lo que evita la sobrecarga de la máquina local.

- Evita la necesidad de mantener el software interno.

Entre las desventajas tendríamos que el código de los servicios Web no es accesible y que no se garantiza que éstos siempre vayan a estar disponibles para su uso.

HOPE es una aplicación web de nueva generación para el análisis automático de mutaciones. El desarrollo de HOPE se lleva a cabo para explicar el origen molecular de una enfermedad que está relacionada con un fenotipo causado por una mutación en proteínas humanas.

HOPE se basa en la idea de e-Science llevándolo un paso más allá, y es que la recopilación de datos que se lleva a cabo en dicha herramienta se hace a través de la utilización de servicios web y Distributed Annotation Systems (DAS) [32]. Los DAS se basan en un protocolo de comunicación usado para el intercambio de anotaciones de secuencias genómicas o proteínicas. La idea es que las anotaciones que se hagan no sean proporcionadas por una única base de datos centralizada sino que ésta se encuentre dispersa en múltiples sitios. Los servidores DAS son sistemas cliente-servidor en los que un solo cliente integra información de múltiples servidores. Esto permite que una sola máquina pueda recopilar información sobre anotaciones de múltiples sitios web, cotejar dicha información, y mostrársela al usuario en una única vista.

La interfaz de la herramienta HOPE permite al usuario introducir la secuencia de proteínas que desea estudiar. En la Ilustración 22 se muestra un ejemplo.

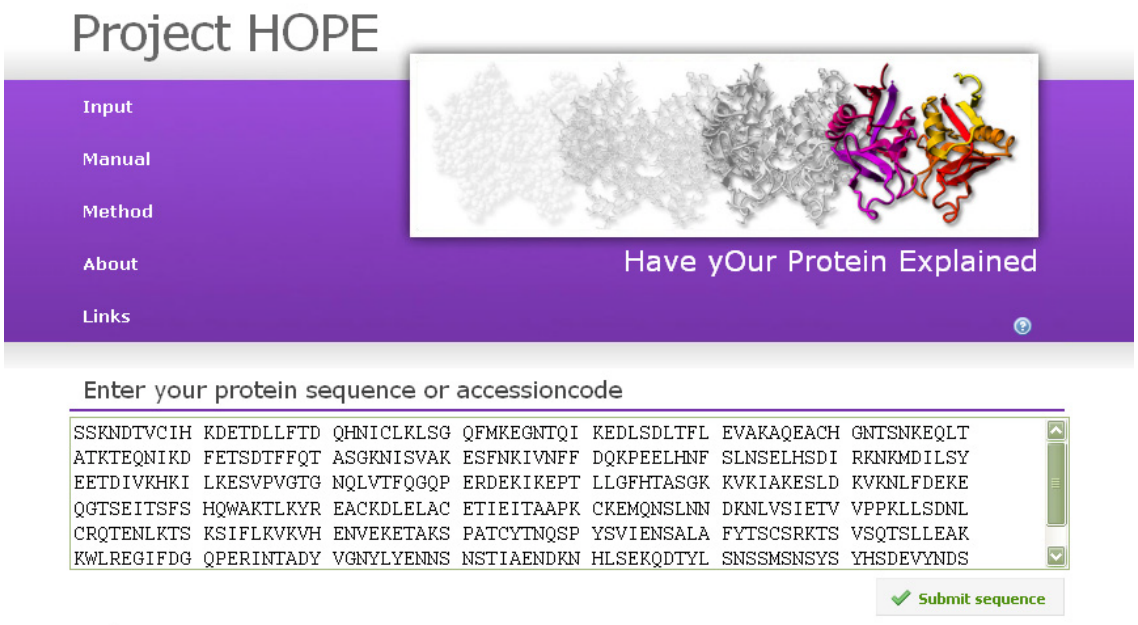

Ilustración 22 Interfaz HOPE - Inserción secuencia proteínica

Una vez se introduce la secuencia y el programa comprueba que dicha secuencia es correcta, el usuario debe escoger la posición en la que se produce la mutación y el cambio, como se muestra en Ilustración 23. En esta imagen se observa que se ha seleccionado (en color morado) la posición 143 donde se señala que existe un cambio de una treonina a una cisteina.

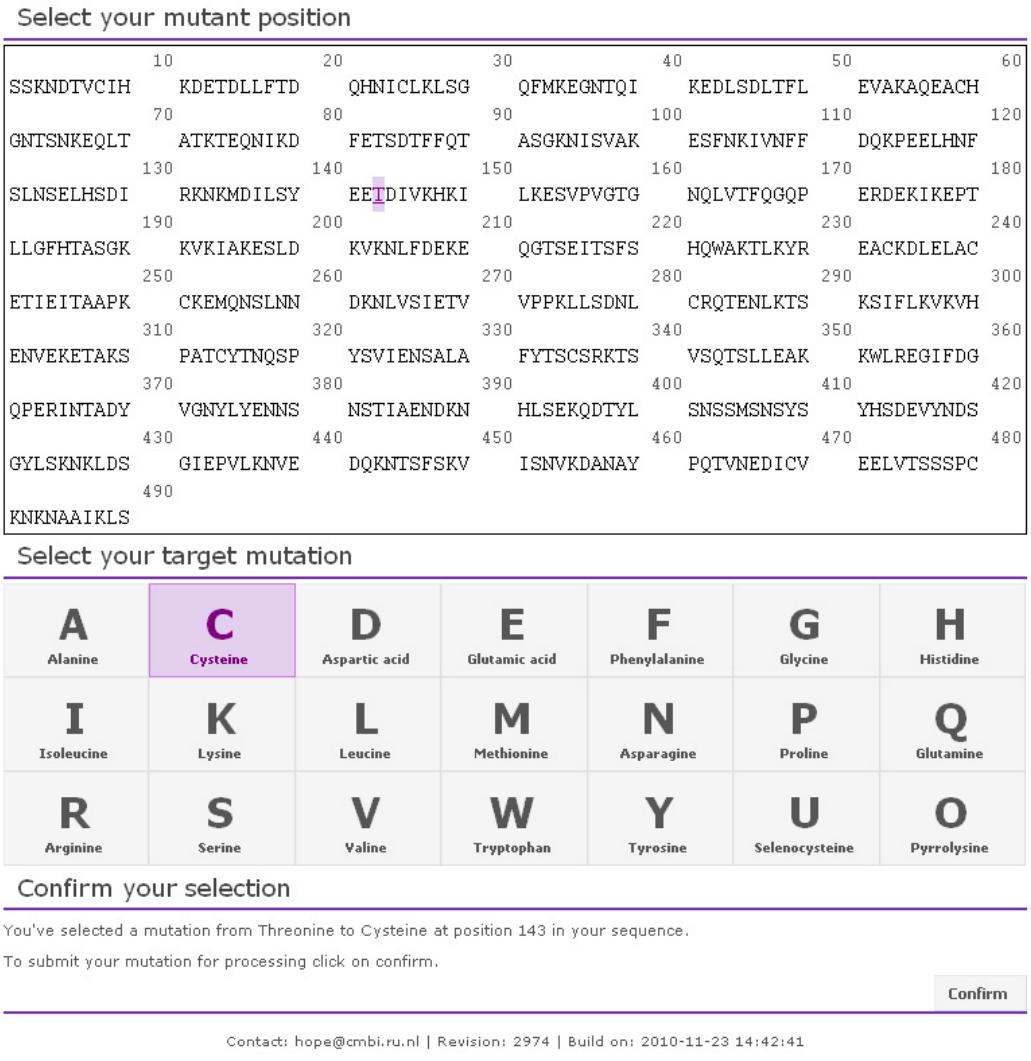

Ilustración 23 Interfaz HOPE - Selección de mutación

Una vez confirmado el cambio por el usuario, el programa comienza con la construcción del informe que mostrará en la misma pantalla. En la Ilustración 24 vemos un ejemplo de dicho informe. En éste, existen cuatro apartados claramente diferenciados:

- **Método**. Se explican los métodos seguidos para recoger la información y se ofrece información sobre la proteína introducida.
- **Sección de aminoácidos**. Se muestran las estructuras esquemáticas de los aminoácidos afectados antes y después de la mutación. Además se evaluará el efecto de la mutación en lo referente al dominio de la estructura y de los residuos mutados. Las características que se tratan solo serán mostradas si se dispone de información acerca de ellas, en cambio siempre se muestra una pequeña conclusión basada solo en las propiedades de los aminoácidos. En el caso de que una estructura tridimensional esté disponible, también se mostrarán ilustraciones y animaciones en el informe.
- **Conservación**. En este apartado se aborda lo perjudicial que puede llegar a ser la mutación.
- **Propiedades del aminoácido**. Aquí se muestran las características que son consecuencia de la mutación.

El informe final se centra en el efecto de las mutaciones en la estructura tridimensional.

Method

There is no structural information known for this protein. No solved 3D-structure or modelling template was found. Therefore, HOPE will use annotated information from the Uniprot-database and predictions from a series of DAS-servers for mutational analysis. More information about your protein of interest can be found in Uniprot-entry P51587 . See the method page for more information.

Amino acids

You are interested in a mutation of a Threonine into a Cysteine on position 143.
The following two figures show the schematic structures of the original (left) and the new mutant (right) amino acid. The backbone, which is the same for each amino acid, is coloured red. The side chain, unique for each amino acid, is coloured black.

Each amino acid has its own specific size, charge and hydrophobicity-value. The original wild-type residue and newly introduced mutant residue often differ in these properties.
The new mutant residue is smaller than the wild-type residue. The new mutant residue is more hydrophobic than the wild-type residue.
The report will evaluate the effect of the mutation on the following features: Contacts made by the mutated residue, structural domains in which the residue is located, modifications on this residue and known variants for this residue. A feature will only be shown when there is information available. A short conclusion based on the amino acid properties only is always shown. In case a 3D-structure is available you will also find images and animations in the report.

Conservation

The wild-type residue is not conserved at this position. Another residue type was observed more often at this position in other homologous sequences. This means that more homologous proteins exist with that other residue type than with the wild-type residue in your protein sequence. The other residue type is not similar to your mutant residue. Therefore, the mutation might be damaging.

Amino acid properties

The wild-type and new mutant amino acids differ in size. The new mutant residue is smaller than the wild-type residue. The mutation will cause an empty space in the core of the protein. The hydrophobicity of the wild-type and new mutant residue differs. The mutation can cause loss of hydrogenbonds in the core of the protein and as a result disturb correct folding.

Ilustración 24 Interfaz HOPE - Informe resultado

El código, aunque libre, todavía no está disponible y tan sólo se puede acceder a una versión de demostración a través de la web http://www.cmbi.ru.nl/hope/input. El esquema de la base de datos se encuentra disponible en http://www.cmbi.ru.nl/hope/schemaspy/index.html.

Internamente HOPE hace uso de BLAST para comparar la secuencia introducida por el usuario contra las bases de datos de UniProt y Protein Data Bank (PDB) [33]. De la búsqueda contra UniProt, se obtienen los identificadores de las proteínas y el código de acceso de éstas. En

cambio de la búsqueda en PDB se obtiene la estructura de la proteína o alguna plantilla para el modelado de homologías.

La estructura de las proteínas se analiza mediante el uso de servicios web WHAT IF [34]. Estos servicios pueden calcular un gran rango de características estructurales. Cuando HOPE no puede utilizar información acerca de la estructura tridimensional, ofrece sus conclusiones en base a la información relacionada con la secuencia y de los resultados que han sido publicados sobre sus variaciones y mutaciones.

En lo referente al almacenamiento de la información, se crea un nuevo sistema de información (SI) para cada proteína estudiada. Las características de estas proteínas son almacenadas en el SI en base al residuo y puede tener uno de los siguientes cuatro tipos de datos:

- **Contactos**. Información referente a la interacción que tiene el residuo con otra entidad.
- **Características variables**. Tipos con valor, como puede ser el ángulo de torsión.
- **Características fijas**. Etiqueta un residuo con una característica sin valor, lo que indica que el residuo está localizado con un cierto dominio como puede ser el que forme parte de una región activa.
- **Variantes**. Hace referencia a las mutaciones o variaciones en la secuencia conocida en esa posición.

La información se almacena físicamente durante un mes, por si el usuario requiere información acerca de otras mutaciones en la misma molécula. Después de un mes, cada SI es eliminado para asegurar que los resultados obtenidos no están basados en información desactualizada. Por tanto, no existe una base de datos permanente. La información de HOPE, de acuerdo con el paradigma del e-Science, está repartida a través de internet y se combina cada vez que se realiza una llamada.

Por lo que respecta al esquema de decisión de HOPE, éste utiliza toda la información recolectada que ha sido combinada con propiedades conocidas, como pueden ser el tamaño o la carga, para predecir el efecto de la mutación en la estructura y la función de la proteína. El esquema consta de seis partes, cada una de las cuales corresponde a un párrafo del informe que se da como resultado. Cada parte analiza el efecto de la mutación en uno de los siguientes aspectos del residuo:

- **Contactos**. Interacciones con otras moléculas o átomos.
- **Dominio estructural**. Cualquier parte de la proteína con un nombre específico, y frecuentemente una función, como dominios, repeticiones, etc.
- **Modificaciones**. Características que puede no afecten directamente a la estructura de la proteína pero que pueden afectar en los procesos que se producen después de la traducción.
- **Variantes**. Polimorfismos conocidos, sitios de mutagénesis, etc.
- **Conservación**. La frecuencia relativa de un tipo aminoácido en cada posición tomada de múltiples alineamientos de secuencias.
- **Propiedades de aminoácidos**. Las diferencias en las propiedades conocidas entre aminoácido conocido y el mutado.

Otra característica de HOPE es que proporciona unos rankings de información debido a que la información que obtiene se extrae de múltiples fuentes que no tienen el mismo grado de confiabilidad. Para obtener las conclusiones HOPE hace uso de la información con mayor ranking y por tanto con un mayor grado de confiabilidad.

4.6. Mutalyzer

El paquete Mutation Analyzer (Mutalyzer) [35] tiene como objetivo facilitar la interpretación de los efectos de las variaciones en secuencias. La piedra angular de este paquete software es la herramienta Mutalyzer sequence variation nomenclature checker. Mutalyzer es un programa escrito en Phyton que se basa en la web, por lo cual es de fácil acceso. Mutalyzer está creado con la idea de que sea lo suficientemente flexible para poder permitir futuras extensiones y puede enlazarse con Leiden Open source Variation Database (LOVD) [36].

Las descripciones de las variaciones que ofrece Mutalyzer están basadas las recomendaciones sobre la nomenclatura dada por HUGO-MDI/HGVS. HGVS estableció estas pautas de nomenclatura para funcionar como un estándar.

Mutalyzer está compuesto por varias interfaces web desarrolladas para recoger descripciones de variaciones en secuencias en diferentes formatos. Todos los datos de entrada proporcionados por dichas interfaces web son transformados a un mismo formato de datos y procesados por el motor de Mutalyzer, la herramienta Mutalyzer sequence variation nomenclature checker, que lleva a cabo la comprobación de la nomenclatura de la variación. La anotación de la secuencia de referencia del archivo se analiza para extraer la información necesaria sobre los genes, transcripciones, y proteínas.

Mutalyzer comprueba el tipo de secuencia para aplicar los sistemas de numeración apropiados, que determinan si la parte de la secuencia determinada por las posiciones de inicio y fin de la variación se encuentra en la secuencia de referencia. Si todo es correcto entonces el programa genera el cambio, describe la variación mediante la aplicación de las reglas de nomenclatura, comprueba si el nombre generado por el programa coincide con el introducido, y devuelve la descripción de la variación correcta en diferentes formatos o un mensaje de error describiendo cuál ha sido el fallo del análisis.

A continuación se describen las cuatro diferentes interfaces, para la comprobación de las descripciones de las variaciones, que conforman Mutalyzer: (1) Sequence Variant Description Generator Input ,(2) Sequence Variant Description Checker Input, (3) SNP Converter Input y, (4) Batch Sequence Variant Description Checker Input.

4.6.1. Sequence Variant Description Generator Input

Esta herramienta permite al usuario introducir el accesion number de la secuencia de referencia, indicar el tipo de secuencia (Genómica, ADN codificante, ADN no codificante, RNA, Mitocondrial ADN, proteína o EST), el tipo de variación y la posición inicial y final de la variación (Ilustración 25).

Una vez se insertan estos datos, mutalyzer ofrece la descripción de la variación siguiendo las reglas de nomenclatura. Esta información se pude observar en la parte inferior de la interfaz en forma de link, el cual enlaza la información obtenida con la herramienta Sequence Variant Description Checker Input.

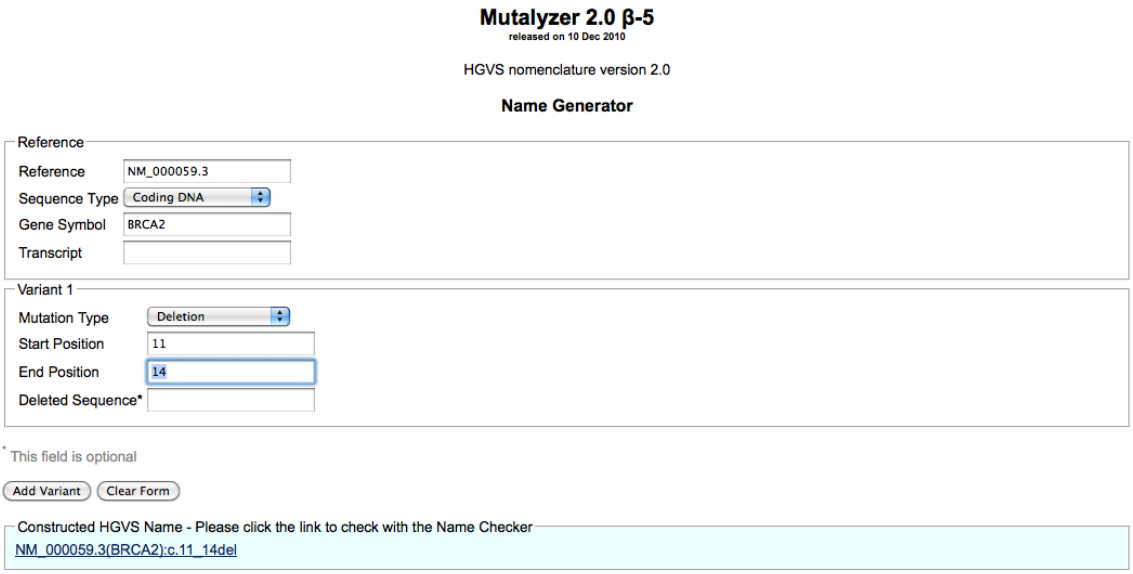

Ilustración 25 Interfaz herramienta Sequence Variant Description Generator Input

Debido a la redundancia de los codones las secuencias de tipo proteínico contienen información insuficiente para permitir una traducción inversa a ADN codificante y poder verificar sin ambigüedad las variaciones. Es por ello que la capacidad que tiene Mutalyzer de revisar descripciones usando referencias proteínicas es muy limitada, y por eso en su nueva versión esta opción no es accesible.

4.6.2. Sequence Variant Description Checker Input

Esta herramienta sirve para poder verificar las descripciones que hemos creados sobre las variaciones antes de publicarlas. Por tanto la información de entrada que necesita dicha herramienta es la de la descripción de la variación conforme a las reglas HGVS. Mutalyzer crea internamente, a partir de la información extraída a partir de la información introducida por el usuario, una descripción propia que compara con la introducida. En el caso de que ambas coincidan entonces se aceptará como correcta la descripción del usuario.

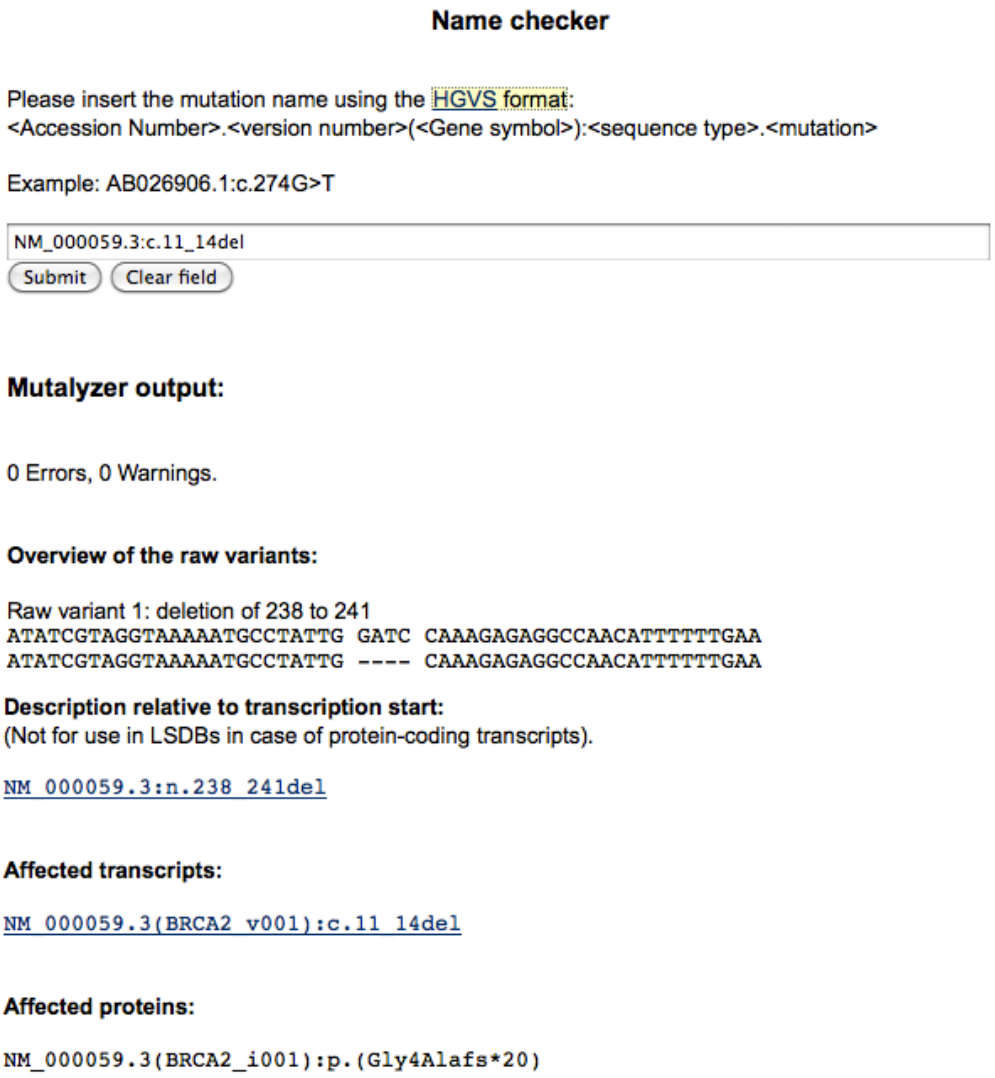

Ilustración 26 Interfaz Sequence Variant Description Checker Input

En el caso de ser correcta, esta herramienta ofrece información sobre la variación insertada. Dicha información se podría dividir en tres apartados: (1) descripción de la variación en diferentes formatos (Ilustración 26), (2) información sobre los transcritos (Ilustración 27) y la proteína predicha y (3) la información referida al fenotipo (Ilustración 28).

Detailed information about the selected transcript and predicted protein:

Reference protein:

```
  1  MPIGSKERPT FFEIFKTRCN KADLGPISLN WFEELSSEAP PYNSEPAEES EHKNNNYEPN
 61  LFKTPQRKPS YNQLASTPII FKEQGLTLPL YQSPVKELDK FKLDLGRNVP NSRHKSLRTV
121  KTKMDQADDV SCPLLNSCLS ESPVVLQCTH VTPQRDKSVV CGSLFHTPKF VKGRQTPKHI
181  SESLGAEVDP DMSWSSSLAT PPTLSSTVLI VRNEEASETV FPHDTTANVK SYFSNHDESL
241  KKNDRFIASV TDSENTNQRE AASHGFGKTS GNSFKVNSCK DHIGKSMPNV LEDEVYETVV
301  DTSEEDSFSL CFSKCRTKNL QKVRTSKTRK KIFHEANADE CEKSKNQVKE KYSFVSEVEP
361  NDTDPLDSNV ANQKPFESGS DKISKEVVPS LACEWSQLTL SGLNGAQMEK IPLLHISSCD
421  QNISEKDLLD TENKRKKDFL TSENSLPRIS SLPKSEKPLN EETVVNKRDE EQHLESHTDC
481  ILAVKQAISG TSPVASSFQG IKKSIFRIRE SPKETFNASF SGHMTDPNFK KETEASESGL
```

Protein predicted from variant coding sequence:

```
1  MPIAKRGQHF LKFLRHAATK QI*
```

Additional information about the transcript:

Exon information:

Number	Start (g.)	Stop (g.)	Start (c.)	Stop (c.)
1	1	188	-227	-40
2	189	294	-39	67
3	295	543	68	316
4	544	652	317	425
5	653	702	426	475
6	703	743	476	516
7	744	858	517	631
8	859	908	632	681
9	909	1020	682	793
10	1021	2136	794	1909
11	2137	7068	1910	6841
12	7069	7164	6842	6937

CDS information:

	g.	c.
Start	228	1
Stop	10484	10257

Ilustración 27 Información transcritos y proteína

Mutalyzer indica el fenotipo asociado a la variación introducida, sin embargo no ofrece una bibliografía asociada a esta información que dé un grado de corroboración. Por el contrario ofrece la posibilidad de descargarse la secuencia de referencia contra la que se ha contrastado la variación.

Legend:

Name	ID	Locus tag	Product	Link method
BRCA2_v001	NM_000059.3			exhaustion
BRCA2_i001	NP_000050.2		breast cancer type 2 susceptibility protein	exhaustion

Links:

Download this reference sequence file: NM_000059.3.gb

Ilustración 28 Información fenotípica

4.6.3. SNP Converter Input

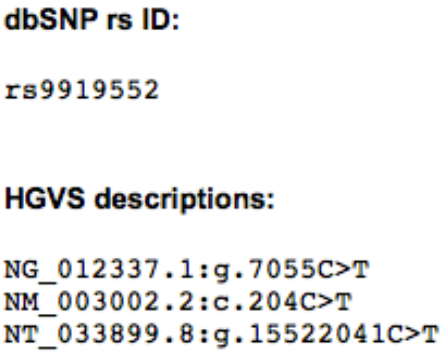

Ilustración 29 Traducción rs a formato HGVS

El principal objetivo de esta herramienta es el de ayudar a los revisores con la transformación al formato HGVS antes de que sean introducidas en la base de datos de dbSNP. Como dato de entrada el programa necesita el identificador rs de la variación en dbSNP. Mutalyzer devolverá las descripciones correspondientes en formato HGVS, como se puede ver en la Ilustración 29 , tanto a nivel génico, mRNA y proteínico.

4.6.4. Batch Sequence Variant Description Checker Input

Esta herramienta fue creada para satisfacer las necesidades de los autores, revisores y comisarios de bases de datos que han de comprobar múltiples cambios en secuencias teniendo que llevar a cabo una revisión por cada variación, lo que conlleva por tanto a una pérdida considerable de tiempo.

Como datos de entrada se necesita un archivo compuesto de tres columnas: accesion number de la secuencia de referencia, el símbolo de referencia aprobado por HGNC, y la descripción de la variación de la secuencia en formato HGVS. Un ejemplo del fichero de entrada sería el siguiente:

AB026906.1(SDHD):g.7872G>T
NM_003002.1:c.3_4insG
AL449423.14(CDKN2A_v002):c.5_400del

Cada línea del documento es analizada y procesada por el módulo principal de Mutalyzer. Los resultados pueden ser enviados al e-mail del usuario o pueden descargarse, una vez la ejecución haya acabado, a través de un enlace que se proporciona en la web.

4.7. Comparativa

Las anteriores herramientas comparten ciertas características y a su vez tienen diferencias destacables que las hacen muy diferentes, todo esto queda resumido en la Tabla 2.

Una de las características más innovadoras que buscan los biólogos en las herramientas de diagnóstico genético es la posibilidad de obtener un fenotipo asociado a las variaciones que se encuentren en el estudio.

En este sentido tan sólo Mutalyzer ofrece esta opción ya que dada una variación, si ésta es expresa fenotípicamente, se asocia el fenotipo que desencadena como se puede ver en la Ilustración 28. Sin embargo, herramientas como SIFT, PolyPhen-2, Alamut o HOPE tratan también el tema fenotípico aunque no en modo tan destacable como Mutalyzer.

Tanto HOPE como Poly-Phen2 se centran en mostrar el cambio que se produce en la estructura de la proteína, prediciendo por consiguiente las consecuencias que se pueden desencadenar. Por otra parte, SIFT trata este apartado desde la diferenciación entre cambios muy deletéreos y polimorfismos neutros. Es decir, no se especifica concretamente qué fenotipo es el asociado al cambio aunque sí trata de algún modo el resultado de dicha variación. Por lo que respecta a Alamut, el trato que ofrece en este aspecto, se limita a la inserción manual del fenotipo para asociarlo así a la variación escogida. Esta herramienta cuenta con una característica que la distingue del resto, y es que permite la introducción de información del paciente, algo que está últimamente de moda y se conoce como la medicina personalizada. Sin embargo se deben de tratar con mucho cuidado estos temas pues están reñidos con las normas éticas, asunto que cae fuera del ámbito de este proyecto.

Entrando en el tema de las variaciones todas estas herramientas están encaminadas al estudio de las variaciones humanas, a excepción de Atlas y Mutalyzer que abarcan otras muchas especies. El problema de algunas de estas herramientas es que tan sólo posibilitan el estudio de un determinado tipo de variación, como por ejemplo es el caso de HOPE, SIFT o PolyPhen-2 que tan sólo permiten el estudio de sustituciones.

En el diagnóstico genético es imprescindible el estudio de variaciones, pero ofrecer algún tipo de ayuda en la creación de un informe clínico que se pueda ofrecer al doctor es una característica que incrementa el valor de la herramienta proporcionada de cara al usuario.

De las herramientas vistas en este capítulo tan sólo tres de ellas ofrecen esta opción. HOPE es quizás la herramienta más especializada en la realización de informes estando éstos diseñados para ser mostrados al usuario final, en el sentido de que ofrecen campos visuales como imágenes y animaciones además de la parte textual pertinente. Los informes a su vez pueden ser almacenados por el usuario. Alamut, al igual que HOPE, permite que el bioinformático pueda descargar el informe generado. Sin embargo, Alamut tan sólo genera informes de las variaciones que han sido introducidas anteriormente por el usuario, dejando así imposibilitada la opción de producir informes de las variaciones ya existentes en el sistema. Por otra parte, PolyPhen-2 genera informes visuales y bien redactados en los que se añade la bibliografía de las variaciones descritas. No obstante, dicha herramienta no permite la exportar estos informes para ser almacenados o tratados más adelante.

La única herramienta que puede ser utilizada como ayuda en el diagnóstico génico es Alamut. El resto de herramientas que proporcionan un informe no proporciona la suficiente información como para poder dar un diagnóstico clínico. Tanto SIFT como PolyPhen-2 se basan en modelos predictivos y por tanto no pueden ser utilizados para con fin.

En la parte tecnológica tanto SIFT como HOPE utilizan la herramienta BLAST para llevar a cabo los alineamientos de secuencias. HOPE además se nutre de las tan de moda herramientas e-Science, y por ejemplo recoge toda su información de los repositorios web disponibles. Es por ello que no mantiene la información localmente, sino que la extrae a través de dichas herramientas según la consulta que se realiza. Esto tiene ciertas ventajas, sobretodo en cuanto

al mantenimiento. Sin embargo, tal y como hemos visto, en el dominio genómico existe una enorme diversidad de datos, una gran redundancia de información e incluso ciertos términos non consensuados. Es por ello que ofrecen un ranking de información para que el usuario pueda discernir entre la información es de más relevancia y la que no. Por su parte, PolyPhen-2 utiliza un clasificador probabilístico para obtener el grado de daño de las variaciones que se estudian.

Precisamente Alamut, que es la herramienta que ayuda al diagnóstico genético, es la única herramienta de pago de entre todas las estudiadas. PolyPhen-2 posee una licencia libre, y es gratuita siempre y cuando el uso que se haga de ella sea académico. Por su parte HOPE todavía no está disponible, y tan sólo se puede acceder a una versión de prueba a través de la web. En lo referente al acceso a dichas herramientas Alamut tan sólo es accesible localmente, lo que hace necesaria su instalación en la máquina en la que se desee utilizar. Atlas, SIFT y PolyPhen-2 también son accesibles localmente, pero a su vez posibilitan un acceso a través de la web. Este último tipo de acceso es el que ofrecen también HOPE y Mutalyzer.

En cuanto a la entrada y salida de datos SIFT, PolyPhen-2 y HOPE esperan la entrada manual de una posición de cambio, el cambio producido y la secuencia en la que esto se ha expresado. En el caso de HOPE y PolyPhen la secuencia a estudiar ha de ser obligatoriamente una secuencia de aminoácidos. SIFT es un poco más abierto en este sentido y permite también la introducción de otro tipo de datos como pueden ser una lista de cromosomas y alelos, códigos de SNPs, etc. Como salida tanto SIFT como PolyPhen-2 ofrecen una visión predictiva. Tanto HOPE como PolyPhen-2 ofrecen como salida un informe.

	Fenotipo	Especies	Licencia	Diag.	Bibliografía	Tecnología	Informe	Input/Output
Atlas	No	Varias	Libre	No	No	SQL, C++, Perl, Java	No	Query/ Información
SIFT	Sí	Humana	Libre	No	No	Linux, BLAST	No	Posición, secuencia, Variación/ Predicción
PolyPhen	Sí	Humana	Libre, uso académico	No	Sí	Clasif. Bayes NAïves	Sí	Posición, secuencia, Variación/ Predicción
Alamut	Sí / No	Humana	Pago	Sí	No	Talamut	Sí HGVS	Varias/Varias
HOPE	Sí	Humana	Libre	No	No	e-Science	Sí	Secuencia y cambio / Informe
Mutalyzer	Sí	Varias	Uso libre	No	No	BLAST	No, HGVS	Varias / HGVS

Tabla 2 Comparativa de las herramientas

Alamut por su parte actúa de una forma más pautada, es decir, primero precisa de la introducción del gen que se desea estudiar. Después es el usuario quien ha de navegar por la aplicación para localizar la posición donde se produce la variación genómica. O introducir a mano toda la información de la variación que desea guardar en el sistema.

En lo referente a la entrada de datos por parte de Mutalyzer depende totalmente de la herramienta que se elija pudiendo ir desde una descripción de una variación en formato HGVS a un listado con el accesion number de la secuencia de referencia, el tipo de secuencia, el tipo de variación y las posiciones inicial y final de la variación. Lo mismo ocurre con la salida, depende del programa que se esté utilizando. Podemos encontrarnos con una descripción HGVS de una variación como con un listado de correcciones de descripciones de variaciones en las que se indiquen además de si son correctas o no, el fenotipo que tienen asociado.

La herramienta Atlas es la que más difiere del resto, en el sentido de que actúa a través de aplicaciones que utilizan un API para llevar a cabo la explotación de la información de la base de datos. Dichas aplicaciones se basan en líneas de comando Unix.

Ejemplo ilustrativo

Teniendo en cuenta las herramientas comentadas en el estado del arte, podemos identificar ciertas carencias que dificultan el trabajo del investigador. El primer conflicto es claro en cuanto a la necesidad de tener que introducir ciertos datos manualmente, en el sentido de que por ejemplo muchas de estas herramientas, relacionadas con la interpretación fenotípica, necesitan la posición de la variación y el cambio que se ha producido para poder iniciar su análisis. Para poder introducir dicha información, es necesario un estudio previo en el que la secuencia de ADN de la que se desea extraer la información sea analizada, extrayendo así las variaciones que se encuentren en ella. Esto conlleva diferentes pasos: (1) el análisis de la secuencia, (2) la identificación de las variaciones, (3) el paso de la información de una plataforma a otra, (4) el análisis de las variaciones con la segunda herramienta y (5) la búsqueda bibliográfica de dichas variaciones entre las múltiples bases de datos disponibles a través de la red.

Una vez se ha llevado a cabo el análisis de variaciones es necesario que éstas queden bibliográficamente documentadas, de este modo se garantiza la veracidad del estudio. Sin embargo, este apartado se convierte en un segundo punto de dificultad para el biólogo, ya que de todas las aplicaciones anteriores tan sólo PolyPhen-2 ofrece un listado bibliográfico adjunto a las variaciones estudiadas. Por tanto, generalmente es el biólogo quien ha de buscar manualmente dicha información entre todos los repositorios de información esparcidos por Internet consumiendo excesivo tiempo.

Una de las tareas más frecuentes en el dominio de la genómica es el análisis de secuencias genómicas [6]. Para realizar un estudio genético, un biólogo ha de llevar a cabo diferentes fases (Ilustración 30). La primera (A) es la extracción de una muestra de ADN del paciente que se desea estudiar y obtener su secuencia. Este proceso de secuenciación se puede llevar a cabo de diferentes formas, como por ejemplo mediante el método enzimático de Sanger o empleando cebadores fluorescentes. Una vez los secuenciadores de ADN han proporcionado una secuencia de estudio, ésta debe ser analizada por el biólogo y se ha de verificar que el proceso de secuenciado ha sido ejecutado con éxito, sin errores. Dada dicha secuencia en formato digital, el biólogo debe identificar (B) manualmente o mediante la ayuda de algún programa las diferencias que pueda haber entre ésta y la secuencia de referencia de la especie estudiada. Teniendo dichas variaciones identificadas, el siguiente paso es buscar (C) en las diferentes bases de datos accesibles en Internet cada una de las variaciones encontradas de modo que éstas puedan ser referenciadas bibliográficamente y, a poder ser, relacionadas con su causa fenotípica. Este paso puede conllevar bastante tiempo dependiendo de la suerte que tenga el biólogo en su búsqueda, es decir, si la variación que ha de buscar es encontrada en la primera base de datos accedida el gasto de tiempo es mínimo, pero si no es así el biólogo deberá cambiar el repositorio de búsqueda en el que está buscando hasta que la halle en alguno. En este último supuesto, la búsqueda de una variación se puede demorar horas y teniendo en cuenta que a veces se obtiene más de una variación en la comparación entre secuencias, este proceso podría alargarse de tal modo que llegara a contabilizarse por días.

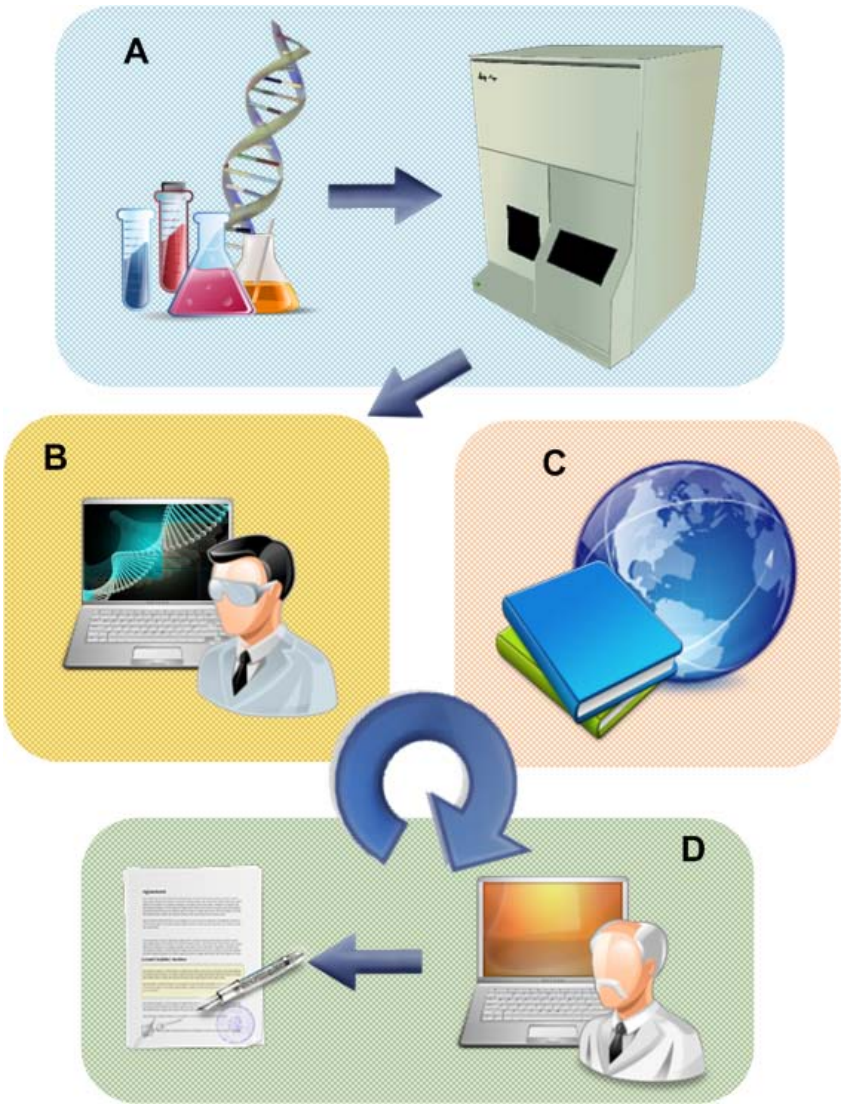

Ilustración 30 Fases de creación de un informe de diagnóstico génico

Por último, teniendo todas las variaciones encontradas referenciadas bibliográficamente, el genetista genera (D) manualmente un informe con toda ésta información en el que se documentarán todas y cada una de las variaciones encontradas en la cadena de ADN estudiada, indicando: el cambio que se ha producido, cuál o cuáles pueden ser o son las consecuencias de cada variación, su interpretación biológica y clínica, el fundamento biológico del estudio, el método de secuenciación que se ha llevado a cabo y cualquier otro estudio adicional o recomendación que se considere oportuna. Dicho informe será después enviado al médico, que será quien se encargue de realizar el diagnóstico y determinar el tratamiento para el paciente estudiado.

Durante la realización de este trabajo se ha contado con la colaboración del Instituto de Medicina Genómica (IMeGen), empresa biomédica especializada en el análisis genético aplicado al sector médico, que proporciona servicios de diagnóstico genético a hospitales y clínicas de toda España. Gracias a su apoyo como expertos en el dominio, la comprensión del campo de la genética se ha podido llevar a cabo de tal modo que ingeniería del software y genética pueden avanzar en el mismo sentido en una relación de simbiosis. Eliminando de este modo la brecha existente entre ambas disciplinas. El resultado de dicha colaboración

interdisciplinar es el diseño de un modelo conceptual que guía el alineamiento de conceptos entre ambos campos. De modo que el diseño e implementación de artefactos software que componen un GeIS se convierte en un proceso más sencillo.

El proyecto que nos abarca comienza a partir de la definición del esquema conceptual Conceptual Schema of The Human Genome (CSHG) [15, 16]. Como consecuencia dos ramas de investigación se abren para dar paso a diferentes proyectos: (1) Creación y carga de una base de datos basada en HGDB [37, 38] y (2) la explotación de la información depositada en dicha base de datos (Ilustración 31).

Ilustración 31 Investigación grupo Genoma

Es en esta última rama en donde se centraría este trabajo, en la explotación de la base de datos creada a partir del modelo conceptual CSHG.

Teniendo en cuenta las dificultades comentadas anteriormente, el proceso que se sigue para realizar un diagnóstico génico y la existencia de la base de datos Human Genome Data Base (HGDB), se pretende llevar a cabo un estudio en el que se observen las dificultades intrínsecas del dominio así como las dificultades adquiridas por la no demasiado adecuada gestión de éste a la hora de producir diferentes herramientas software de diagnóstico genético. Como resultado de dichas lecciones aprendidas se pretende desarrollar un prototipo para la creación de un informe de diagnóstico génico, llevando a cabo dicho proyecto desde el punto de vista de la ingeniería del software y los sistemas de información.

El caso de estudio que se aborda en este trabajo es el de la creación de un informe genético a partir de la búsqueda de las variaciones existentes en una cadena de ADN. En este supuesto será el biólogo quien indique el gen sobre el que se desea llevar a cabo el estudio e introduzca una secuencia de ADN a analizar. Para que el estudio se realice tan sólo será necesaria dicha información. Esto evita que el biólogo encargado del análisis deba realizar un estudio previo para determinar las variaciones que se encuentran en la secuencia, disminuyendo por tanto el tiempo de espera y la incomodidad de tener que introducir una a una, y a mano, las diferentes variaciones a analizar. La disminución del tiempo de creación de un diagnóstico clínico es de suma importancia ya que es bien sabido que la detección temprana de una enfermedad puede llegar a ser determinante en el tratamiento o la eliminación de ésta.

Por otra parte, la fase de búsqueda manual de la bibliografía que documenta las variaciones será excluida. Esto es debido a la existencia de HGDB. Esta base de datos contiene toda la información relacionada a un gen y sus variaciones, teniendo en cuenta sus referencias bibliográficas, fenotipos asociados, etc. Las ventajas de tener una base de datos relacional que dispone de esta información son varias:

- Unificación de los formatos en los que la información se encuentra en las diferentes fuentes de datos de Internet.
- Facilidad de llevar a cabo consultas SQL, ya sean más o menos complicadas, debido a la propiedad local de los datos.
- Menor tiempo de consulta de datos.
- Información relacionada y gestionada a partir de un modelo conceptual.

Por tanto, todas las carencias o dificultades que habían sido expuestas quedan resueltas.

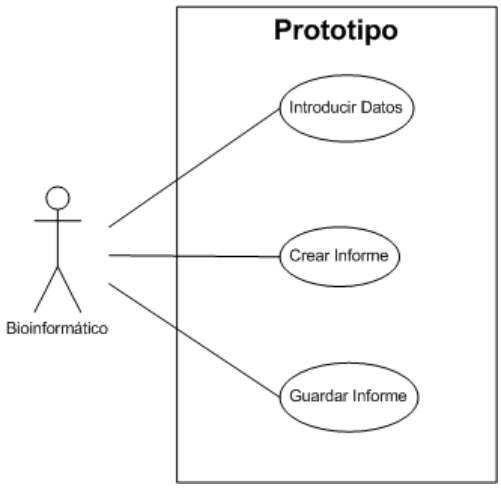

Ilustración 32 Funciones bioinformático

Con la realización de este estudio se busca automatizar de la mayor forma posible la creación de un diagnóstico génico. Sin embargo, tras analizar varios casos se ha llegado a la conclusión que una automatización completa es actualmente imposible. La fase de introducción de datos no es suprimible, de hecho es preferible que haya una interacción entre el usuario y la aplicación encargada de la generación del informe. Sin embargo, cuánto menor sea el trabajo a realizar por el usuario, más sencilla será la utilización y el aprendizaje del software. Por ahora es imposible la eliminación y la automatización del procesado preliminar de la información, que es lo que comprende: (1) la extracción de la muestra de ADN, (2) la secuenciación del ADN, (3) el limpiado de la secuencia y paso de un electroferograma a una secuencia de letras, correspondiente a los aminoácidos. Estas tareas requieren de la mano de expertos, de la experiencia que han obtenido a lo largo de años de investigación. Sin embargo, los pasos que han de desempeñar una vez realizados estos primeros pueden reducirse. Así pues, en este trabajo se quiere estudiar la forma en la que se puede automatizar la extracción de las variaciones a partir de la alineación de la secuencia de ADN estudiada. De este modo, la fase de introducción de datos se ve reducida en este sentido a la elección del gen de estudio y la inserción de la secuencia de ADN que se quiere analizar, como se puede observar en la

Ilustración 33 . La idea es que el usuario tan sólo deba introducir dichos datos, pudiendo así crear un informe y guardarlo (Ilustración 32).

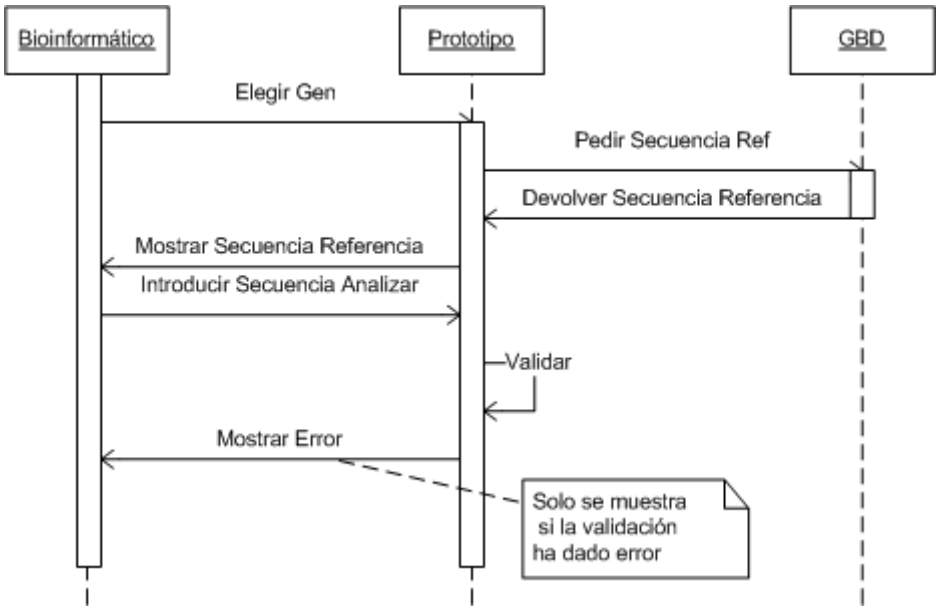

Ilustración 33 Introducción de datos

Para la fase de creación el modo en el que se elabora el informe ha de ser invisible de cara al usuario, éste tan sólo introducirá los datos de inicio y el resto de operaciones se llevarán a cabo en segundo plano (Ilustración 34). En esta fase se quiere abordar la recolección de la información introducida por el usuario, en la fase previa, y que ya ha sido validada.

Una vez recopilada toda la información el segundo paso que se ha de llevar a cabo para generar el informe es extraer las variaciones de la secuencia de ADN que ha sido introducida para ser estudiada. Para ello se debe realizar un alineamiento entre la secuencia de referencia del gen y la cadena introducida. Las diferencias que se encuentren en dicho alineamiento serán las variaciones que deberán reportarse en el informe más tarde. Así pues es requisito indispensable buscar una forma de alinear ambas secuencias, ya sea mediante la utilización de alguna aplicación, el uso de algún algoritmo o la implementación de algún otro. Este estudio se ha llevado a cabo en este proyecto y su descripción se realiza en un capítulo más adelante. Teniendo localizadas las variaciones en la secuencia, la siguiente fase es la de la búsqueda de dichas variaciones en la base de datos HGDB. El hecho de que una variación se encuentre en la base de datos implica que dicho cambio ha sido documentado y estudiado previamente, y por tanto se puede ofrecer mucha más información sobre la variación que su simple posición y cambio de aminoácidos.

Las variaciones que se encuentran en la base de datos tienen asociada su posición, el cambio que originan, el fenotipo asociado, la base de datos de la cual se extrajo la información y la bibliografía que documenta dicha variación.

El último paso que queda para llevar a término esta fase es la compilación de un informe en el que se reúna toda la información obtenida de una forma fácil de entender e interpretar para el usuario. Debe tenerse en cuenta las necesidades del biólogo, por ejemplo, poder ofrecer la información de forma redactada y no sólo de manera esquemática facilita su trabajo, con el mismo propósito se puede pensar en la usabilidad que brinda el ofrecer enlaces a la documentación asociada a la variación encontrada así como ofrecer la descripción de las variaciones mediante el uso de algún formato estándar.

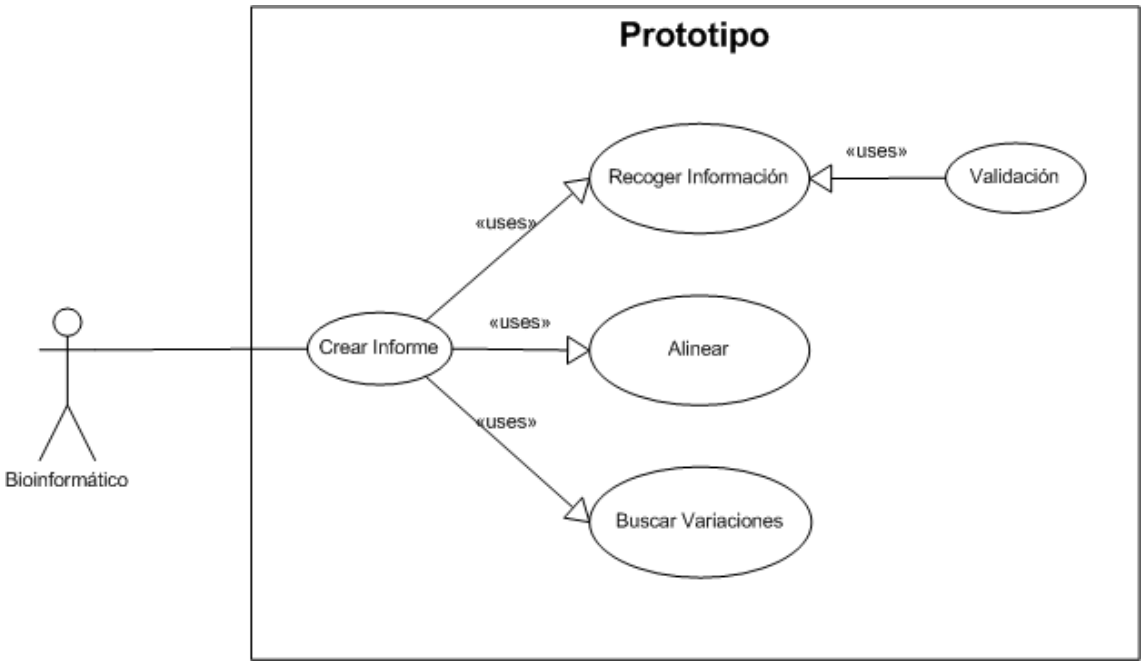

Ilustración 34 Fase de creación del informe

Así pues la fase de creación del informe constaría de cuatro partes bien diferenciadas: (1) recopilación de la información, (2) fase de alineamiento, (3) fase de búsqueda de las variaciones y, (4) fase de composición del informe y validación científica de los resultados (Ilustración 35).

La última fase que quedaría por especificar sería la de guardar el informe. En el ámbito de la genómica, al igual que en el de la medicina, es importante guardar un histórico sobre los estudios que se llevan a cabo con el propósito no sólo de almacenar la información, sino de poderla comparar a la hora de hacer diferentes estudios o revisar diagnósticos. Es por ello que ofrecer la posibilidad de almacenar automáticamente la información que se está representando en pantalla es sin duda un servicio necesario. Esta fase (Ilustración 36) constará principalmente de dos partes: (1) la comprobación de los datos, ya sean del informe como de la ruta donde el usuario desea almacenar el informe creado y, (2) la creación de dicho informe, mediante la compilación de toda la información mostrada en pantalla de modo que sea plasmable en otro formato de datos sin la pérdida de información que pudiera suponer este cambio de formato.

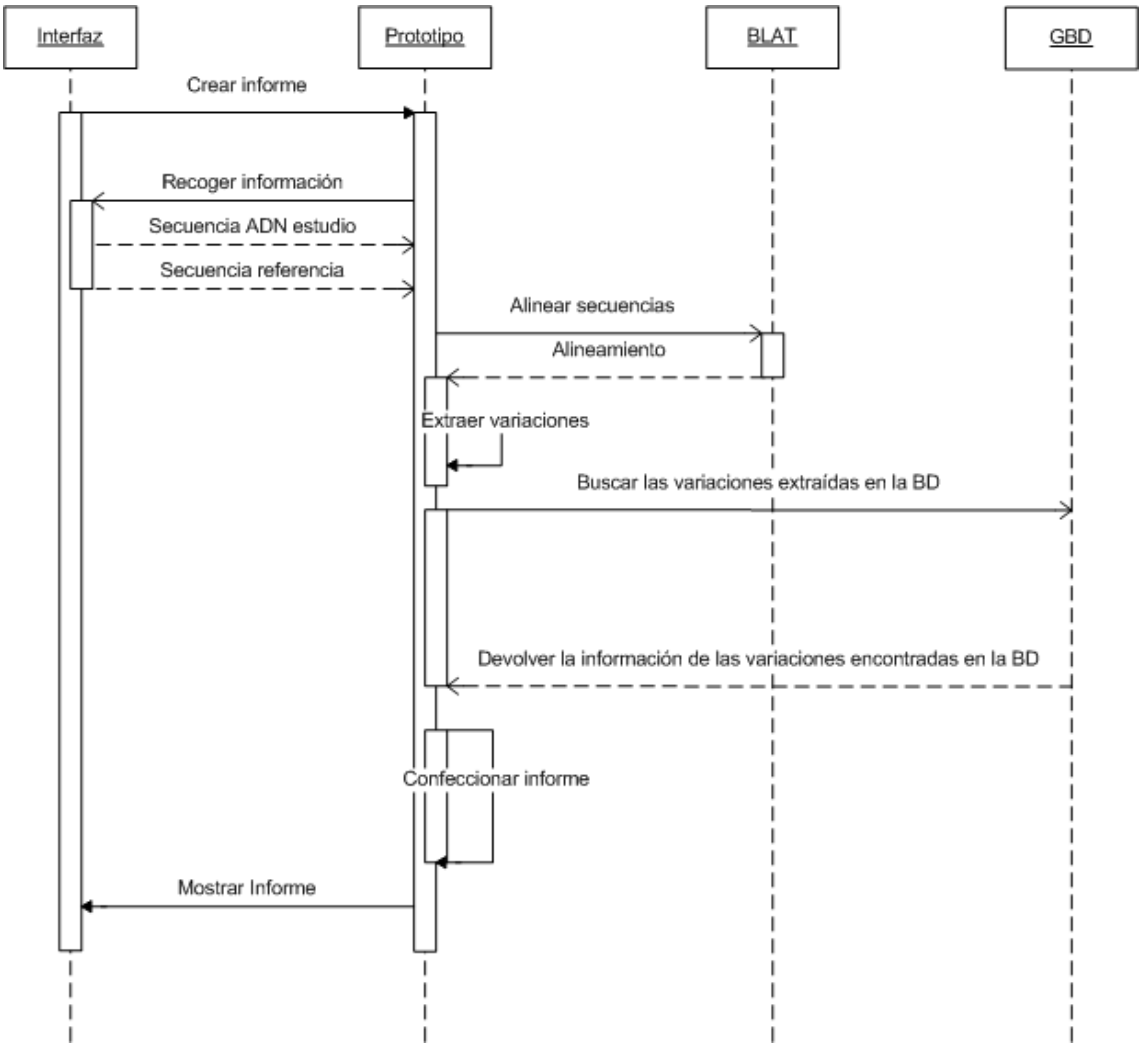

Ilustración 35 Diagrama de secuencia de la fase de creación del informe

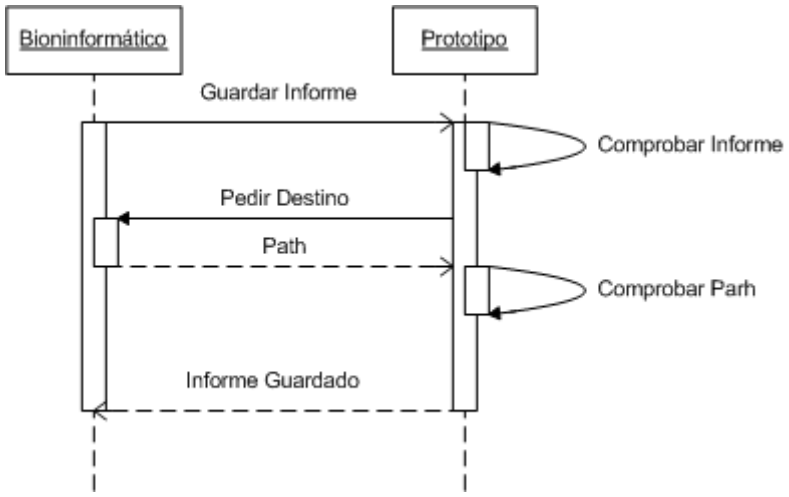

Ilustración 36 Diagrama de secuencia para la fase de almacenamiento del informe

Así pues, como resumen podemos decir que a partir de la sustracción de los requisitos demandados por Imegen y extraídos del estudio tanto de las herramientas expuestas en el capítulo del estado del arte como del caso de estudio planteado, se ha podido determinar la especificación de requisitos que debe cubrir este trabajo. Como punto clave se ha establecido la minimización del esfuerzo que debe realizar un genetista a la hora de llevar a cabo un

diagnóstico génico. Para obtener dicha disminución, se debe solventar tanto los problemas encontrados en la introducción de datos y el uso de diferentes programas para obtenerlos, como la traducción a un formato estándar de las variaciones encontradas y la búsqueda de información a través de internet de las variaciones encontradas. Este segundo punto es en el que más hincapié ha hecho Imegen, siendo para ellos su talón de Aquiles.

Alineamiento

Uno de los requisitos que se comentaron en el capítulo anterior es el de la disminución del trabajo que ha de realizar el biólogo. Se quiere llevar a cabo un proceso en el cual la interacción con el biólogo sea mínima pero produciendo una respuesta máxima por parte del software. De este modo el biólogo se ve liberado de la carga de trabajo que supone el tener que llevar a cabo diferentes estudios previos para obtener las variaciones a partir del alineamiento de una secuencia de ADN con otra de referencia.

Sin duda la obtención de la muestra de ADN no podrá ser automatizada por el momento. Del mismo modo, la limpieza de la secuencia que se extraiga de esa muestra ha de realizarse de una manera manual, bajo la mirada crítica del biólogo.

Cada gen tiene una secuencia de ADN asociada, que se denomina secuencia de referencia y que es una secuencia consenso basada en la secuencia de varios individuos de la misma especie. La base de datos RefSeq contiene una colección pública de secuencias de nucleótidos, ya sean como ADN o ARN, y sus correspondientes productos proteicos disponibles. RefSeq fue construida por NCBI, por lo tanto es de acceso libre y proporciona un registro único de cada molécula biológica para organismos modelo. Para cada modelo de organismo, RefSeq tiene como objetivo proporcionar un conjunto completo de secuencias no-redundante y bien anotado que incluye ADN genómico, transcritos y proteínas. RefSeq es una fundación de estudios médicos, funcionales y diversos que proporciona una referencia estable para la anotación del genoma, la identificación y caracterización de genes, la mutación y el análisis de polimorfismos, estudios de expresiones, y análisis comparativo.

Para llevar a cabo la búsqueda de las variaciones en una secuencia de ADN de un paciente se debe realizar un alineamiento entre ésta y la secuencia de referencia del gen correspondiente.

En términos coloquiales, alinear dos secuencias es poner una junto a la otra de forma que se resalten las diferencias y similitudes, pero sin cambiar el orden de los residuos. Entendiendo como residuo las partes que no casen en la alineación. Existen formas diversas de hacer esto, aunque unas se prestan más a análisis que otras. Si por ejemplo queremos saber qué secuencia es más larga de dos dadas, podemos alinear simplemente el primer residuo de la primera cadena con el primero de la segunda y así sucesivamente. Dando como resultado algo como lo que se muestra en la Ilustración 37. De este alineamiento se puede ver fácilmente como la secuencia 2 es más larga que la secuencia 1.

```
ATTTGCTA      Secuencia 1
ATTGCTGTATTC  Secuencia 2
```

Ilustración 37 Alineamiento básico

No obstante, esta información no suele ser la que interesa. Normalmente es más interesante saber si dos secuencias tienen sub-secuencias iguales en el mismo orden. En bioinformática, el alineamiento de secuencias se define como una forma de representar y comparar dos o más secuencias de ADN, ARN, o estructuras primarias proteicas para resaltar sus zonas de similitud, pudiendo indicar relaciones funcionales o evolutivas entre los genes o proteínas consultados.

Los alineamientos de secuencias de aminoácidos proporcionan una herramienta poderosa para comparar secuencias relacionadas, permitiendo detectar orígenes evolutivos similares y

representar una estructura común y/o un rol catalítico. Si dos secuencias en un alineamiento comparten un ancestro común, las no coincidencias pueden interpretarse como mutaciones puntuales (sustituciones), y los huecos en alguna de las dos secuencias como indels (mutaciones de inserción o eliminado) introducidas en uno o ambos linajes en el tiempo que transcurrió desde que divergieron. En el alineamiento de secuencias proteicas, el grado de similitud entre los aminoácidos que ocupan una posición concreta en la secuencia puede interpretarse como una medida aproximada de conservación en una región particular, o secuencia motivo, entre linajes. La ausencia de sustituciones, o la presencia de sustituciones muy conservadas (la sustitución de aminoácidos cuya cadena lateral tiene propiedades químicas similares) en una región particular de la secuencia indica que esta zona tiene importancia estructural o funcional. Aunque las bases nucleotídicas del ADN y ARN son más similares entre sí que con los aminoácidos, la conservación del emparejado de bases podría indicar papeles funcionales o estructurales similares.

Las secuencias muy cortas o muy similares pueden alinearse manualmente. Sin embargo, los problemas más interesantes necesitan del alineamiento de secuencias largas, muy variables y extremadamente numerosas que no pueden ser alineadas por humanos. El conocimiento humano se aplica principalmente en la construcción de algoritmos que producen alineamientos de alta calidad, y también en el ajuste del resultado final para representar patrones que son difíciles de introducir en algoritmos, especialmente en el caso de secuencias de nucleótidos.

```
ALIGN calculates a global alignment of two sequences
 version 2.2u
Please cite: Myers and Miller, CABIOS (1989) 4:11-17
silvestre 175 bp                                    172 aa vs.
mutante 164 bp                                      162 aa
 using matrix file: BLOSUM50, gap open/ext: -14/-4
  85.5% identity in 172 aa overlap;              Global score: 902

               10        20        30        40        50        60
silves GIREFNPBETAGLBINHMSAPIENSMVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVV
              ::::::::: :.:   ..:::::::::.:::::::::::::::::::::::
mutant -------BETAGLBINAMTA---DAMVHLTPEEKSVVTALWGKVNVDEVGGEALGRLLVV
                      10            20        30        40        50

               70        80        90       100       110       120
silves YPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCD
       .:::::::::::::::::::::::::::::::::::::::::::::::::: :::::::::
mutant WPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTTATLSELHCD
               60        70        80        90       100       110

              130       140       150       160       170
silves KLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
       :::::::::   :::::::::: ::::::::::::::::  :::::::: .::
mutant KLHVDPENFAAHGNVLVCVLAHTFGKEFTPPVQAAYQKYKAGVANALAAQYH
              120       130       140       150       160
```

Ilustración 38 Alineamiento global

Las aproximaciones computacionales en el alineamiento de secuencias se dividen en dos categorías: alineamiento global y alineamiento local. Calcular un alineamiento global es una

forma de optimización global que “fuerza” al alineamiento a ocupar la longitud total de todas las secuencias introducidas. Por contra, los alineamientos locales identifican regiones similares dentro de largas secuencias que normalmente son muy divergentes entre sí. A menudo se prefieren los alineamientos locales, pero pueden ser más difíciles de calcular ya que se añade el desafío de identificar las regiones de mayor similitud.

Los alineamientos globales (Ilustración 38) son especialmente útiles cuando lo que se quiere es comparar dos genes en toda su longitud o al momento de ofrecer sobre-solapamientos.

Los alineamientos globales, que intentan alinear cada residuo de cada secuencia, son más útiles cuando las secuencias problema iniciales son similares y aproximadamente del mismo tamaño. Una estrategia general de alineamiento global es el algoritmo Needleman-Wunsch basado en programación dinámica. Sin embargo, cuando se parte de la base de que sólo algunas regiones están conservadas es mejor usar otro método de alineamiento. Tomando como ejemplo dos cadenas iguales introduciendo en la segunda mutaciones de forma indiscriminada, si hacemos un alineamiento global se detectaría que hay cambios entre ambas secuencias pero las regiones que tienen alta conservación no serían destacadas. Es posible que al alinear ambas secuencias globalmente se encuentren regiones conservadas, pero al hacer este tipo de alineamiento se ha de conservar el orden de las secuencias y esto puede llevar a ocultar información.

```
 Comparison of:
(A) ./wwwtmp/.8938.1.seq unknown 98 bp                                  - 97 aa
(B) ./wwwtmp/.8938.2.seq unknown 98 bp                                  - 97 aa
 using matrix file: BL50 (15/-5), gap-open/ext: -14/-4 E(limit)   0.05

  92.7% identity in 41 aa overlap (12-52:48-88); score:  251 E(10000): 2.8e-19

              20        30        40        50
unknow EESNEEMPLCNSTRIDCNFINESDEILSTRARNASITACIN
       .:. ::::::::::::::::::::::::::::::::::::
unknow NEAREEMPLCNSTRIDCNFINESDEILSTRARNASITACIN
        50        60        70        80

-------------------------------------------------------------------------------

 100.0% identity in 37 aa overlap (52-88:15-51); score:  232 E(10000): 2.3e-17

              60        70        80
unknow NENDNDEALINEARLCALMENTEESMERQEALINEAR
       :::::::::::::::::::::::::::::::::::::
unknow NENDNDEALINEARLCALMENTEESMERQEALINEAR
           20        30        40        50

-------------------------------------------------------------------------------

  38.5% identity in 104 aa overlap (1-97:1-97); score:  135 E(10000): 1.4e-07

               10           20        30        40        50
unknow SECENCIAESTEESNE---EMPLCNSTRIDCNFINESDEILSTRARNASITACINENDND
       ::::::::::::::::   .  : : .:. : ......: .  .       : :::  ..
unknow SECENCIAESTEESNENDNDEALINEARL-CALMENTEESMERQE------ALINEAREE
               10        20         30        40              50

        60         70           80        90
unknow EALINEARL-CALMENTEESME---RQEALINEARGLALMENTE
        : : .:. : ......: .    :. ..     :::::::::
unknow MPLCNSTRIDCNFINESDEILSTRARNASITACINGLALMENTE
           60        70        80        90
```

Ilustración 39 Alineamiento local

El alineamiento local (Ilustración 39) permite encontrar subsecuencias que tienen una alta similitud, entendiendo por similitud la observación o medición de parecido y diferencia independientemente del origen de ese parecido. En otras palabras, la similitud es la medida del parecido entre dos secuencias que se puede cuantificar.

Los alineamientos locales son más útiles para secuencias diferenciadas en las que se sospecha que existen regiones muy similares o motivos de secuencias similares dentro de un contexto mayor. El algoritmo Smith-Waterman es un método general de alineamiento local basado en programación dinámica. Con secuencias suficientemente similares, no existe diferencia entre alineamientos globales y locales.

Los métodos híbridos, conocidos como semiglobales o métodos “glocales” intentan encontrar el mejor alineamiento posible que incluya el inicio y el final de una u otra secuencia. Puede ser especialmente útil cuando la parte “aguas arriba” de una secuencia se solapa con la parte “aguas abajo” de la otra. En este caso, ni el alineamiento global ni el local son completamente adecuados: un alineamiento global intentará forzar a la alineación a extenderse más allá de la región de solapamiento, mientras que el alineamiento local no cubrirá totalmente la región solapada.

La búsqueda en las bases de datos con el objetivo de extraer secuencias homólogas es uno de los fundamentos para el análisis de secuencias. Se han desarrollado y aplicado diversos métodos en paquetes de programas y servidores web para llevar a cabo este propósito. Los programas de búsqueda en bases de datos se diferencian en la forma en la que están diseñados los algoritmos que usan. Esto influye en el tiempo de ejecución y la sensibilidad a la hora de realizar los alineamientos. Los algoritmos de alta velocidad usan principios simplificados para establecer la similitud entre secuencias, el tiempo en que tarda en llevarse a cabo el alineamiento depende de la sensibilidad del algoritmo, parámetro que está fuertemente relacionado con la longitud de la secuencia y el tamaño de la base de datos. Por otra parte, los algoritmos basados en métodos de programación dinámica buscan algoritmos locales óptimos siendo en consecuencia lentos para realizar búsquedas en grandes bases de datos. Los programas que utilizan algoritmos FASTA o BLAST, por el contrario, se desarrollaron con el objetivo de ser de alta velocidad y baja sensibilidad, ya que están basados en estrategias heurísticas que concentran sus esfuerzos en las regiones de la secuencia más probablemente relacionadas en un tiempo de ejecución corto, ofreciendo buenos resultados. Las consultas a través de bases de datos públicas en internet constituyen un recurso invaluable para investigadores que están trabajando en el campo de la bilogía molecular, química de proteínas, y diagnóstico molecular.

A continuación se describen algunas de las herramientas bioinformáticas que emplean métodos de alineamiento de secuencias.

6.1. BLAST

Es un conjunto de programas que tiene como objetivo obtener similitudes entre secuencias alineadas. La entidad administradora de Basic Alignment Search Tool (BLAST) [22] es NCBI. BLAST está diseñado para explotar todas las bases de datos disponibles independientemente de que sean proteínas o ADN. El fundamento de los algoritmos de BLAST es comparar

secuencias creando matrices de sustitución generales, como por ejemplo Blosum 62 (Block Substitution Matrix) en las que son propuestos cuáles son los aminoácidos que menos difieren y las mutaciones más frecuentes. A partir de estas matrices se establece una puntuación que indicará la similitud entre pares de secuencias. Es importante mencionar que BLAST usa un algoritmo heurístico, lo cual quiere decir que nunca podrá garantizar que la solución devuelta es la correcta pero sí que el alineamiento devuelto es razonablemente bueno.

A continuación se comentan las partes que componen el resultado típico obtenido a partir de BLAST al someter un pedazo de ADN humano que ha sido secuenciado. El resultado se divide en tres secciones. La primera tiene información general acerca de la búsqueda (Ilustración 40).

Ilustración 40 Sección 1 de los resultados

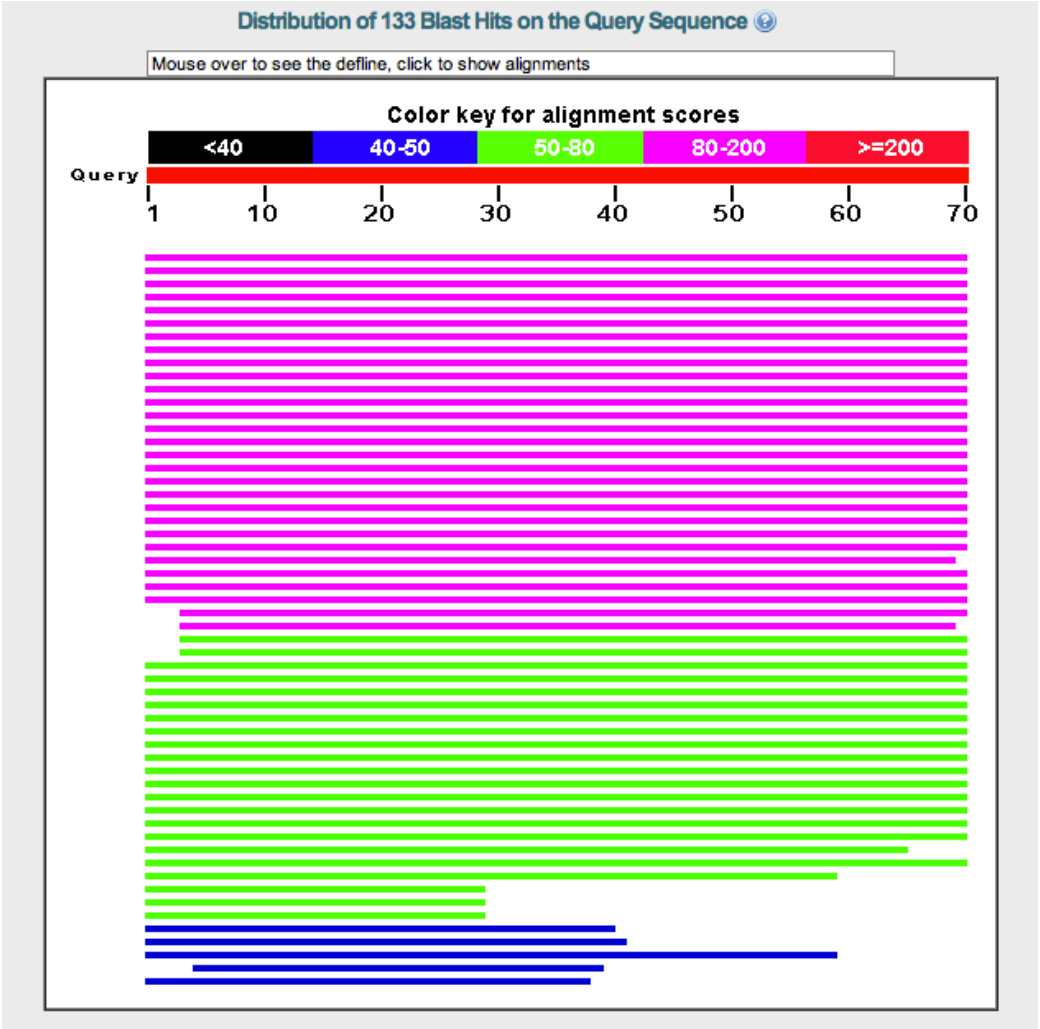

Ilustración 41 Alineamiento BLAST parte gráfica

La segunda sección contiene los lineamientos, primero en forma gráfica (Ilustración 41), diferenciando por colores los porcentajes de identidades, y después en forma de letras y espacios (Ilustración 42).

```
>gi|302313186|gb|HM015597.1|  Homo sapiens sonic hedgehog homolog (Drosophila) (SHH) gene,
complete cds
Length=17425

 Score =  127 bits (140),  Expect = 5e-27
 Identities = 70/70 (100%), Gaps = 0/70 (0%)
 Strand=Plus/Minus

Query  1     CTCGCTCTCTCCCTCGCTGGCTGCCTCGCTCTTTCTCTTCCTATATAACCTTGCCCGCCG  60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  4029  CTCGCTCTCTCCCTCGCTGGCTGCCTCGCTCTTTCTCTTCCTATATAACCTTGCCCGCCG  3970

Query  61    CGGCTGCGGG  70
             ||||||||||
Sbjct  3969  CGGCTGCGGG  3960


>gi|172044658|ref|NG_007504.1|  Homo sapiens sonic hedgehog (SHH), RefSeqGene on chromosome 7
Length=16410

 Score =  127 bits (140),  Expect = 5e-27
 Identities = 70/70 (100%), Gaps = 0/70 (0%)
 Strand=Plus/Minus

Query  1     CTCGCTCTCTCCCTCGCTGGCTGCCTCGCTCTTTCTCTTCCTATATAACCTTGCCCGCCG  60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  5029  CTCGCTCTCTCCCTCGCTGGCTGCCTCGCTCTTTCTCTTCCTATATAACCTTGCCCGCCG  4970

Query  61    CGGCTGCGGG  70
             ||||||||||
Sbjct  4969  CGGCTGCGGG  4960


>gi|34595719|gb|AY307424.1|  Pan paniscus strain NG05253A-Coriell sonic hedgehog (shh) gene,
promoter region and 5' UTR
Length=611

 Score =  127 bits (140),  Expect = 5e-27
 Identities = 70/70 (100%), Gaps = 0/70 (0%)
 Strand=Plus/Minus

Query  1    CTCGCTCTCTCCCTCGCTGGCTGCCTCGCTCTTTCTCTTCCTATATAACCTTGCCCGCCG  60
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  486  CTCGCTCTCTCCCTCGCTGGCTGCCTCGCTCTTTCTCTTCCTATATAACCTTGCCCGCCG  427

Query  61   CGGCTGCGGG  70
            ||||||||||
Sbjct  426  CGGCTGCGGG  417
```

Ilustración 42 Alineamientos en BLAST, formato texto

La tercera sección contiene el listado de todas las secuencias que produjeron alineamientos significativos junto con un valor de puntuación (Score) que representa el grado de similitud entre pares de secuencias. El valor E para una determinada puntuación indica cuántos alineamientos esperamos que por azar alcancen un valor igual o mayor.

6.2. Clustal W

Es un programa que aplica métodos de alineamiento globales de alta velocidad para calcular los niveles de semejanza entre secuencias. No es aconsejable utilizarlo para hacer alineamientos de secuencias con largos sectores disímiles. Es altamente utilizado para realizar alineamientos múltiples de secuencias.

Clustal W (Ilustración 43) acepta un alto rango de formatos de entrada, ofreciendo del mismo modo diferentes formatos de salida.

Para realizar los alineamientos múltiples se pasa por tres etapas. En la primera se realizan alineamientos por pares. En la segunda etapa se crea un árbol filogenético, es decir, un árbol que muestra las relaciones evolutivas entre varias especies que se cree poseen una

ascendencia común. En la última etapa se utiliza este árbol para llevar a cabo el alineamiento múltiple.

Los alineamientos múltiples se utilizan para encontrar patrones de diagnóstico para caracterizar familias de proteínas, detectar o demostrar homologías entre secuencias, ayudar a predecir las estructuras secundarias y terciarias de nuevas secuencias, o para sugerir oligonucléicos para PCRs.

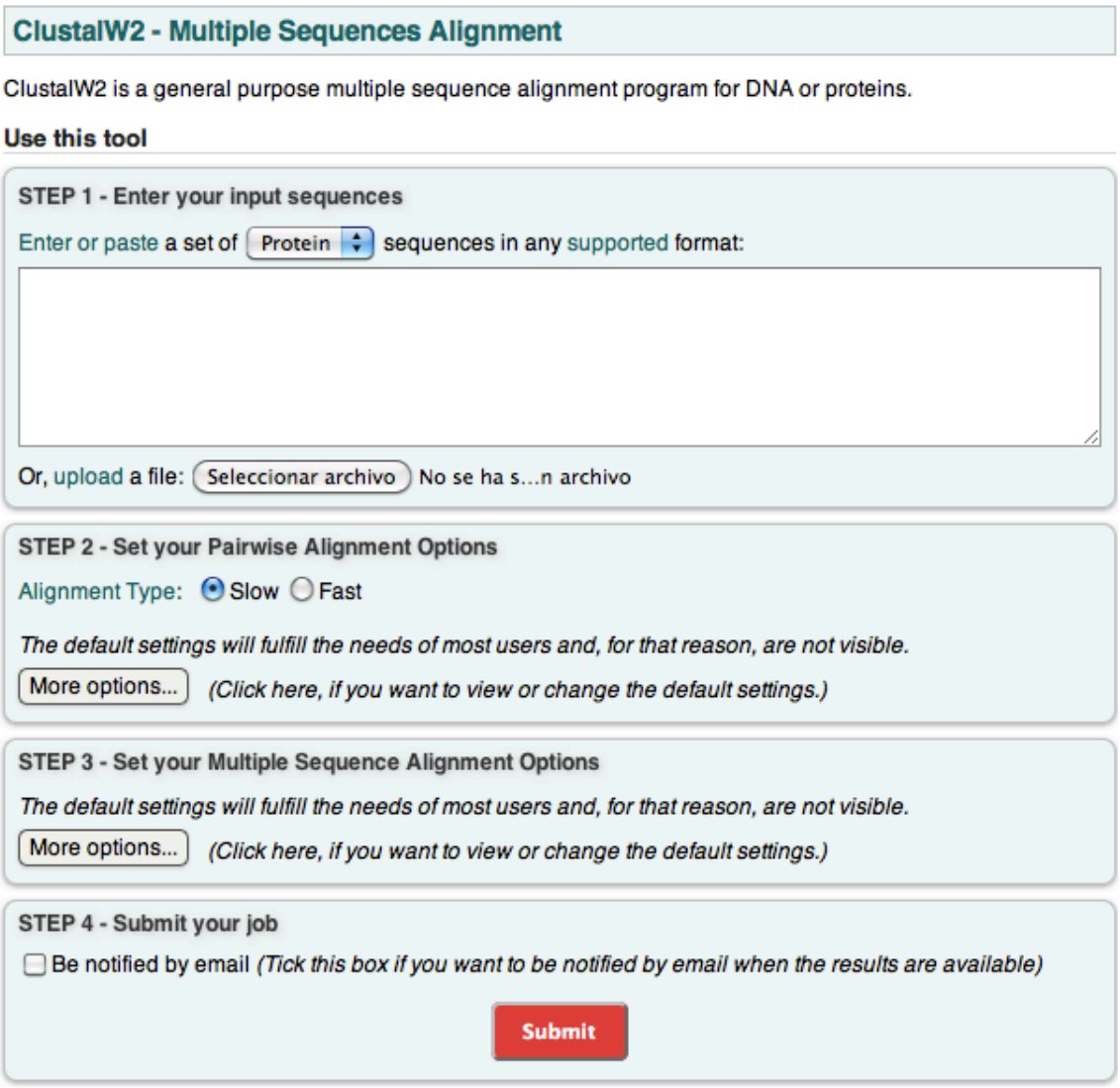

Ilustración 43 Interfaz Clustal W

6.3. BLAT

BLAST-Like Alignment Tool (BLAT) [39] es un programa software desarrollado por Jim Kent en la UCSC para la identificación de similitudes entre secuencias de ADN y proteínas. BLAT es mucho más rápido, en cuanto a los alineamientos de proteínas y nucleótidos, que otras herramientas más antiguas como BLAST. Concretamente es 500 veces más rápido para alineamientos de secuencias de ADN, y 50 veces para alineamientos de proteínas.

BLAT es similar a BLAST en varios aspectos. El programa escanea rápidamente para secuencias relativamente cortas, y extiende esto a pares con puntuaciones altas. Sin embargo, BLAST

difiere de BLAT en algunos aspectos significativos. Mientras que BLAST construye un índice de la secuencia y luego escanea linealmente la base de datos, BLAT construye un índice de la base de datos y luego escanea linealmente la secuencia, el índice tan sólo se construye una vez por cada gen. Por otra parte, mientras que BLAST devuelve cada área de homología entre las dos secuencias comparadas como alineamientos separados, BLAT las une en un alineamiento mayor.

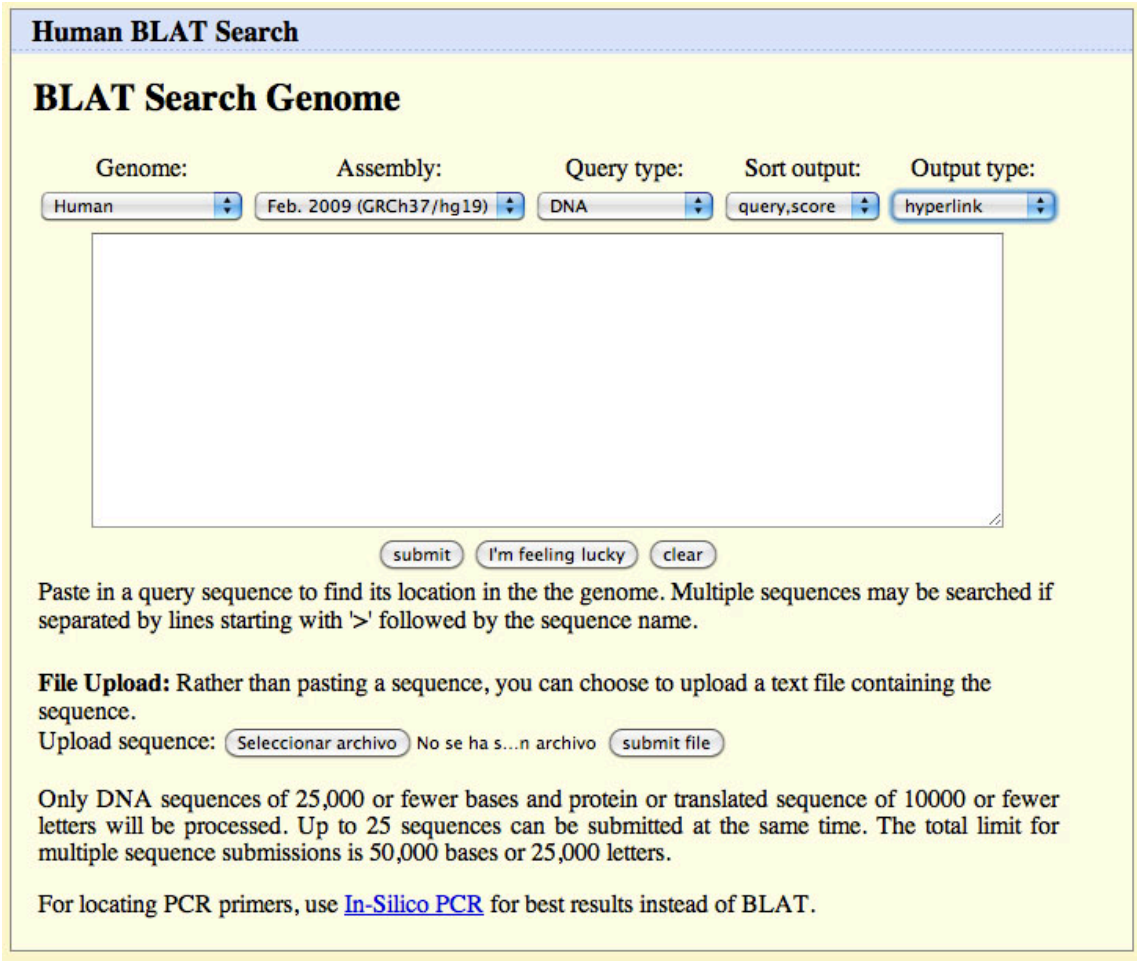

Ilustración 44 Interfaz web BLAT

Comparando BLAT con BLAST podemos encontrar las siguientes ventajas del primero con respecto al segundo:

- Mayor velocidad a costa de una menor profundidad de homología.
- Posibilidad de introducir una larga lista de consultas simultáneas en formato FASTA.
- Cinco formatos diferentes de salida
- Vínculo directo al navegador UCSC
- Detalles de alineamientos en bloques en un orden natural
- La posibilidad de lanzar el alineamiento más tarde según el usuario indique.

Normalmente BLAT se utiliza para buscar la ubicación de una secuencia en el genoma o para determinar la estructura de un exón en el RNA mensajero. BLAT puede ser accedido a través de la web o mediante su descarga, utilizándolo así de manera local. Este último modo de

acceso permite modificar ciertos parámetros manualmente que no podrían llevarse a cabo a través de la interfaz web (Ilustración 44).

Habiendo realizado el estudio sobre qué herramientas de alineamiento había en el mercado y cuáles podrían ser útiles para nuestro caso, se escogió la herramienta BLAT para encontrar las variaciones existentes en una secuencia a partir del alineamiento de ésta con su secuencia de referencia. Una de las razones por las que se eligió dicho programa, y no otro, fue la necesidad de llevar a cabo alineamientos locales en lugar de globales. No se pretendía obtener un alineamiento en longitud de la cadena, sino encontrar las similitudes y por tanto las diferencias entre las dos secuencias estudiadas. Esto es posible ya que el usuario elige de antemano la secuencia de referencia del gen caso de estudio, por lo que la similitud entre dicha secuencia y la introducida es muy alta.

En esta versión del proyecto los alineamientos que se pretenden llevar a cabo son aquellos que involucran la comparación de dos secuencias, es por ello que se ha descartado la utilización de programas de alineación múltiple. De entre los alineadores locales con posibilidad de alinear dos secuencias el más conocido y utilizado en el campo de la bioinformática es BLAST. Sin embargo, en esta versión del proyecto BLAT a sido el elegido debido a las diversas ventajas que éste, expuestas en párrafos anteriores, muestra frente a BLAST. La velocidad de escaneo que BLAT ofrece para los emparejamientos relativamente cortos, su tendencia a realizar alineamientos mayores y los diferentes formatos de salida de datos, que facilitan la extracción automatizada de información de ellos para introducirla en fases siguientes del prototipo, han sido las principales razones para su elección.

Cuando BLAST devuelve cada área homóloga de las secuencias como alineamientos separados, BLAT los une para hacer un alineamiento mayor. Esto se puede apreciar en la Tabla 3, en donde se han llevado a cabo diversas alineaciones de secuencias mediante el uso de BLAT y BLAST pudiéndose comparar los resultados ofrecidos en ambos casos. Como se puede observar BLAT tiende a unificar los alineamientos, haciéndolos más largos. Para nuestro propósito, la búsqueda de variaciones para la ayuda en el diagnóstico clínico, el resultado que ofrece BLAT es más útil. Esto es porque la probabilidad de encontrar más de una variación genómica en una secuencia de ADN es pequeña, y por tanto es preferible obtener un hueco en lugar de varios.

El formato que se muestra en la Tabla 3 es el tipo de formato raya-espacio muy usado para representar visualmente alineamientos, existen diferentes versiones de este formato donde se utilizan por ejemplo dos puntos en lugar de raya, añadiendo además el uso de un símbolo nuevo que es el de un punto. En el formato aquí comentado se muestran tres líneas con diferente información, la primera y la última línea de caracteres corresponden a las secuencias de nucleótidos que están alineándose, siendo en este caso la línea de arriba la perteneciente al estudio y la de abajo la secuencia de referencia del gen estudiado. Siempre que hay una correspondencia exacta de nucleótidos en una posición se representa mediante una línea vertical. Los espacios representan por tanto una no-correspondencia y serán debidos a una variación. Si la línea inferior contiene guiones en lugar de letras querrá decir que se ha producido una inserción en la cadena estudiada. Si por el contrario es la línea superior la que contiene los guiones, la variación encontrada corresponderá con una eliminación de nucleótidos en la secuencia de estudio con respecto a la secuencia de referencia. Se puede dar

el caso de encontrar espacios en blanco en la línea central y sin embargo no tener ningún guión en las otras dos líneas, esto se dará cuando se haya producido algún tipo de sustitución de nucleótidos o algún indel.

BLAST Query: **TAAGCCATTCTCAAGAGGCAGTCAGCCTGCAGATGTGGATCTAATGATTGACTGCCTTGT** **||||||||||||||||||||||||||||||||||||||||||||||||** Ref : **TAAGCCATTCTC---A---AGTCAGCCTGCAGATGTGGATCTAATGATTGACTGCCTTGT**
BLAT Query: **taagccattctcaagaggcagtcagcctgcagatgtggatctaatgattgactgccttgt** **||||||||||||||||||||||||||||||||||||||||||||||||** Ref: **taagccattctcaag------tcagcctgcagatgtggatctaatgattgactgccttgt**
BLAST Query: **TAAGCCATTCTCAAGAGGCAGTCAGCCTGCAGATGTGGATCTAATGATTGACTGCCTTGT** **||||||||||||||||||||||||||||||||||||||||||||||||** Ref: **TAAGCCATTCTCAAGAGGCAGTCAGCCTGCAGATGTGGATCT-A-A-TGACTGCCTTGT**
BLAT Query :**taagccattctcaagaggcagtcagcctgcagatgtggatctaatgattgactgccttgt** **||||||||||||||||||||||||||||||||||||||||||||||||** Ref: **taagccattctcaagaggcagtcagcctgcagatgtggatctaatg----actgccttgt**
BLAST Query: **TAAAGCAAGTACTTACATCAATTGGGAAGATAACTAACTCTGTCATTTTCCTACTTGTTC** **||||||||||||||||||||||||||||||||||||||||||||||||** Sbjct **TAAAGCAAGTACTTACATCAATTGGGAAG---A-TAACTCTGTCATTTTCCTACTTGTTC**
BLAT Query: **taaagcaagtacttacatcaattgggaagataactaactctgtcattttcctacttgttc** **||||||||||||||||||||||||||||||||||||||||||||||||** Sbjct: **taaagcaagtacttacatcaattgggaagataact----ctgtcattttcctacttgttc**

Tabla 3 Comparativa de alineamientos BLAT vs BLAST

BLAT posibilita la salida da datos en este formato. Sin embargo, pese que de cara al biólogo visualmente este formato es muy expresivo, en lo referente a la extracción automática de datos no es el más adecuado. Los formatos de salida que ofrece BLAT se comentan a continuación.

6.3.1. Formato AXT

Los ficheros de alineamiento axt son producidos por Blastz, una aplicación independiente del algoritmo BLAST específicamente diseñado para alinear dos secuencias genómicas largas.
A continuación (Tabla 4) se muestra un ejemplo del formato axt. En dicho ejemplo se ven los dos primeros bloques del alineamiento que se ha llevado a cabo entre una muestra humana con el cromosoma 19 del ratón.

Cada bloque de alineamiento en un fichero axt contiene tres líneas: una línea de resumen y dos líneas de secuencia. Cada bloque se separa del siguiente por líneas en blanco. La línea de resumen contiene información sobre el tamaño y las posiciones cromosómicas del alineamiento, se requieren 9 campos que se explican en la Ilustración 45.

```
0 chr19 3001012 3001075 chr11 70568380 70568443 - 3500
TCAGCTCATAAATCACCTCCTGCCACAAGCCTGGCCTGGTCCCAGGAGAGTGTCCAGGCTCAGA
TCTGTTCATAAACCACCTGCCATGACAAGCCTGGCCTGTTCCCAAGACAATGTCCAGGCTCAGA

1 chr19 3008279 3008357 chr11 70573976 70574054 - 3900
CACAATCTTCACATTGAGATCCTGAGTTGCTGATCAGAATGGAAGGCTGAGCTAAGATGAGCGACGAGGCAATGTCAC
CACAGTCTTCACATTGAGGTACCAAGTTGTGGATCAGAATGGAAAGCTAGGCTATGATGAGGGACAGTGCGCTGTCAC
```

Tabla 4 Formato axt

Ilustración 45 Campos de la línea de resumen

Las secuencias de líneas contienen el primer ensamblaje (línea 2) y el ensamblaje alineado (línea 3) con inserciones. Las repeticiones se indican con letras minúsculas.

6.3.2. Formato MAF

```
##maf version=1 scoring=humor.v4
# humor.v4 R=30 M=10 /cluster/bluearc/hg16/bed/blastz.mm3/mafNet300/chr21.mm3.maf
/cluster/bluearc/hg16/bed/blastz.rn3/mafNet300/chr21.rn3.maf
a score=9502.0
s hg16.chr21 9928623 118 + 46976097
AGCTTGTCAAGTAAGCTACCTATTTAGTGCTCGGAATGAAAGGGAGTGTGTGTTGGGAGTTGGGGGACTG----
CTTGCGTGAAACATTTCTCTCTTCTGGATTTAAAAC-TTAGTCTTGGTT s mm3.chr8 107811077 114 -
128923138 AGTTTGTCTCATAAGCCACCTGTGCACTGCCAG----
AGAAAGGGAAGTGAGCTAGCAGTGAGGGCCCAG----TGTGTGTGTGTGCTTTC-
CTTCTCTGGATTTAGAACCTTGGTTTTGCTT
s rn3.chr16 74307054 118 + 90224819 AGTTTGTCATGTCTGCTACCTGCGTTGCTAGAG----
AAAAGGGAAAGGGAGGTAGCAGTGAGGGCCTAGGAGTTTTGTTTGTTTTTTTTC-
CTTCTCTGGGTTTAAAACATTGGTTTTGCTC

a score=378.0
s hg16.chr21 9928803 10 + 46976097 AATAAATCTG
s mm3.chr8 107811442 10 - 128923138 AATCAATTA
```

Tabla 5 Formato MAF

El formato Multiple Alignement Format (MAF) almacena una serie de alineamientos múltiples en un formato fácil de analizar y de leer. Este formato almacena múltiples alineamientos a nivel de ADN entre genomas enteros. El formato .maf consiste en líneas, cada alineamiento se separa con una línea en blanco. Puede contener comentarios que son reconocidos por líneas

que comienzan por #, si la línea comienza por ## entonces la información que le sigue son metadatos.

En la Tabla 5, se muestra un ejemplo de este formato.

6.3.3. Formato WU-BLAST

Washington University BLAST (WU-BLAST) es el formato del paquete software para la identificación de genes y proteínas. El formato de salida (Ilustración 46) es parecido al que da BLAST, aunque WU-BLAST muestra las secuencias de nucleótidos en mayúsculas mientras que BLAST lo hace en minúsculas.

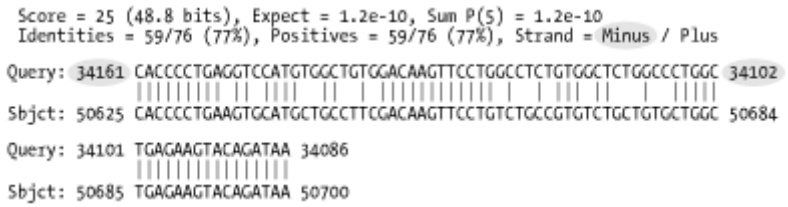

Ilustración 46 Formato WU-BLAST

6.3.4. Formato BLAST

Este formato se ha comentado en el apartado BLAST más atrás.

6.3.5. Formato PSL

En este formato la información se separa mediante tabulaciones, en la Ilustración 47 se muestran los elementos que lo componen y debajo de cada uno un ejemplo de qué podrían contener. En la Tabla 6 Ejemplos PSL se muestran otros ejemplos de este formato. Este formato ha sido el que se ha escogido para extraer la información del alineamiento.
Las columnas que forman la tabla son las siguientes:

1. Matches. Contiene el número de bases que coinciden y que no están repetidas.
2. MisMatches. Número de bases que no coinciden.
3. repMatches. Número de bases que coinciden pero son parte de las repetidas.
4. N's. Número de 'N' bases.
5. QgapCount. Número de inserciones en la secuencia que se estudia.
6. QgapBases. Número de bases insertadas en la secuencia estudiada.
7. TgapCount. Número de inserciones en la secuencia de referencia.
8. TgapBases. Número de bases insertadas en la secuencia de referencia.
9. Strand. '+' ó '-'para indicar la lectura de la hebra.
10. Qname. Nombre de la secuencia estudiada.
11. Qsize. Tamaño de la secuencia que se estudia.
12. Qstart. Posición inicial de alineamiento en la cadena consultada.
13. Qend. Posición final del alineamiento para la secuencia estudiada.
14. Tname. Nombre de la secuencia de referencia.
15. Tsize. Tamaños de la secuencia de referencia con la que se compara.
16. Tstart. Posición inicial de alineamiento en la secuencia de referencia.

17. Tend. Posición final del alineamiento para la secuencia de referencia.
18. BlockCount. Número de bloques en el alineamiento, cada bloque está exento de contener huecos.
19. Qstarts. Lista separada por comas de las posiciones de partida de cada bloque en la secuencia estudiada.
20. Tstarts. Lista separada por comas de las posiciones de inicio de cada bloque que componen la secuencia de referencia.

match	Mis-match	Rep. match	N's	Q gap count	Q gap bases	T gap count	T gap bases	strand	Q name	Q size
282701	0	0	0	1	2	0	0	+	secuencia	282703
Q start	**Q end**	**T name**	**T size**	**T start**	**T end**	**Block count**	**Block sizes**	**qStarts**	**Tstarts**	
0	282703	ref|NG_009018.1	282701	0	282701	2	68337,214364,	0,68339,	0,683,37	

Ilustración 47 Formato PSL

track name=fishBlats description="Fish BLAT" useScore=1
59 9 0 0 1 823 1 96 +- FS_CONTIG_48080_1 1955 171 1062 chr22 47748585 13073589 13073753 2 48,20, 171,1042, 34674832,34674976,
59 7 0 0 1 55 1 55 +- FS_CONTIG_26780_1 2825 2456 2577 chr22 47748585 13073626 13073747 2 21,45, 2456,2532, 34674838,34674914,
59 7 0 0 1 55 1 55 -+ FS_CONTIG_26780_1 2825 2455 2576 chr22 47748585 13073627 13073748 2 45,21, 249,349, 13073627,13073727,

Tabla 6 Ejemplos PSL

De esta información, dependiendo de en qué lugar se indica que hay huecos, se puede saber si se ha producido una inserción, una eliminación, o un indel cuyo número de nucleótidos eliminados e insertados no sea el mismo.

La fórmula de la Ilustración 48 permite extraer dos valores que determinarán el tipo de variaciones que han sido encontradas.

$$OpcQuery = \sum_{i=0..NumBloques-1(} posicionQuery[i+1] - (tamañoBloque[i]+BlancosAcumuladosQuery))$$

$$OpcRef = \sum_{i=0..NumBloques-1(} posicionRef[i+1] - (tamañoBloque[i]+BlancosAcumuladosRef))$$

Ilustración 48 Fórmulas de extracción de variaciones

Las variables posicionQuery y posicionRef hacen referencia a la lista de posiciones que se da en Qstarts y Tstarts respectivamente. La variable tamañoBloque es la lista de tamaños que tiene

cada bloque y viene dada por el valor de BlockSizes. Por último, BlancosAcumuladosQuery y BlancosAcumuladosRef hacen referencia al número de huecos el cual se va incrementando en cada pasada del bucle, es decir, si se encuentra una inserción de dos nucleótidos implica que la cadena de referencia tendrá dos huecos y por tanto un valor de dos para la variable BlancosAcumuladosRef.

Una vez encontrados estos valores, para cada valor de i, se ha de comprobar qué valore contienen las variables de OpcQuery y OpcRef.

Si tenemos que OpcQuery = 0 y OpcRef≠ 0 entonces la variación encontrada corresponderá a una deleción, es decir, a una eliminación de nucleótidos con respecto a la cadena de referencia.

Si en cambio encontramos que OpcQuery ≠ 0 y OpcRef = 0, la variación encontrada será una inserción. Que es lo mismo que decir que se ha encontrado una inserción de nucleótidos en la secuencia estudiada con respecto a la cadena de referencia.

En el caso de que lo que se encuentre es que OpcQuery ≠0 y OpcRef ≠ 0, entonces la variación hallada es un indel de tamaño diferente, es decir, se habrá producido una eliminación de 'x' nucleótidos y una inserción de 'y' nucleótidos en la secuencia de estudio con respecto a la secuencia de referencia.

Si nos encontramos con algún valor en el campo MisMatches lo que se indica es que ha habido un indel cuyo número de nucleótidos eliminados e insertados son iguales.

La posición de cada cambio se obtiene de los campos Qstarts y Tstarts.

Por tanto se demuestra que la extracción de información a partir de este formato es mucho más sencilla en comparación con el resto de formatos disponibles que se han comentado en los apartados anteriores. Sin embargo, este formato posee una limitación a la hora de enfrentarse a variaciones en las que se ha producido, en un lugar determinado de la secuencia, una eliminación y una inserción de una misma cantidad de nucleótidos. Si observamos bien la información de la que se dispone, tan sólo se muestra la existencia de este tipo de cambio sin dar más información al respecto. Existe la posibilidad de dejar este apartado aislado por el momento y de solucionarlo en versiones posteriores al prototipo que se está diseñando en este proyecto y dejarlo así documentado e indicado en el informe genético resultante. La otra opción es la de analizar si este tipo de cambio se ha producido en el estudio de la secuencia que se esté llevando a cabo. De no ser así, entonces se extraería la información tal y como se ha comentado anteriormente, ofreciendo una mayor rapidez de extracción de información. Si por el contrario ese tipo de cambio se hubiera producido, se podría extraer la información del resto de variaciones del mismo modo que se ha hecho hasta ahora y después lanzar de nuevo el programa BLAT pero obteniendo los resultados en el formato tradicional BLAST. De este modo se recorrería el fichero buscando este tipo de variaciones. Dado que este tipo de variaciones suelen ser SNPs nos encontramos con cambios muy pequeños, normalmente de un nucleótido, y es por ello que ya que el formato BLAT devuelve los alineamientos por bloques se deberían leer tan sólo aquellos bloques cuyos inicios en la secuencia estudiada y la de referencia fueran el mismo o muy cercanos tal y como se muestra en la Ilustración 49.

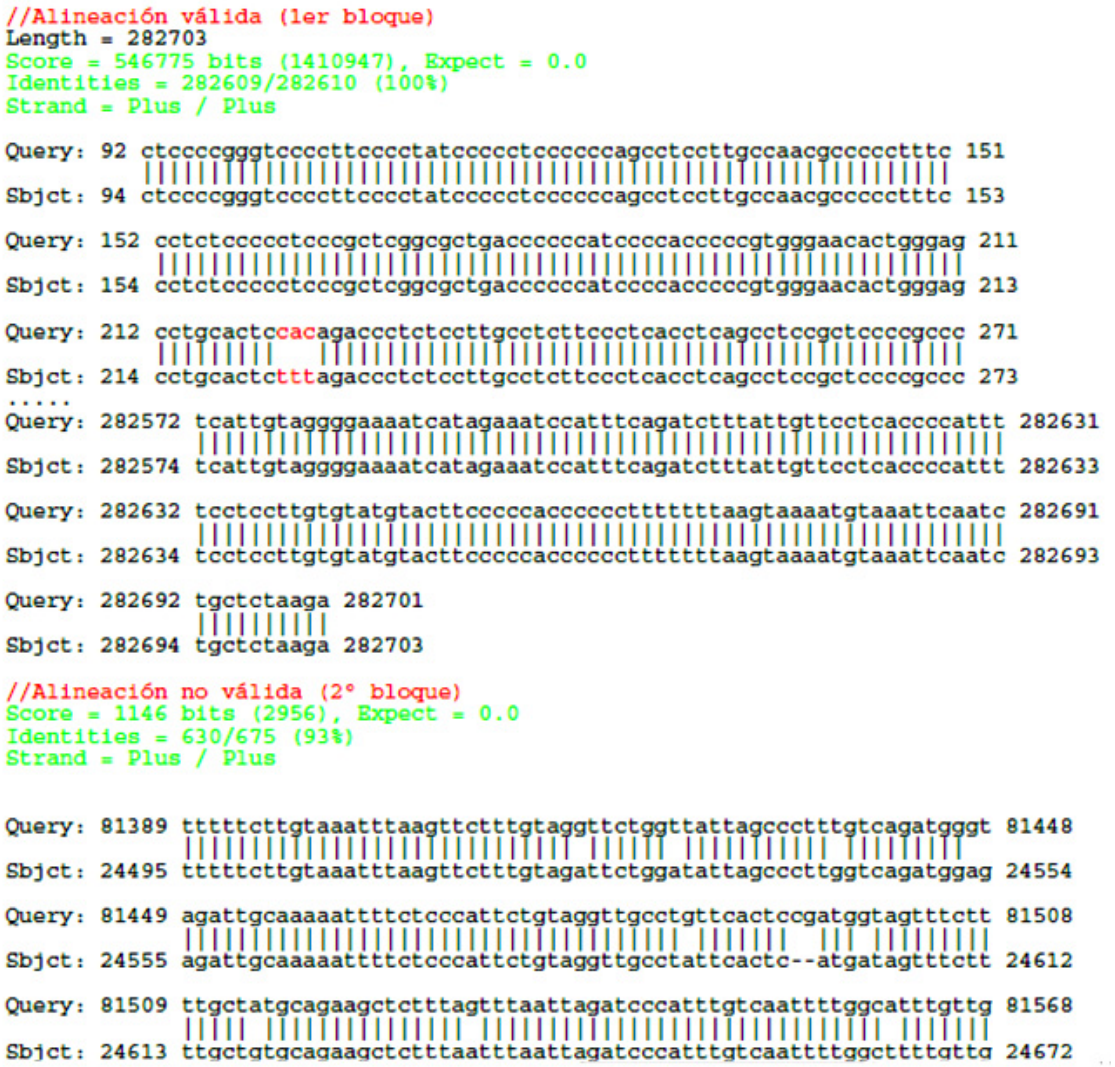

Ilustración 49 Descarte de bloques de lectura

En este proyecto se eligió realizar la segunda opción expuesta en el párrafo anterior, con el fin de poder facilitar el mayor número de variaciones posibles en el informe génico.

6.4. Lecciones aprendidas

El estudio del dominio en el que se trabaja es imprescindible. Gracias a esto se ha aprendido que dependiendo de lo que se desea analizar en una secuencia se deben utilizar unas herramientas de alineamiento u otras, no sirve cualquier herramienta de alineamiento escogida al azar. Una vez escogida la técnica, la elección de la herramienta en particular es un apartado importante y se deben sopesar los pros y los contras consecuentes de la elección de cada herramienta.

La interacción con especialistas en el campo es indispensable tanto a la hora de obtener una guía cuando se inicie el estudio del domino, como a la hora de adquirir los requisitos que necesita el usuario y no llevar a cabo un trabajo a ciegas sin tener en cuenta la visión del biólogo, lo que éste necesitará como resultado de nuestro trabajo. Por tanto, tener en mente al usuario final es un punto inamovible. Es por ello que en este capítulo de este trabajo se ha

obtenido la forma de facilitar y minimizar el trabajo que el usuario debe realizar para obtener las variaciones que más tarde deberá analizar.

De este capítulo se puede extraer que ninguna de las herramientas de alineamiento estudiadas aquí está pensada para obtener automáticamente y de forma sencilla la información que posibilitan. Como excepción el formato psl de BLAT ofrece de una forma más estructurada su información, aunque no de manera completa. Todas estas herramientas están pensadas para facilitar al biólogo los alineamientos realizados de forma visual y para que después éste sea el que realice el trabajo de limpiar esta información y extraer las posiciones de las variaciones y los cambios originados. En ningún momento se han diseñado con el fin de facilitar este apartado al biólogo, o con una idea de poder proporcionar ambas partes: información visual y datos concretos.

Actualmente no existe alguna herramienta libre que ofrezca al mismo tiempo una forma visual de ver los alineamientos generados y a su vez una manera estructurada de representar la información que de dichos alineamientos se extrae. Esto es debido a que dichas herramientas han sido implementadas para resolver unos problemas puntuales y no han sido diseñadas bajo un punto de vista de un sistema de información, en el que la interacción con el usuario final es imprescindible a la hora de saber qué es lo que el usuario espera poder hacer con lo que se le ofrece y cuál es la forma más adecuada de ofrecérselo y facilitárselo.

Búsqueda de variaciones

Una variación es un cambio que se produce en una secuencia de ADN como resultado de la inserción, eliminación o sustitución de uno o más nucleótidos en dicha cadena. Las variaciones, desde el punto de vista del efecto que conllevan, pueden ser tanto perjudiciales, polimorfismos neutros como de efecto desconocido. Las variaciones perjudiciales son aquellos cambios que tienen como consecuencia un efecto perjudicial en el organismo, por ejemplo el cambio de un nucleótido que produce un aminoácido diferente teniendo como consecuencia la pérdida de la funcionalidad que tenía la proteína de la cual formaba parte. A este tipo de cambios se les denomina mutaciones. Por otra parte, los polimorfismos neutros son aquellos cambios que no producen alteración alguna. Esto es posible debido a que las secuencias genómicas están formadas por tripletas de nucleótidos que pueden ser sólo de cuatro tipos: guanina (G), citosina ©, timina (T) y adenina (A), y por tanto la variabilidad de los elementos no es mucha. Por ejemplo, una variación en el último nucleótido del aminoácido de la alanina, GCT, nunca generará un cambio visible ya que cualquier combinación en esa última posición tendrá como resultado ese mismo aminoácido (tanto GCT, GCA, GCC cómo GCG se traducen como adenina) y por tanto no tendrá un efecto apreciable, el cambio es silencioso. El resto de variaciones se clasifican dentro del grupo de variaciones con efecto desconocido. En este grupo se englobarían todas aquellas variaciones cuya consecuencia aún no se conoce, dentro de éstas estarían las variaciones que se producen en el genoma a causa de la adaptación de las especies al medio con el que interactúan. Estas variaciones son aquellas que, gracias a la selección natural, permanecen en el organismo haciendo que las especies evolucionen.

Las variaciones pueden producirse tanto en zonas codificantes de las secuencias de ADN como fuera de éstas. Si clasificamos las variaciones por su posición tendríamos dos grupos, (1) cromosómicas, cuando la variación afecta a partes del genoma o (2) génicas, cuando la variación afecta a un solo gen.

Otra forma de clasificar las variaciones encontradas sería a través de los detalles que tienen, como son la posición exacta y el cambio ocurrido. Si dichos detalles son desconocidos para una variación, ésta será catalogada en el grupo de las variaciones imprecisas. Sin embargo, todas aquellas variaciones cuya posición y cambio sean conocidos se agruparán en la sección de variaciones precisas. Este tipo de variaciones pueden diferenciarse según su tipología en cuatro grupos: (1) Inserciones, se da cuando uno o varios nucleótidos se insertan en la secuencia de referencia, (2) borrados o deleciones, surgen de la eliminación de uno o varios nucleótidos con respecto a la cadena de ADN original, (3) indels, que son consecuencia de la unión de una inserción y una deleción en una misma posición determinada y (4) inversiones, que se produce cuando una sección de ADN se invierte. Las inversiones pequeñas pueden afectar a unas pocas bases de un gen, mientras que las inversiones más grandes afectan a grandes regiones de cromosomas que contienen varios genes.

Todas estas posibles clasificaciones y la descripción de variación quedan especificadas en la parte del esquema conceptual CSHG que se puede ver en la Ilustración 50.

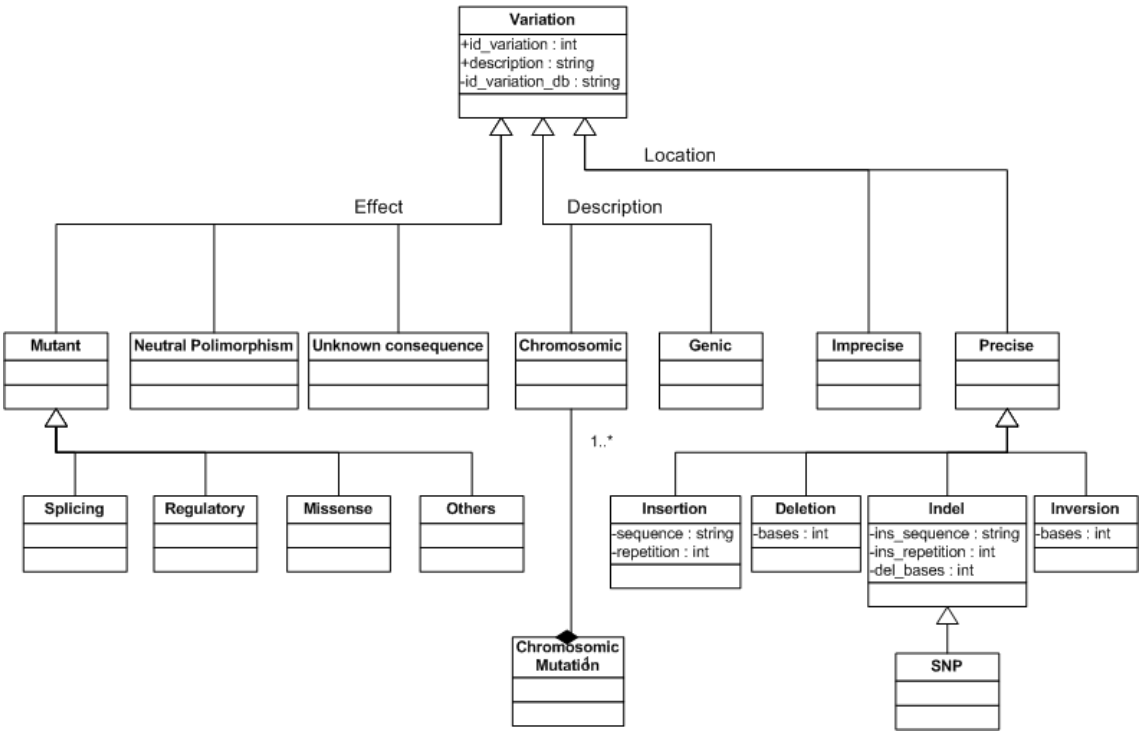

Ilustración 50 Esquema conceptual - Variaciones

Por tanto, una variación queda determinada por su efecto, localización y descripción. Retornando al caso de estudio que nos concierne en este trabajo, la generación de un diagnóstico genético, es necesario ofrecer una búsqueda de variaciones eficaz y guiada por el dominio. Gracias al esquema conceptual, que remarca los conceptos que especifican una variación, la generación de una base de datos respetuosa con el dominio se ve simplificada a una traducción de este en un esquema entidad relación. La extracción de la información sobre variaciones de este repositorio resulta sencilla y rápida.

En el diagnóstico genético se tiene una posición y un cambio, y lo que se quiere es la obtención de una variación ya estudiada que concuerde con esos datos. En términos de posición de la variación que se busca es una variación precisa, puesto que se conocen sus atributos concretos, y en lo referente a la descripción buscaremos una variación genética, dado que hablamos de un diagnóstico génico. Por tanto, mediante el uso de estos parámetros podremos obtener fácilmente qué tipo de efecto tiene dicha variación. El hecho de encontrar una variación que case con estos parámetros significará que el cambio ya ha sido estudiado con anterioridad y por tanto ya está catalogado. Este proceso puede verse claramente en la Ilustración 51 en donde la parte del modelo que casa con la información introducida en dicho ejemplo se muestra en un color rosado, y el efecto asociado a la variación buscada se colorea en verde. En este caso se busca un indel que se ha producido en la posición 132 del gen, en el que se ha cambiado una adenina por una guanina resultando en una mutación con consecuencia patógena.

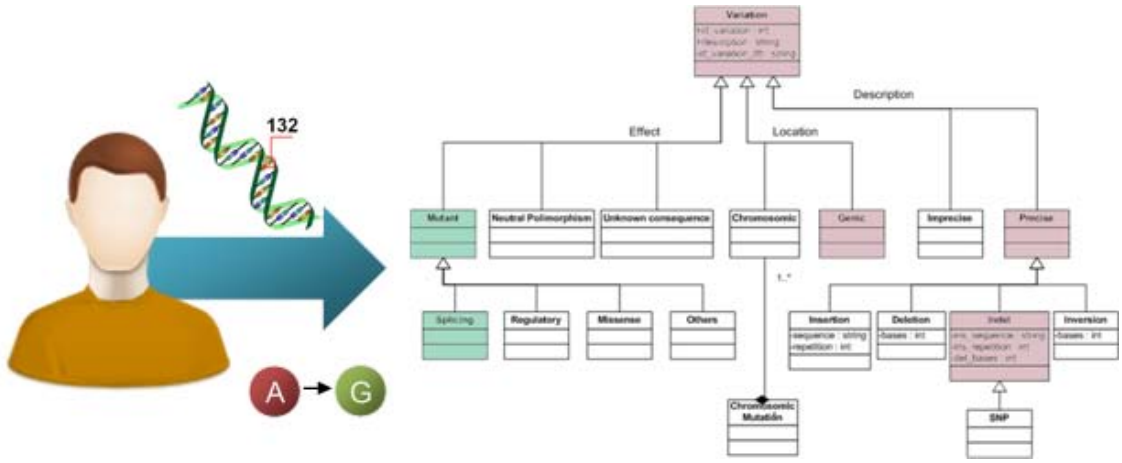

Ilustración 51 Búsqueda de variaciones guiada por el modelo

7.1. Variaciones sinónimas

A la hora de buscar una variación, los métodos que se emplean para extraer la información referente al cambio son varios. Como vimos en el apartado de alineamiento existen diversas herramientas que ayudan en la extracción de la información necesaria para tratar una variación: posicionamiento y cambio.

Cada científico utiliza su técnica favorita o la que cree es más conveniente para obtener estos datos, pero no existe un estándar a la hora de llevar a cabo esta extracción de información. La falta de un estándar tiene como resultado una especie de subjetividad a la hora de trabajar, en el sentido de libertad a la hora de elegir una herramienta para realizar el alineamiento de secuencias de ADN, libertad en lo referente a la especificación de los cambios encontrados como veremos en capítulos siguientes, etc.

En este sentido ya vimos un ejemplo en el capítulo más atrás, concretamente el capítulo sobre el alineamiento. En éste se ofrecía un ejemplo en el que una misma secuencia de ADN alineada, con respecto a una misma secuencia de referencia, utilizando dos herramientas distintas ofrecía resultados diferentes (Tabla 3). Y es que cada herramienta de alineación sigue un determinado algoritmo y cada uno de éstos se guía por una técnica diferente. Esto se ve agravado debido a la redundancia genética existente en el genoma humano, y es que una variabilidad de cuatro bases es insuficiente si se pretende no tener algún tipo de repetición o alguna combinación azarosa que termine dando como resultado otro resultado distinto al esperado.

Veámoslo mediante un ejemplo (Ilustración 52) de forma más clara. Supongamos una secuencia de ADN en la cual se ha insertado un nuevo conjunto de nucleótidos AGGTAT en la posición 235. Sin embargo, al realizar el alineamiento nos encontramos con una inserción distinta, en este caso de la cadena TAGGTA en la posición 234. ¿Es esto correcto? ¿A qué se debe este imprevisto? Contestar a este caso, en este ejemplo concreto, es bastante sencillo y resulta de una combinación de los dos factores que comentamos en el apartado anterior.

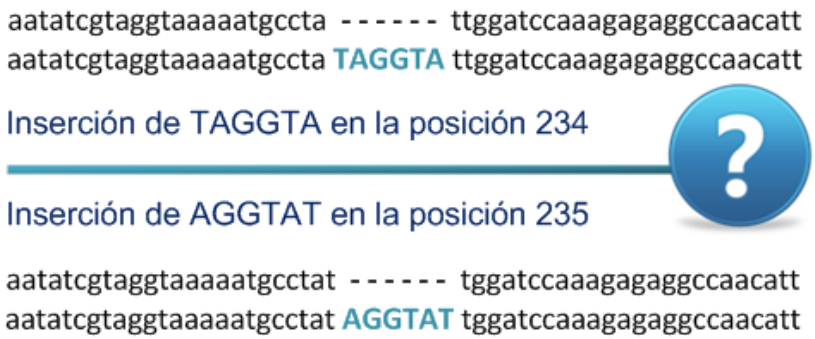

Ilustración 52 Variaciones sinónimas

Si observamos la imagen podemos comprobar que ambos alineamientos son acertados, por tanto la respuesta a nuestra primera pregunta sería "sí, es correcto". Simplemente, en este caso, se ha llevado un algoritmo de alineamiento distinto. En el primer alineamiento se ha realizado una correspondencia uno a uno de derecha a izquierda, y en el segundo alineamiento se ha llevado el mismo tipo de alineamiento pero de izquierda a derecha. No obstante, el hecho de que se usen diferentes técnicas de alineamiento no es condición suficiente para que esto se dé. En este ejemplo se produce esta variabilidad en la respuesta esperada debido a que la cadena insertada contiene como primer nucleótido una timina, y en la posición siguiente a la que se inserta también hay una timina. Como consecuencia la posición del alineamiento puede variar según la dirección con la que se alineen las secuencias, dando por tanto en cada caso una secuencia insertada diferente. Esto es debido, como se comentó anteriormente, a que sólo existen cuatro posibles valores a insertar y por tanto la probabilidad que surja un caso como el anterior no es nada despreciable.

En este ejemplo el porqué de la diferencia encontrada es bastante simple y se puede deducir fácilmente observando ambos ejemplos, sin embargo hay casos en que la justificación no es tan evidente y es resultado de varias combinaciones o de repeticiones de grupos de nucleótidos.

Todo esto provoca que a la hora de documentar variaciones, los biólogos introduzcan tanto la posición como el cambio que ellos observen según el resultado que su estudio específico, dado por una alineación llevada a cabo con una herramienta específica, les haya dado. Dos estudios de una misma muestra que utilicen dos técnicas distintas, para encontrar los cambios en la secuencia de ADN estudiada, podrán dar dos resultados distintos. Y por tanto dichos estudios serán documentados de manera diferente, como dos cambios distintos habiendo sido en realidad producidos por el mismo cambio en la secuencia génica.

Mirándolo desde el punto de vista de las variaciones y no del alineamiento podríamos decir que dos cambios diferentes, como los cambios resultantes vistos en el ejemplo anterior, pueden generar una misma secuencia de ADN resultante. Esto es lo que denominamos variaciones sinónimas.

Las variaciones sinónimas conllevan un problema grande en lo que se refiere a la búsqueda de variaciones. Como se observaba en el modelo de la Ilustración 50, se precisa de una posición concreta para buscar una variación. Sin embargo, debido al problema de las variaciones sinónimas podemos encontrarnos con que la posición que la herramienta de alineamiento utilizada nos proporciona como resultado no corresponda con la posición que ha sido

documentada, pese a que la secuencia de ADN resultante sea la misma. Como consecuencia el sistema no encontraría ninguna variación que se ajustase a los datos proporcionados, mientras que en realidad sí existiría una variación que casaría con los resultados. Por tanto, aunque conceptualmente una variación tenga una posición concreta, en la práctica este dato debe tratarse teniendo en cuenta la existencia de variaciones sinónimas que pueden ser omitidas durante el proceso de búsqueda.

7.1.2. Solución planteada

Se trata pues de estudiar los casos para los que el algoritmo de alineamiento da una posición que no se encuentra dentro del sistema de información, y sin embargo otra posible alineación daría otra posición que llevaría a la obtención de un resultado positivo en la búsqueda de información en el GeIS.

El caso en cuestión sería buscar en la base de datos una única variación, teniendo la siguiente información extraída del alineamiento: el tipo de alteración producida (inserción, borrado o indel), la posición donde se produce (P) y la secuencia alterada (SA). Se conocen tanto la secuencia de referencia (SR) como la secuencia que se compara, también llamada secuencia muestra (SM).

A continuación se realiza un análisis por casos de menos a mayor grado de dificultad, separándolos por tipo de alteración y longitud de la cadena alterada.

Borrado de un nucleótido

- La secuencia borrada, SA, es igual a un nucleótido, N.
- La posición del borrado en SR es P.

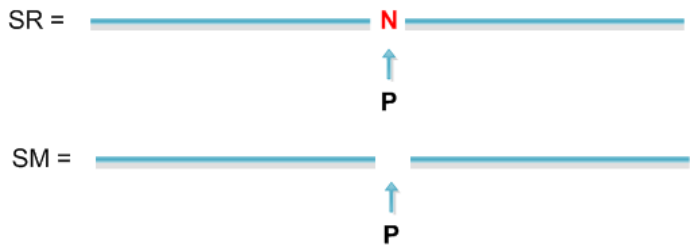

Ilustración 53 Borrado de un nucleótido

Solución:

1. Buscar N en la posición P en SR.
2. Determinar las posiciones a derecha e izquierda en las que N se va repitiendo a partir de P en SR, obteniendo la posición de la derecha (PD) y la de la izquierda (PI).

Ilustración 54 Posiciones derecha e izquierda a la repetición

3. Buscar en la base de datos una variación precisa que sea una eliminación de un nucleótido N con una posición P' tal que PI≤P'≤PD.

Inserción de un nucleótido

- La secuencia insertada SA es igual a un único nucleótido N.
- La posición de la inserción en SR es P.

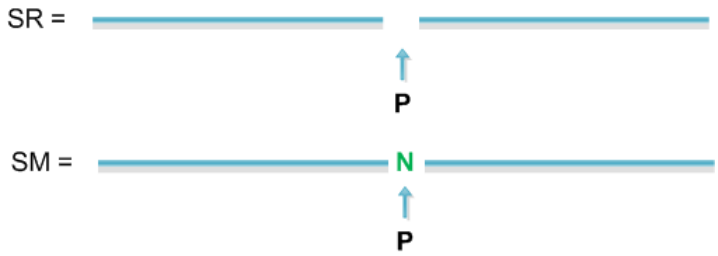

Ilustración 55 Inserción de un nucleótido

Solución:

1. Ubicarse en la posición P en SR.
2. Determinar las posiciones a derecha e izquierda en las que N se va repitiendo a partir de P en SR, obteniendo la posición a derecha (PD) y a izquierda (PI).

Ilustración 56 Posiciones a derecha e izquierda de una inserción

3. Buscar en la base de datos una variación precisa que sea una inserción de un nucleótido N, con una posición P' tal que PI≤P'≤PD.

Indel de un nucleótido

- La secuencia modificada SA es igual a un único nucleótido N y el nucleótido por el que se sustituye es un único nucleótido M.
- La posición de la inserción en SR es P.

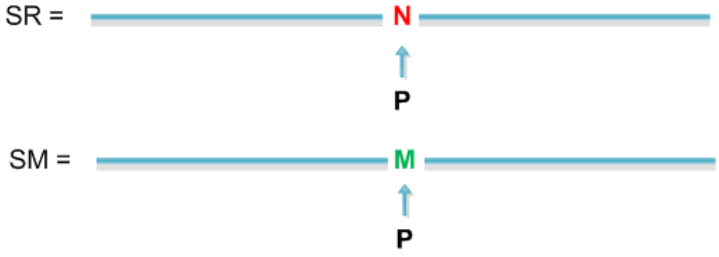

Ilustración 57 Indel de un nucleótido

Solución:

1. En este caso no hay que hacer nada, no existe posibilidad alguna de que haya variaciones equivalentes por repetición, ya que o la cadena SR tiene repeticiones de N, y por tanto SM no tiene de M en la misma posición, o al contrario.
2. Buscar en la base de datos una variación precisa, de tipo indel, de un nucleótido N por otro M en la posición P.

Borrado de varios nucleótidos

- La secuencia borrada SA es $N_1,\ldots,N_i$, donde i>1.
- La posición del borrado en SR es P.

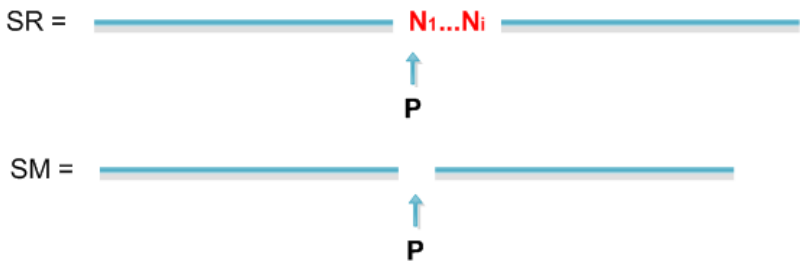

Ilustración 58 Eliminación de varios nucleótidos

Solución:

1. Buscar SA en la posición P en SR.
2. Determinar la posición a la derecha en la que se va repitiendo SA parcial o totalmente (una o varias veces) a partir de P+i en SR, obteniendo la posición a la derecha PD.

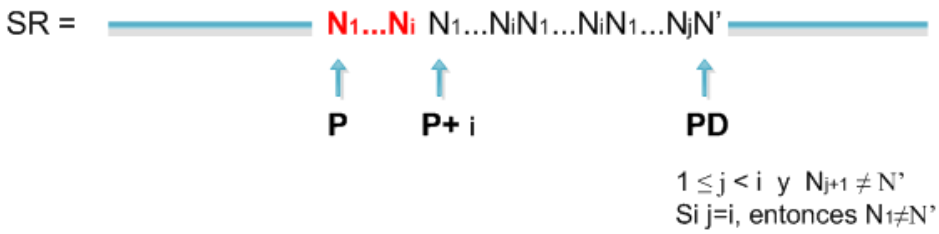

Ilustración 59 Obtención de PD en el caso del borrado

3. Determinar la posición izquierda en la que se va repitiendo SA parcial o totalmente (una o varias veces) a partir de P en SR, obteniendo la posición de PI.

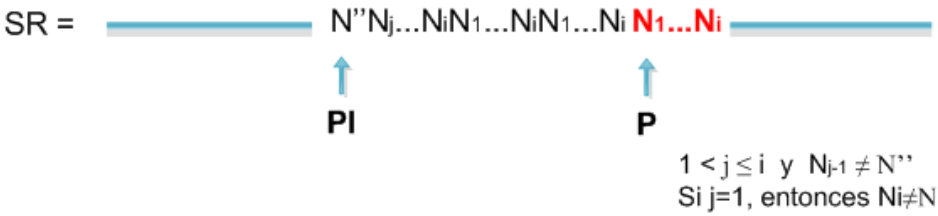

Ilustración 60 Obtención de PI en el caso del borrado

4. Buscar en la base de datos una variación precisa de en la que se haya producido una eliminación de la cadena SA con una posición P' tal que PI≤P'≤PD.
5. Aplicar el borrado de SA sobre SR en P' y comprobar que la cadena SM' es igual a SM.

Inserción de varios nucleótidos

- La secuencia insertada SA es $N_1,\ldots,N_i$, donde i>1
- La posición de la inserción en SR es P.

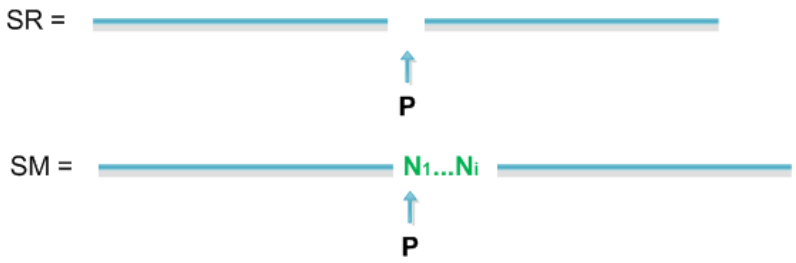

Ilustración 61 Inserción de varios nucleótidos

Solución:

1. Ubicarse en la posición P en SR.
2. Determinar la posición a la derecha en la que se va repitiendo SA parcial o totalmente (una o varias veces) a partir de P en SR, obteniendo la posición a derecha PD.

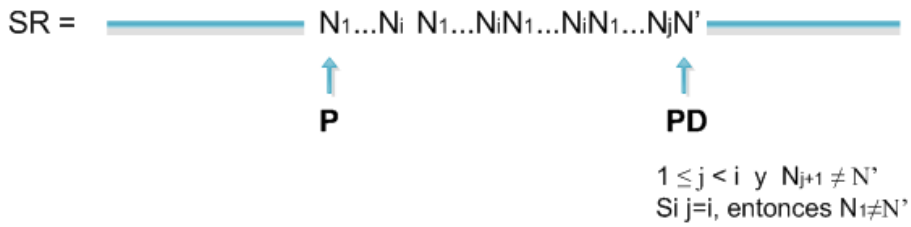

Ilustración 62 Determinar posición derecha en el caso de la inserción

3. Determinar la posición a la izquierda en la que se va repitiendo SA parcial o totalmente (una o varias veces) a partir de P en SR, obteniendo la posición a derecha PI.

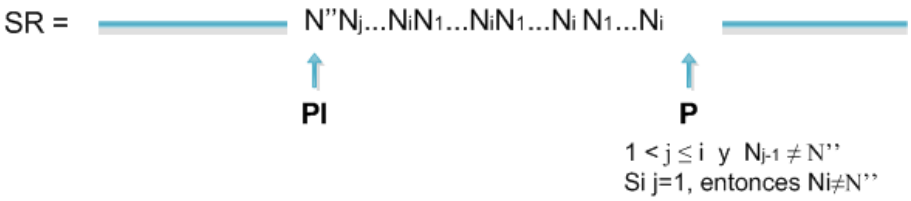

Ilustración 63 Determinar posición izquierda en el caso de la inserción

4. Buscar en la base de datos una variación precisa que inserte una secuencia SA con una posición P' tal que PI≤P'≤PD.
5. Aplicar la inserción de SA sobre SR en P' y comprobar que la cadena SM' es igual a SM.

Indel de Varios Nucleótidos:

- La secuencia modificada SA es N1, …, Ni, donde i>1 y la secuencia sustituta SS es N'1, …, N'i.
- La posición de la inserción en SR es P.

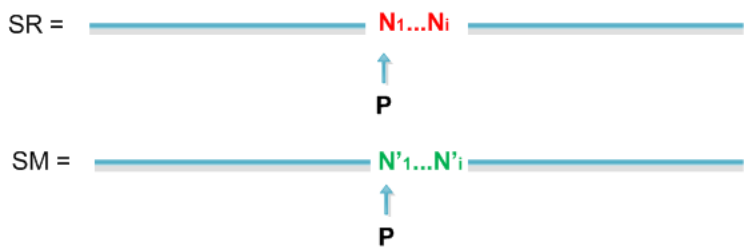

Ilustración 64 Indel de varios nucleótidos

Solución:

1. No hay que hacer nada, no hay posibilidad de que existan variaciones equivalentes por repetición, ya que o la cadena SR tiene repeticiones (totales o una parcial) de SA (por tanto SM no tiene de SS en la misma posición) o al contrario.
2. Buscar en la base de datos una variación precisa que sea un indel en el cual se haya eliminado la secuencia SA en la posición P y se haya insertado, en esa misma posición, la secuencia SS.

7.2. Relación genotipo - fenotipo

Una de las partes del estudio de la genética que más llama la atención es la que corresponde a la relación gen-fenotipo. Un gen se expresa en un individuo de un determinado modo, por ejemplo el color azul de los ojos. El hecho de que las personas seamos diferentes unas de otras viene dado por las variaciones que se encuentran en el ADN de cada ser humano. Por tanto las variaciones que veíamos antes se traducen en un resultado que a veces puede ser perceptible al ojo humano, como en el caso del color del pelo, o que a veces no es tan obvio, como por ejemplo la predisposición a sufrir un cáncer.
Esta es la razón de la importancia de los estudios genéticos, el poder valorar qué consecuencias puede tener una variación en un ser humano. Es por ello que dentro del esquema conceptual CSGH existe una parte que modela la parte fenotípica (Ilustración 65) relacionándola con la parte de variaciones. De este modo es posible poder introducir dentro del informe genético el resultado que producen los cambios que se estudian.

La entidad Variation se relaciona con la entidad Syndrome, que es la que define las enfermedades, como por ejemplo la diabetes o la neurofibromatosis. Una enfermedad está caracterizada por diversos signos y síntomas que están definidos en la clase Feature. La entidad Value indica el valor que poseen dichas características, en el caso de la nurofibromatosis tendríamos como característica "manchas", y como valor "muchas". Sin embargo, para otros valores que sean mesurables se necesita de algo más, es por ello que

existe la especialización Measurable que cuelga de Value. Esto sucede por ejemplo con la característica glucemia relacionada con el síndrome de la diabetes. En este caso la glucemia se mide en mg/dl. La entidad Measurable no indica sólo las medidas en las que se miden las características, sino también el método que se ha utilizado para llevar a cabo dichas medidas. Por último, para que una variación esté asociada a un fenotipo es necesario tener una publicación científica como prueba de confianza. Es por ello que a esta vista del modelo se le añade la clase Bibliography Reference, en la que se describe la publicación en la que fue encontrada dicha asociación. A su vez la entidad Bibliography DB indica en qué base de datos se encontró esa publicación.

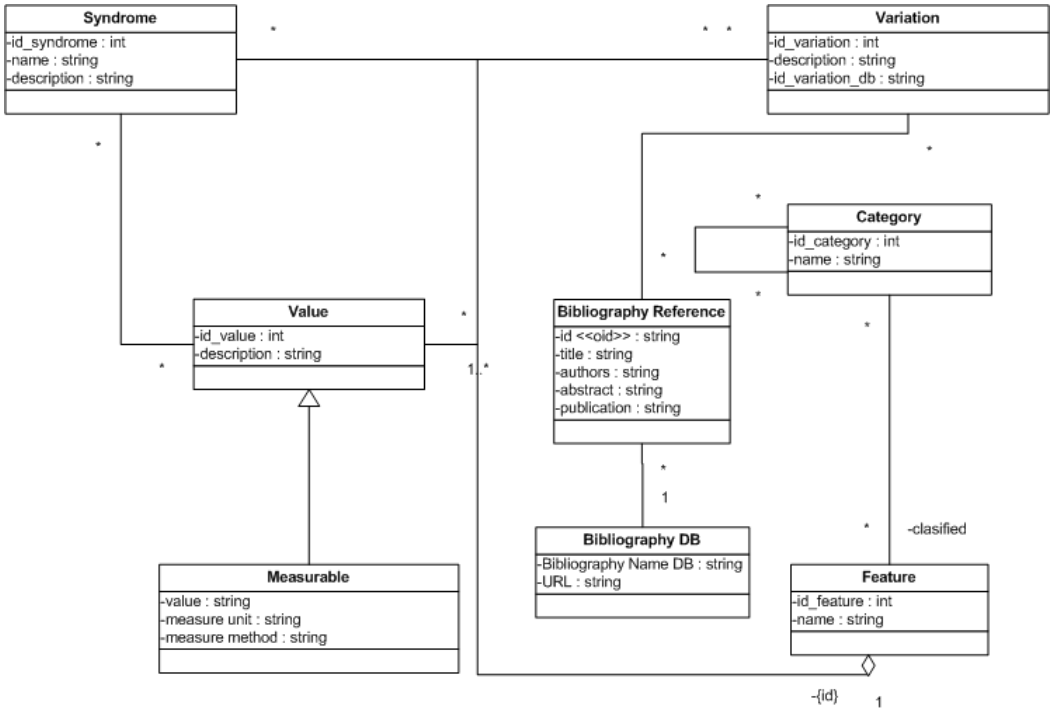

Ilustración 65 Esquema Conceptual de la parte de Fenotipo

7.3. Lecciones aprendidas

La incorporación de técnicas de sistemas de información, como el modelado conceptual, al dominio de la genómica ha tenido dos contribuciones fuertes en el campo del diagnóstico genético. Estos dos puntos tratan los puntos de (1) la búsqueda de variaciones y (2) la relación fenotípica que poseen éstas.

Hasta ahora los biólogos podían pasarse horas delante del ordenador, buscando en diversas fuentes de datos distribuidas por internet, hasta tener la suerte de encontrar la variación que estaban estudiando. El estudio del dominio y su integración en el mundo de los Sistemas de Información posibilitan que dicha búsqueda se reduzca a temporalidades de minutos. El modelado del caso de estudio permite saber cuáles son las entidades que definen el dominio y cómo se interrelacionan, permitiendo así facilitar la búsqueda de información y obtener resultados en un tiempo mucho menor.

Lo mismo sucede a la hora de relacionar una variación genotípica con su resultado fenotípico. No todas las bases de datos disponibles incluyen una bibliografía asociada a una variación indicando si ésta es patógena o no y cómo afecta ésta al organismo en el que se produce. Del mismo modo, no todas las bases de datos que ofrecen información de variaciones proporcionan el fenotipo que está asociado a éstas, y muchas de las que lo hacen lo llevan a cabo de manera escueta y muy simplificada. Añadiendo esta parte fenotípica al modelo, tal y como se muestra en la Ilustración 65, se mejora esta forma de interactuar, pudiendo ofrecer a los usuarios de estos GeIS genotipos y fenotipos asociados de forma fácil y rápida. Esto es una gran ventaja, sobre todo de cara al estudio de los diagnósticos genéticos en los que la velocidad con la que se puedan ofrecer a un médico o paciente a veces es determinante a la hora de empezar a tratar una enfermedad o incluso eliminarla.

Durante este estudio se ha definido el concepto de variaciones sinónimas como todas aquellas que, pese a ser cambios inicialmente diferentes, producen un mismo resultado. El hecho de la existencia de este tipo de cambios dificulta la búsqueda de variaciones, pues la variación obtenida en cada estudio es dependiente del tipo de alineamiento que se realice. Pudiendo así dar una variación diferente al cambiar el método de alineamiento empleado sin que variase la secuencia de ADN resultante, con respecto a la obtenida en el otro alineamiento. En esta tesis se ofrece una solución a la hora de tratar dicho tipo de variaciones, de modo que su búsqueda en el sistema de información sea satisfactoria en cualquier caso. Se deja abierta la cuestión de si otra forma de referenciar variaciones sería más correcta, dado que el posicionamiento absoluto no es válido en estos casos.

Con respecto al algoritmo presentado, como ventaja se ofrece que siempre se va a encontrar la variación dentro del Sistema de Información, siempre y cuando existe una variación que case con el cambio encontrado en la fase de alineamiento. Sin embargo, este proceso tiene como desventaja un incremento en el tiempo de búsqueda, así como una mayor carga de ejecución debido a que es necesario realizar una fase previa para obtener los rangos de posiciones de las variaciones sinónimas.

Notación de variaciones

Durante esta tesis de máster se ha hablado de la heterogeneidad en la información existente en el campo de la genómica, surgida en cierto modo por el gran impacto que han tenido descubrimientos tales como la estructura del ADN a modo de una "doble hélice" [40] ó la obtención de la primera secuencia consenso de ADN humano completa [41] y también por el abaratamiento y la mejora de las técnicas de secuenciación. Esto ha llevado a un incremento exponencial en la generación de datos bioinformáticos. Cada instituto de investigación genera resultados y los clasifica según sus necesidades y sus estándares, guardándolos en sus propios repositorios de información. Ejemplos muy conocidos de estos almacenes de datos son OMIM [42], o Human Genome Mutation DataBase (HGMD) [43]. Muchos de estos repositorios son de acceso libre, lo cual produce una gran oferta de información pero sin un estándar asociado para ser interpretada o accedida.

Centrándonos en el campo de las variaciones, el hecho de que cada instituto utilice una forma diferente para describirlas incrementa la heterogeneidad de la que hablábamos antes y complica tanto la labor de los biólogos como la de los ingenieros del software. Por ejemplo, la búsqueda y extracción de datos de diferentes fuentes es complicada. Entre las formas utilizadas para describir variaciones se encuentran:

- El cambio del aminoácido que había sido deducido a partir de la sustitución de nucleótidos estudiada.
- La posición con respecto al gen.
- La posición con respecto a la cadena de DNA codificante.
- Un sobrenombre basado en alguna restricción de un sitio o incluso en el nombre del paciente que se estaba estudiando.

Con tal variedad de nomenclaturas saber, por ejemplo, si una variación ya ha sido estudiada se convierte en una ardua tarea. Es por eso que en 1993 comenzaron a plantearse una serie diferente de artículos [10, 11] en lo que se clamaba por una nomenclatura única, no ambigua. Las sugerencias presentadas fueron ampliamente discutidas, modificadas y ampliadas, convirtiéndose en las recomendaciones actuales ofrecidas por HGVS, que han sido aceptadas en gran medida y aplicadas en todo el mundo.

8.1. HGVS

Human Genome Variation Society (HGVS) es una sociedad cuyo objetivo es el de fomentar el descubrimiento y la caracterización de variaciones genómicas. Su intención es la de categorizar e identificar los cambios del genoma humano que puedan dar lugar a algún tipo de enfermedad, para poder ayudar así a la mejora de la salud humana. Para llevar a cabo esto es necesario cotejar la información genómica necesaria para el diagnóstico molecular, la investigación sobre los mecanismos básicos y el diseño de los tratamientos de las enfermedades humanas.
Con este fin, HGVS ha realizado una serie de recomendaciones tanto para la nomenclatura de las variaciones, como para el contenido de las bases de datos de mutaciones. Estas recomendaciones pretenden servir para describir variaciones en secuencias de una manera

estable, significativa e inequívoca. Intentando por tanto poner fin a la heterogeneidad en la descripción de las variaciones que existe en la actualidad, tanto en documentos científicos como en las diferentes bases de datos.
Sin embargo, dichas recomendaciones sufren ligeras modificaciones de vez en cuando con el propósito de solventar pequeñas inconsistencias y/o clarificar convenciones que puedan resultar confusas. Es por ello que HGVS proporciona una lista de versionados para que de este modo los usuarios de dicha nomenclatura puedan indicar hasta qué grado siguen la última versión de las recomendaciones dadas.
El propósito final de HGVS por tanto es el de mediante el uso de una guía de recomendaciones, poder describir cualquier tipo de variación en una secuencia y ayudar, de este modo, a conseguir la instauración de un estándar reconocido y aceptado.

8.1.2. Recomendaciones generales

A la hora de describir una variación, ésta debe ser descrita en el nivel más básico, por ejemplo a nivel de ADN. Las descripciones a su vez han de ir ligadas siempre a una secuencia de referencia, tanto si es una secuencia de referencia genómica o de ADN codificante. A efectos prácticos la secuencia de ADN codificante es más utilizada ya que es más fácil extraer las posiciones con respecto a un gen, pues los índices que se usan a niveles genómicos son muy grandes. Sin embargo, en términos teóricos la secuencia de referencia genómica es mejor ya que incorpora toda la información disponible, incluida la propia cadena de ADN codificante. Según las recomendaciones de HGVS, a la hora de reflejar en un documento varias variaciones, éstas deben darse de una manera listada ofreciendo los resultados de manera resumida mediante el uso de columnas separadas en las que se indique el cambio en ADN, ARN y en la proteína, indicando claramente si los cambios fueron obtenidos experimentalmente o deducidos teóricamente.

Para evitar confusiones, a la hora de describir se debe indicar sobre qué tipo de secuencia se está expresando el cambio. Esto se lleva a cabo añadiendo al principio de la descripción de la variación una letra para indicar el tipo. En la Tabla 7 se muestran los diferentes tipos de descriptores que se pueden utilizar y en la Ilustración 66 se muestran las diferentes secuencias de disponibles así como la relación que hay entre ellas.

Descriptor	Significado
“g.”	Secuencia genómica
“c.”	Secuencia de ADN codificante
“p.”	Secuencia proteínica
“r.”	Secuencia de RNA
“m.”	Secuencia mitocondrial

Tabla 7 Descriptores de tipo de secuencia de referencia.

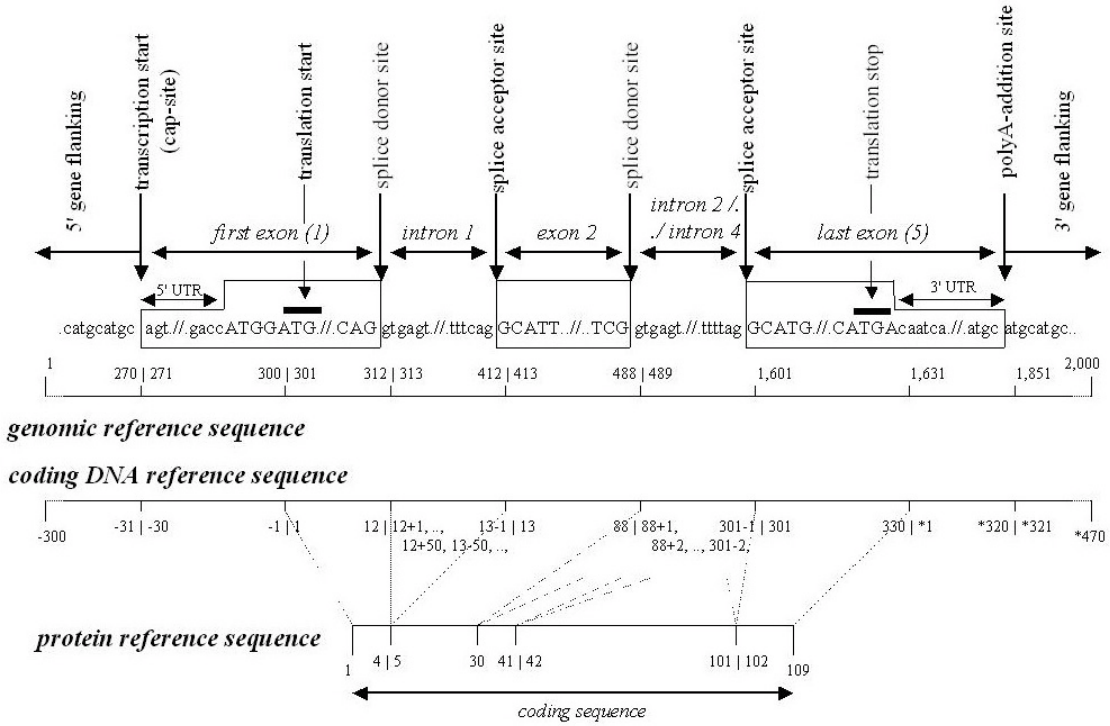

Ilustración 66 Secuencias de referencia (Fuente: página web de HGVS)

A la hora de seleccionar las secuencias de referencia que se utilizan, se recomienda elegir principalmente aquellas que estén en la base de datos RefSeq [44]. Así mismo se deberá indicar tanto el número de acceso de la base de datos como su número de versión. Un ejemplo de descripción sería NM_004006.1:c.3G>T. Esta descripción indicaría que en el gen NM_004006 con número de versión 1, en la secuencia de ADN codificante ha habido una variación en la posición 3 en donde se ha cambiado el nucleótido de la guanina por el de la timina.

A continuación se listan los símbolos de las diferentes variaciones que se pueden describir.

- ">" indica que se ha producido una substitución de nucleótidos.
- "_" se utiliza para hacer referencia a un rango de residuos afectados, separando el primer y el último residuo afectado, por ejemplo c.78_80delACT.
- "del" indica que se ha producido una eliminación de nucleótidos. En el ejemplo anterior significaría que los nucleótidos ACT de las posiciones 78, 79 y 80 han sido eliminados con respecto a la secuencia de referencia del ADN codificante.
- "ins" se utiliza para hacer referencia a una inserción de nucleótidos. Por ejemplo c.78_79ª tendría como resultado una inserción de la adenina entre los nucleótidos 78 y 79 de la secuencia de referencia del ADN codificante.
- "dup" hace referencia a una duplicación de algún nucleótido. En el siguiente ejemplo c.76dupA se está describiendo que la adenina que se encuentra en la posición 76 se duplica, y ahora en lugar de tener tan sólo una adenina tendremos dos. Este tipo de variación podría confundirse con una inserción, pero biológicamente hablando lo que se hace es duplicar el nucleótido de la posición 76, en este caso, y no introducir un nuevo nucleótido entre las posiciones 76 y 77. Por tanto, duplicaciones e inserciones no son lo mismo conceptualmente hablando.

- “inv” indica que se ha producido una inversión de los nucleótidos indicados. Por ejemplo c.76_78inv indicaría que los nucleótidos de este rango se han encontrado en sentido inverso.
- “[]” representa el cambio en un alelo. Como ejemplo podríamos tener c.[76ª>T].
- fs indica que se ha originado un *frameshift* o cambio en la pauta de lectura provocado por la variación que se ha producido. En el ejemplo p.Leu88AlafsX12 se ha producido un cambio en el aminoácido número 88 correspondiente a la Leucina y éste se ha transformado en una Alanina, teniendo como consecuencia un cambio en la pauta de la lectura de la secuencia. Es por ello, que el codón de parada perteneciente a la secuencia a variado y el nuevo codón de parada se encuentra 12 codones después del primer cambio del aminoácido, en este caso el codón 88.

A la hora de describir una variación varios conceptos se han de tener en cuenta, como por ejemplo, que la adenina de la secuencia de referencia de una ADN codificante se referenciará como posición 1, dado que la primera tripleta de nucleótidos en un ADN codificante siempre es ATG, que corresponde a la metionina. Para una cadena de ADN el primer nucleótido que haya será el 1. Del mismo modo también es importante entender cómo se relacionan las posiciones a nivel genético entre el nivel de ADN y el del ADN codificante, así como la relación de estos con la descripción a nivel de aminoácidos (nivel de proteínas).

En la Tabla 8 se muestra dicha correspondencia. En la columna de la izquierda se muestran las diferentes partes del gen, comenzando por las partes cercanas a los 5’ y 3’, seguidas por el conjunto de intrones y exones que forman el ADN. En la segunda columna se muestran los rangos de las posiciones correspondientes a cada exón, intrón, UTR, etc. Los UTR (*Untraslated Region*) son aquellas partes del ADN que no llegarán a codificarse y por tanto no tendrán traducción en cuanto a la posición proteínica. Esto último es debido a que las proteínas se forman a partir de la parte codificante del ADN y por tanto los nucleótidos que se encuentran tanto en intrones como en los UTR no formarán parte de este proceso. En la tercera columna se muestran los rangos de posición para cada nucleótido con respecto a la secuencia de referencia del ADN codificante. Se ha de resaltar que en los intrones, la posición de los nucleótidos se indica con referencia al número del exón que le precede o le sigue dependiendo de si dicho nucleótido está más cerca de un exón u otro. Por último la tercera columna muestra las posiciones de las proteínas, siendo la posición 1 la que hace referencia a la metionina (ATG).

En lo referente a las descripciones realizadas a nivel proteínico se debe saber que lo que describen es la consecuencia del cambio, el origen de éste reside a nivel de ADN. Este tipo de descripción será raramente verificada experimentalmente, y son por tanto normalmente deducidas teóricamente a partir de la información que se tiene a nivel génico. Sin embargo se ha de ser cuidadoso en este sentido, tan sólo se deberá describir el efecto a nivel proteínico y en ningún cambio se deberá introducir información que se conozca a partir del cambio a nivel DNA. Estos dos apuntes pueden parecer contradictorios, pero no lo son. Veámoslo con el siguiente ejemplo, la variación c.9_10insAGT (una inserción de tres nucleótidos: adenina, guanina y timina, entre las posiciones 9 y 10 de la secuencia de referencia de ADN codificante).

Parte del gen		Nº nucleótido Sec. Referencia genómica	Nº nucleótido Sec. Referencia ADN codificante	Nº nucleótido Sec. Referencia proteina
5' región flanqueante		1 a 270	(-300 a -31)	-
Exón 1	5' UTR	271 a 300	-30 a -1	-
	Región codificante	301 a 312	1 a 12	1 a 4
Intrón 1		313 a 412	12+1 … 12+50, 13-50 … 13-1	-
Exón 2		413 a 488	13 a 88	5 a 29 (30)
Intrón 2		489 a 689	88+1 … 88+100, 89-100 … 89-1	-
Exón 3		689 a 723	89 a 123	30 a 41
Intrón 3		724 a 1023	123+1 … 123+150, 124-150 … 124-1	-
Exón 4		1024 a 1200	124 a 300	42 a 100
Intrón 4		1201 a 1600	300+1 ... 300+200, 301-200 ... 301-1	-
Exón 5	Región codificante	1601 a 1630	301 a 330	101 a 109
	3' UTR	1631 a 1850	*1 a *220	-
3' región flanqueante		1851 a 2000	(*221 a *370)	-

Tabla 8 Correspondencia entre los diferentes niveles (ADN, ADN codificante y proteína)

Supongamos que la secuencia que tenemos es ATGTCAAGCTCT, que traducida a nivel proteínico es MetSerSerSer. Una vez introducimos el cambio en la secuencia obtenernos las siguiente cadena ATGTCAAGCAGTTCT, que en este caso codifica como MetSerSerSerSer. De este modo, a partir de la información a nivel genético obtenemos la nueva cadena de aminoácidos. En este caso el cambio debería describirse como p.Ser3dup, ya que se ha originado un duplicado de la Serina. El error al que nos referíamos más arriba estaría en dar a nivel proteínico la siguiente descripción p.Ser2_Ser3insSer, ya que sabemos que lo que se ha originado es una inserción de los nucleótidos AGT que componen la nueva Serina, y por tanto no tendríamos una duplicación sino una inserción. Debemos tener claro que no se debe ofrecer más información a nivel proteínico que lo que vemos en dicho nivel.

8.2. Traducción

Para el estudio de la notación de las variaciones se deben conocer conceptos como los de exón, intrón, ADN codificante ó ADN, así como la relación existente entre ellos. Estas partes son las que componen la parte central de la noción de gen y han sido modeladas tal y como se muestran en la Ilustración 67. Este esquema modela las relaciones existentes entre un gen y su transcripción. También se ha añadido al esquema la vista de variaciones de cara a representar

la relación existente entre dicha vista y la vista de transcripción a la hora de llevar a cabo una descripción en formato HGVS.

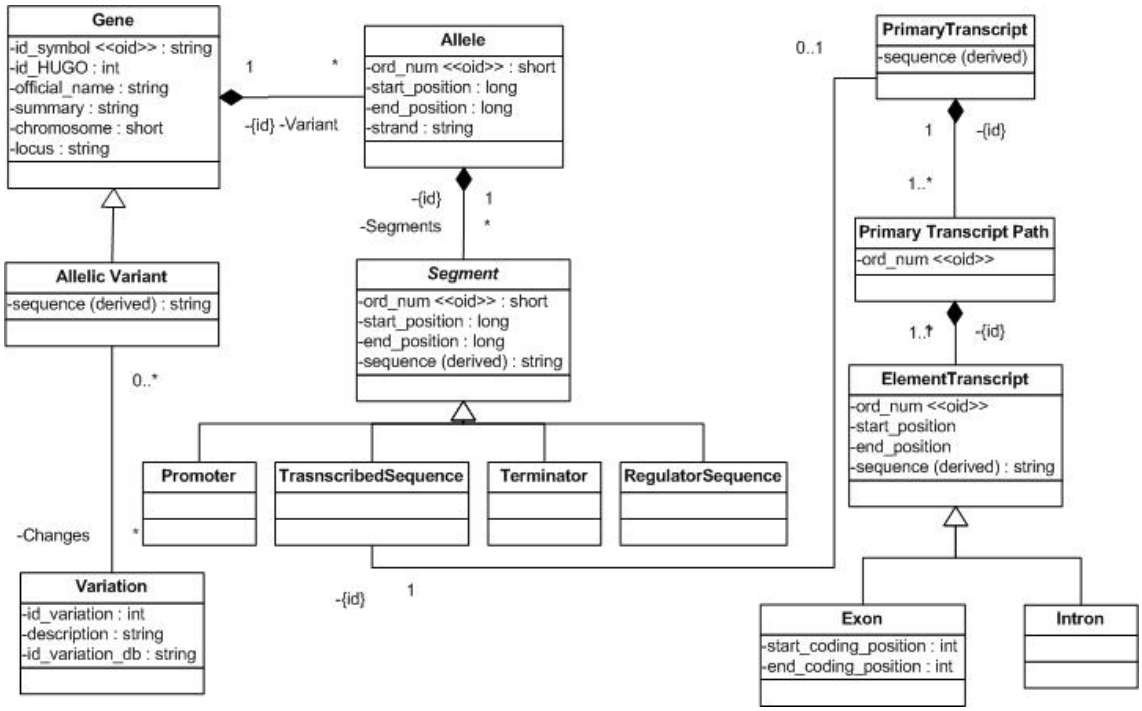

Ilustración 67 Esquema conceptual transcripción

La clase *PrimaryTranscript* representa la copia transcrita de ADN a ARN de la secuencia transcribible, que viene dada por la entidad *TranscribedSequence*. El atributo *sequence* de la clase *PrimaryTranscript* viene derivado a partir de la clase *Segment*. La entidad *PrimaryTranscriptPath* se incluye en el modelo para modelar las diferentes particiones del transcrito primario. Un transcrito primario se compone de varios elementos que tienen un orden determinado, éste orden viene determinado por el atributo *ord_num* de la entidad *PrimaryTranscriptPath*. Los elementos de un transcrito pueden ser exones e intrones. Ambas entidades vienen representadas por las clases *Exon* e *Intron*, respectivamente y cuelgan de la entidad *ElementTranscript*. Cualquier elemento de transcripción vendrá dado por una posición de inicio y otra de fin, con respecto a la secuencia de ADN a la que pertenecen. A su vez dicha subsecuencia viene representada por el atributo *sequence* que se deriva del transcrito primario. Los exones por su parte poseen dos atributos más que indican la posición en la que empiezan a codificarse y en la que termina. Esto es debido a que a la hora de producirse una traducción a proteínas tan sólo se lee la parte codificante de la secuencia estudiada, y esta viene dada por la unión de los exones que lo componen a excepción de los UTRs que forman parte del primer exón y del último exón (Ilustración 68). Los UTRs vienen determinados por las posiciones comentadas.

Conocer esta información es imprescindible a la hora de poder realizar una descripción correcta de las variaciones extraídas mediante el uso de GeIS. De este modo se dispondrá en el Sistema de Información Genético de todos los datos necesarios para llevar a cado dicha descripción.

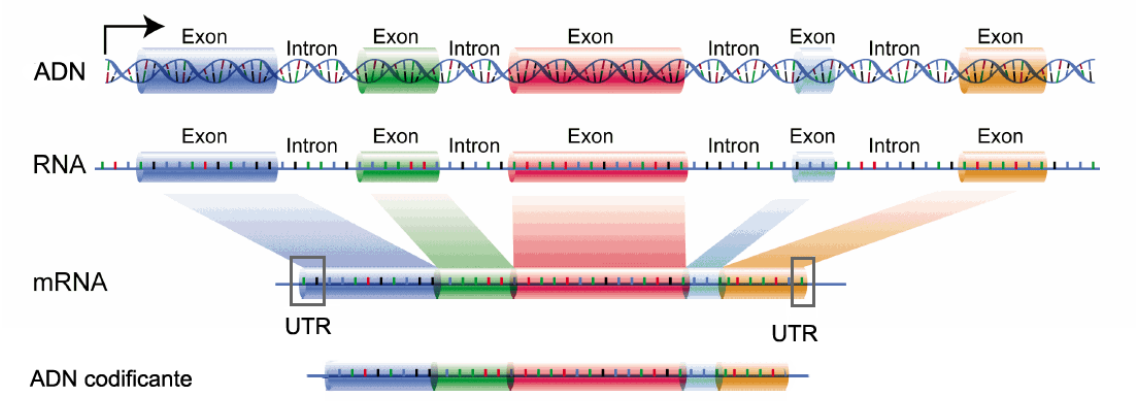

Ilustración 68 Transformación de ADN hasta ADN codificante (imagen modificada de http://www.genome.gov)

En el GeIS la información de las variaciones se encuentra a nivel genético, y las posiciones por tantos se encuentran expresadas por tanto en base al gen al que pertenecen.

Realizar una descripción de una variación simple de tipo inserción, borrado o sustitución a nivel genético a partir de la información disponible es trivial. Tan sólo es seguir las pautas dadas por HGVS. Por ejemplo para describir adecuadamente la inserción de tres nucleótidos AAA en la posición 5991 del gen BRCA2, será necesario indicar el nombre de la secuencia del gen y la versión sobre la que se está trabajando, en este caso por ejemplo NG_012772.1. Seguidamente se indicará el nivel en el que se describe la variación (ADN, ARN, Proteína) y el rango en el que se introduce. Para que la variación se encuentre en la posición actual 5991, la inserción de nucleótidos deberá hacerse entre las posiciones 5990 y 5991. Finalmente se introducirá el tipo de cambio producido, en este caso una inserción, y los nucleótidos que se han insertado. Por tanto la variación aquí expuesta quedará descrita como NG_012772.1:g.5590_5991insAAA.

Ya que toda la información está expresada a nivel genético en el GeIS propuesto, la descripción a nivel de ARN deberá extraerse de manera derivada. Del GeIS podemos obtener la secuencia codificante compuesta por los exones sin su UTR. Teniendo las posiciones de los exones a nivel genético podemos extraer sus posiciones a nivel ARN mediante una simple traslación de posiciones, que se traduce en una resta entre la posición que se desea traducir y la posición inicial del comienzo de la secuencia codificante. El resto de información de la descripción a nivel de ADN no variará, tan solo cambiará el nombre de la secuencia, el código de tipo, y las posiciones. Así pues la descripción anterior ahora sería NM_000059.3:c.9_10insAAA. En este apartado las variaciones intrónicas no serán traducidas, ya que no forman parte de la secuencia que será codificada. La única complicación existente en este apartado es aquella referente a las variaciones que empiezan o acaban en un intrón. Para comentar este caso se hará uso de un ejemplo, Ilustración 69. En esta imagen se muestra un borrado de 24 nucleótidos respecto a la secuencia de referencia. El borrado comienza en una posición intrónica, en este caso la posición 96, y por tanto a la hora de traducirse se debe dar dicha posición en base a la posición de la secuencia codificante, en este caso la posición 20. Es por ello que como resultado en la descripción se dice que el borrado de nucleótidos comienza en la posición “20-3” y termina en la 24 (que es una posición totalmente exónica).

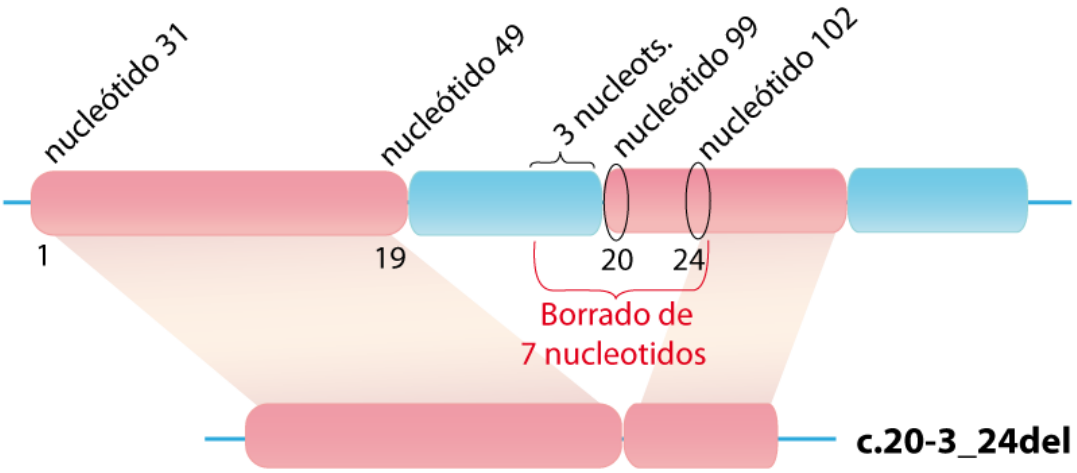

Ilustración 69 Descripción a nivel de ARN

Sin embargo, la traducción a nivel proteínico no es un cambio trivial. Esto es debido a su representación mediante aminoácidos. Los aminoácidos son el resultado de un conjunto de tres nucleótidos, y un mismo aminoácido puede ser traducido a partir de diferentes combinaciones de nucleótidos. La redundancia en el código genético es algo que ya se ha comentado en capítulos anteriores. La traducción a nivel proteínico se debe hacer desde el nivel de ARN y una vez originados los cambios traducir tanto la secuencia de referencia como la resultante a una secuencia de aminoácidos y comentar los cambios que se observan.

8.2.1. Algoritmo propuesto

Con toda la información en el GeIS y todo el conocimiento adquirido se propone en este apartado un algoritmo descrito en pseudocódigo para llevar a cabo la descripción de las variaciones a nivel proteínico. A continuación se explican los diferentes casos posibles, adjuntando algún ejemplo de manera visual, y ofreciendo la solución en pseudocódigo. Los casos se han separado según el tipo de variación y si ésta producía un cambio en la pauta de lectura o no. Para saber si un cambio produce *frameshift* se puede seguir el algoritmo de la Tabla 9.

Si variación es Borrado o Inserción
    Si no es múltiplo de 3 → **Frameshift**
    Si lo es → No es Frameshift
Si variación es Indel
    Si la cantidad de nucleótidos eliminados no es igual a la insertada y la longitud de alguna de las cadenas (eliminada/insertada) o ambas no es múltiplo de 3 → **Frameshift**
    Si no → No es frameshift (cadenas de longitud igual, cadenas de longitud diferente con ambas cadenas múltiplo de 3)

Tabla 9 Pseudocódigo para saber si se produce frameshift

8.2.1.1. Inserciones **sin frameshift**

Este caso viene identificado por la inserción de un número múltiplo de tres de nucleótidos. Al poderse dividir entre tres la cantidad de nucleótidos introducida, no se produce ningún cambio en la pauta de lectura y por tanto no se genera lo que se denomina *frameshift.*

Inserción de tres nucleótidos en el gen BRCA2 justo detrás de un codón completo

La descripción a nivel de ADN es NG_012772.1: g.5990_5991insAAA y a nivel de ARN es NM_000059.3:c.9_10insAAA.

A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 236

Referencia: gaggaatatcgtaggtaaaaatgcctaat - - - ggatccaaagagaggccaaca

Muestra: gaggaatatcgtaggtaaaaatgcctattAAAtggatccaaagagaggccaaca

En rojo y en mayúscula se resaltan los nucleótidos que se insertan en la cadena mutada, en azul los nucleótidos que está fuera de la parte codificante del ADN.

En este caso se introduce un codón completo nuevo, justo detrás de otro codón. Por tanto el cambio que se produce es la introducción del aminoácido Lisina entre los aminoácidos Isoleucina y Glicina: NP_000050.2:p.Ile3_Gly4insLys.

Inserción de tres nucleótidos en el gen BRCA2 rompiendo un codón

Resultado indel

El cambio que se produce a nivel ADN es NG_012772.1: g.5988_5989insGAC y a nivel de ADN codificante NM_000059.3:c.7_8insGAC.

A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 234

Referencia: atatcgtaggtaaaaatgccta- - - ttggatccaaagagaggccaacattttt

Muestra: atatcgtaggtaaaaatgcctaGACttggatccaaagagaggccaacattttt

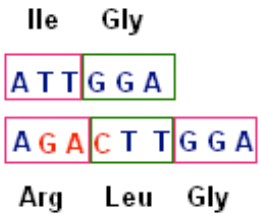

Ilustración 70 Indel en secuencia de aminoácidos

En rojo se muestran los nucleótidos que se insertan en la cadena mutada, en gris los nucleótidos que está fuera del DNA codificante.

En este caso se han introducido 3 nucleótidos rompiendo el codón tres. Dicho codón correspondía a la Isoleucina, que por tanto no aparecerá en el aminoácido resultante y en su lugar se producen la Arginina y la Leucina. Esto se representa como un indel, puesto que el

aminoácido que existía desaparece y en su lugar aparece dos aminoácidos nuevos: NP_000050.2:p.Ile3delinsArgLeu tal y como se muestra en la Ilustración 70.

Resultado inserción

Teniendo el cambio NG_012772.1:g.5988_5989insGAA, traducido a nivel mRNA como NM_000059.3:c.7_8insGAA. Se muestra a continuación parte de las cadenas de referencia y de la mutada. Los nucleótidos que se han insertado se muestran en color rojo y en mayúscula.

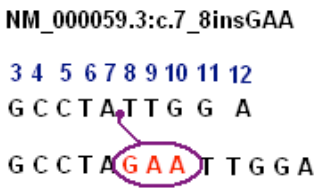

Ilustración 71 Inserción de 3 nucleótidos

Localización: 234

Referencia: ggaatatcgtaggtaaaaatgccta - - - ttggatccaaagagaggccaacatt

Mutación: ggaatatcgtaggtaaaaatgcctaGAAttggatccaaagagaggccaacatt

Si se mira más detenidamente se puede ver como se ha originad una inserción de los nucleótidos GAA entre las posiciones 7 y 8 del mRNA. (Ilustración 71)

En este caso se produce una inserción de un aminoácido en lugar de producirse un indel, que sería el caso habitual. Esto tiene lugar debido al hecho de que un aminoácido puede ser el resultado de diferentes combinaciones de aminoácidos. Como se ve en la Ilustración 72, la rotura del codón produce dos aminoácidos, uno de los cuales es igual al inicial.

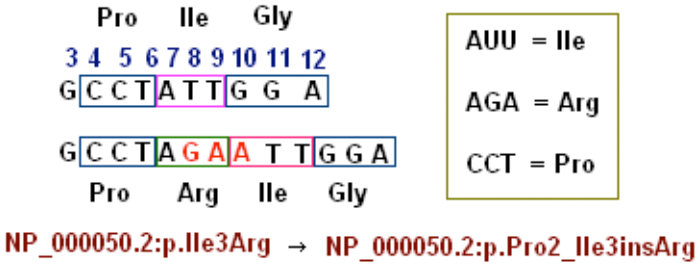

Ilustración 72 Cambio traducido a una inserción de un aminoácido

Conclusión

De este estudio se obtiene que una inserción de una cadena de nucleótidos, que no produce frameshift, puede traducirse como una inserción o un indel. Dicha traducción dará como resultado normalmente una inserción cuando esa cadena se introduzca justo detrás de un codón, dando paso a codones nuevos.

Por otra parte cuando la tripleta de nucleótidos (o el conjunto múltiplo de tres) se inserte de manera que rompa la estructura de un codón de la secuencia de referencia, entonces el

cambio afectará a más de un aminoácido. En este caso normalmente se traducirá como un indel. Sin embargo, si algún codón nuevo se traduce como el mismo aminoácido ya existente, podrá traducirse como una inserción como se ha visto en el ejemplo anterior.

En este caso el algoritmo en pseudocódigo sería el que se muestra en la Tabla 10.

Si aminoácido viejo = aminoácido nuevo 1 ó aminoácido viejo = último aminoácido nuevo **Inserción** Recorrer la cadena de nucleótidos insertada, sin contar el primer aminoácido que se forma, e indicar el cambio como una inserción a partir de la siguiente posición. Sino **Indel** Recorrer la cadena de nucleótidos insertados en la nueva secuencia y obtener la nueva cadena de aminoácidos. Indicar el aminoácido que varía y los nuevos que han sido insertados.

Tabla 10 Pseudocódigo para la nomenclatura de las inserciones sin frameshift

8.2.1.2. Inserción con frameshift

Inserción de cuatro nucleótidos en el gen BRCA2 (caso 1)

El cambio que se produce a nivel de ADN codificantes es NM_000059.3:c.7_8insGAAA. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 234

Referencia: ggaatatcgtaggtaaaaatgccta - - - - ttggatccaaagagaggccaacatt

Muestra: ggaatatcgtaggtaaaaatgcctaGAAAttggatccaaagagaggccaacatt

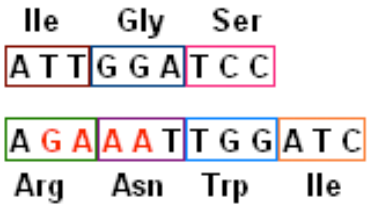

Ilustración 73 Inserción con frameshift

En este caso se indica que ha habido un cambio del aminoácido 3, la Isoleucina, por la Arginina. En la Ilustración 73 vemos que a partir de la introducción de esta cadena de nucleótidos, no se produce ninguna coincidencia, y todo se ve alterado a consecuencia del cambio en la pauta de lectura. La traducción de la nomenclatura a nivel proteínico es NP_000050.2:p.Ile3Argfs*12. Encontrándose 12 nucleótidos más a la derecha el nuevo codón de parada.

Inserción de cuatro nucleótidos en el gen BRCA2 (caso2)

El cambio que se produce a nivel de ADN codificantes es NM_000059.3:c.8_9insCAAA. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 235

Referencia: gaatatcgtaggtaaaaatgcctat - - - - tggatccaaagagaggccaacattt

Muestra: gaatatcgtaggtaaaaatgcctatCAAAtggatccaaagagaggccaacattt

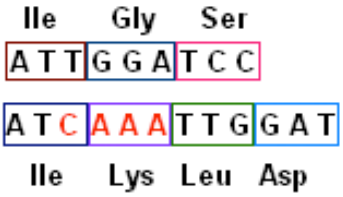

Ilustración 74 Inserción con frameshift, caso 2

En este cambio se ha producido un cambio del aminoácido 4, Glicina, por uno nuevo, lisina. En este caso, como se ve en la Ilustración 74, lo que sucede es que ATT y ATC se corresponden con el mismo aminoácido, la isoleucina. Por tanto lo que se debe hacer es referenciar al siguiente aminoácido que sea distinto. La traducción a nivel proteínico sería pues NP_000050.2:p.Gly4Lysfs*11. Encontrándose el nuevo codón de parada 11 nucleótidos a la derecha del cambio.

Conclusiones

Para las inserciones en las que se produce un cambio en la pauta de lectura lo que se ha de hacer es (Tabla **11**) comparar los aminoácidos de izquierda a derecha desde el aminoácido donde se produjo el cambio. Si este ya no coincide con el aminoácido que estaba en la cadena de referencia, ese será el resultado que se ofrezca en la descripción en formato HGVS. Si por el contrario se produjera una coincidencia se deberán recorrer las secuencias, comparando aminoácidos hasta llegar a uno que no coincida, y entonces se escribirá ese cambio como resultado.

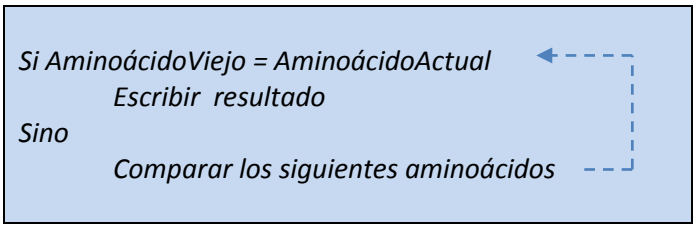

Tabla 11 Pseudocódigo para la nomenclatura de las inserciones con frameshift

8.2.1.3. Deleción sin frameshift

En esta opción tan sólo hay dos casos a estudiar: (1) el estudio del borrado de un número concreto de codones, es decir, de aminoácidos completos y (2) el borrado de codones parciales. A continuación hay un ejemplo de cada caso.

Borrado de codones completos

Se produce un borrado de tres nucleótidos en el gen BRCA2, NM_000059.3:c.7_9del. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 234

Referencia: aggaatatcgtaggtaaaaatgcctATTggatccaaagagaggccaacatttt

Muestra: aggaatatcgtaggtaaaaatgcct- - - ggatccaaagagaggccaacatttt

Este es el caso más sencillo de codificar. La eliminación corresponde a un número concreto de aminoácidos completos, tal y como se ve en la y por tanto la codificación consta tan sólo de la traducción de dichos nucleótidos en sus aminoácidos correspondientes, para indicar cuáles han sido eliminados. La traducción a nivel proteínico será NP_000050.2:p.Ile3del.

Borrado de codones parciales

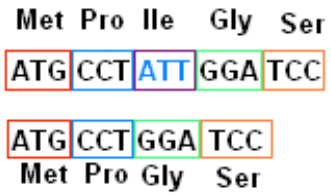

Ilustración 75 Borrado de un codón

Se produce un borrado de tres nucleótidos en el gen BRCA2, NM_000059.3:c.8_10del. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 235

Referencia: ggaatatcgtaggtaaaaatgcctaTTGgatccaaagagaggccaacattttt

Muestra: ggaatatcgtaggtaaaaatgccta - - - gatccaaagagaggccaacattttt

En este caso, al producirse una eliminación de nucleótidos que afecta a más de un aminoácido provoca que no sólo haya una eliminación sino que los nucleótidos restantes que no han sido eliminados, pero que formaban parte de los aminoácidos cuyos nucleótidos han sido eliminados, formen un nuevo aminoácido. De ahí que se represente como un indel. Se eliminan la Isoleucina y la Glicina y se "inserta" (aparece como nueva) la Arginina. NP_000050.2:p.Ile3_Gly4delinsArg.

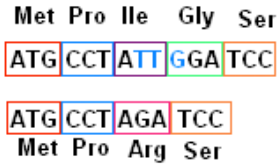

Ilustración 76 Borrado parcial de codones

Conclusiones

Para llevar a cabo la notación de este tipo de variaciones lo que se debe de hacer es obtener las posiciones que se ven afectadas por la variación y mirar por qué aminoácidos se traducen. Para así saber si la variación afecta a codones completos o no (Tabla 12).

Obtener las posiciones afectadas por la variación.

Obtener proteínas pertenecientes.

Mirar si la posición del primer nucleótido afectado está en la posición 1 del aminoácido.
Si lo está

Afecta a aminoácidos completos:
Traducir los nucleótidos a sus aminoácidos correspondientes. Deleción.

Sino

No afecta a aminoácidos completos:
Seleccionar los nucleótidos que están antes del primer nucleótido afectado y pertenecen al mismo aminoácido.

Seleccionar los nucleótidos que están después del primer nucleótido afectado y pertenecen al mismo aminoácido..

Unir los nucleótidos obtenidos y traducir el aminoácido resultante, se traducirá como un indel. Tener en cuenta que si el nuevo aminoácido coincide con el primero o el último que se eliminó, entonces en lugar de ser un indel, se traducirá como una deleción.

Tabla 12 Pseudocódigo para la nomenclatura de deleciones sin frameshift

8.2.1.4. Deleciones con frameshift

En el caso de los borrados de nucleótidos que producen un cambio en la pauta de lectura, el tipo de cambio que ha de documentarse a nivel proteínico es el de substitución. El único caso especial con el que puede aparecer aquí es que el primer aminoácido que se elimine coincida con el que se sustituye, y por tanto se deberá indicar que la substitución ocurre en el aminoácido siguiente, a no ser que este también coincida con el nuevo aminoácido y así sucesivamente. Esto se puede verse reflejado en el siguiente ejemplo.

Borrado de diez nucleótidos en el gen BRCA2, NG_012772.1:g.5986_5995del, NG_012772.1(BRCA2):c.5_14del. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 5986

Referencia: ggaggaatatcgtaggtaaaaatgcCTATTGGATCcaaagagaggccaaca

Muestra: ggaggaatatcgtaggtaaaaatgc - - - - - - - - - - caaagagaggccaaca

En este caso el primer aminoácido que se elimina CCT se sustituye por el codón CCA y ambos corresponden a la Prolina, es por ello que se hace referencia al siguiente grupo eliminado, en este caso el de la Isoleucina, cambiándolo por la Lisina. La nomenclatura en formato HGVS de esta variación sería NP_000050.2(BRCA2):p.Ile3Lysfs*19.

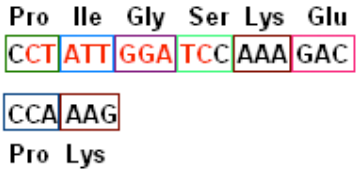

Ilustración 77 Borrado con frameshift

Conclusión

Como conclusión podemos indicar que los borrados de nucleótidos en frameshift se codifican como sustituciones para el caso de la nomenclatura HGVS a nivel proteínico.

Se deberá (Tabla 13) ir comparando uno a uno los aminoácidos de la secuencia de referencia con los de los aminoácidos de la secuencia consenso a partir de la posición donde se produce la deleción. Esta comparación terminará cuando nos topemos con dos aminoácidos que no coinciden. Llegados a este momento cuando podremos indicar el cambio que se ha producido.

Si AminoácidoViejo ≠ AminoácidoActual *Escribir resultado* *Sino* *Comparar los siguientes aminoácidos*

Tabla 13 Pseudocódigo para la nomenclatura de deleciones con frameshift

8.2.1.5. Indel sin frameshift

Borrado de tres nucleótidos e inserción de seis en el gen BRCA2

Cambio representado a nivel de mRNA en notación HGVS como NM_000059.3:c.11_13delinsAAATTT. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 238

Referencia: atatcgtaggtaaaaatgcctattgGAT- - - - - - ccaaagagaggccaaca

Muestra: atatcgtaggtaaaaatgcctattg - - - AAATTTccaaagagaggccaaca

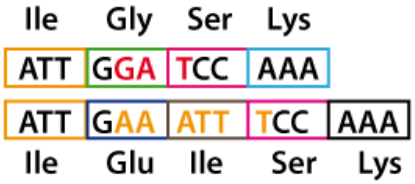

Ilustración 78 Indel sin frameshift

En este caso como el último aminoácido afectado coincide con el último aminoácido nuevo insertado, éste queda fuera del resultado y es por eso que tan sólo que tan sólo se indica que

ha habido una deleción de la Glicina y en su lugar se ha insertado una Glutamina y una Isoleucina, NP_000050.2:p.Gly4delinsGluIle.

Borrado de un codón e inserción de dos en BRCA2

Cambio representado a nivel de mRNA en notación HGVS como NM_000059.3:c.7_9delinsCCCCCC. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 234

Referencia: aggaatatcgtaggtaaaaatgcctATT- - - - - ggatccaaagagaggccaac

Muestra: aggaatatcgtaggtaaaaatgcct - - -CCCCCCggatccaaagagaggccaac

Este caso es de los sencillos, se elimina un aminoácido entero y se introducen dos aminoácidos nuevos que no coinciden ni con el que se ha eliminado ni con el siguiente a éste. Por tanto se produce un cambio sencillo de un aminoácido por dos, una Isoleucina por dos Prolinas. Siendo traducido en formato HGVS como NP_000050.2:p.Ile3delinsProPro.

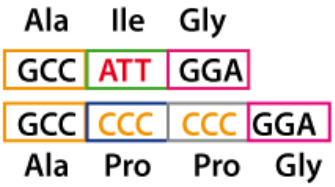

Ilustración 79 Indel sin frameshift 2

En el caso de los indels, a diferencia de los de borrado o inserción, se han separado los algoritmos de resolución según si se produce un cambio en la pauta de lectura o no.

Conclusión

Para obtener los aminoácidos que cambian y deben ser mostrados en la nomenclatura HGVS se obtiene la cadena de aminoácidos afectados por la eliminación de nucleótidos en la secuencia de referencia, y la cadena de aminoácidos nueva que se origina después de la eliminación y la inserción de nucleótidos.

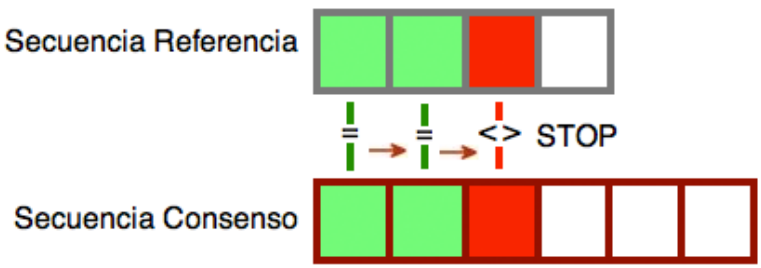

Ilustración 80 Indel sin frameshift - Paso 1

Después de esto se ha de ir comparando la similitud entra aminoácidos tal y como se muestra en la Ilustración 80.

Cuando se encuentren dos aminoácidos que no coincidan entonces se para la iteración y se deberá realizar la misma comparación pero esta vez comenzando desde el final hacia el principio, habiendo eliminado de las secuencias a comparar los aminoácidos que coincidieron en la fase previa (Ilustración 81).

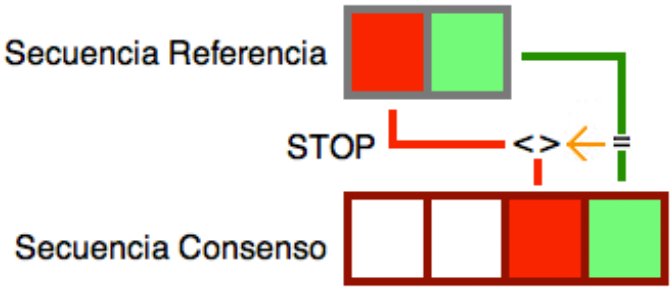

Ilustración 81 Indel sin frameshift - Paso 2

Después de pasar por las dos fases, en el resultado se mostrarán los aminoácidos restantes, en color rosado en la Ilustración 82.

Ilustración 82 Indel sin frameshift - Paso 3

En el caso de que en la cadena de aminoácidos de la secuencia de referencia no quede ningún elemento, entonces el cambio se representa como una inserción.

Si en la cadena de aminoácidos de la secuencia consenso no queda ningún aminoácido, entonces el cambio se representa como una deleción o borrado de aminoácidos.

En cualquier otro caso el cambio será representado como un indel o una sustitución, si en este último caso tan sólo se contara con un aminoácido en cada cadena.

En el caso excepcional de que no quedaran aminoácidos no coincidentes entonces se expresaría el cambio como “p.?”.

8.2.1.6. Indel con Frameshift

Deleción de dos nucleótidos e inserción de tres en BRCA2, caso 1

Este cambio se expresa en la nomenclatura HGVS a nivel de ADN codificante como NM_000059.3:c.7_8delinsCCC. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 234

Referencia: aggaatatcgtaggtaaaaatgcctAT - - - tggatccaaagagaggccaaca

Muestra: aggaatatcgtaggtaaaaatgcct - - CCCtggatccaaagagaggccaaca

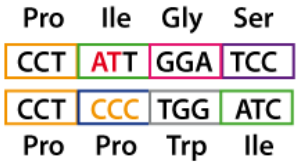

Ilustración 83 Indel con frameshift

Este caso es el de un indel que produce un cambio en la pauta de lectura. Al producirse un *frameshift* los aminoácidos que van detrás del cambio serán totalmente distintos al resto de aminoácidos iniciales, y por tanto no tiene sentido estudiarlos. Así pues, como la Isoleucina no coincide con la Prolina se indica el cambio que se ha encontrado, el de una sustitución de Isoleucina por Prolina en el aminoácido número 3 con su correspondiente codón de parada en el aminoácido 11. El cambio en nomenclatura HGVS sería NP_000050.2:p.Ile3Profs*11.

Borrado de dos nucleótidos e inserción de tres en el gen BRCA2, caso 2

Este cambio se expresa en la nomenclatura HGVS a nivel de ADN codificante como NM_000059.3:c.7_8delinsATC. A continuación se muestra parte de la cadena de referencia y de la mutada.

Localización: 234

Referencia: aggaatatcgtaggtaaaaatgcctAT- - -tggatccaaagagaggccaaca

Muestra: aggaatatcgtaggtaaaaatgcct- -ATCtggatccaaagagaggccaaca

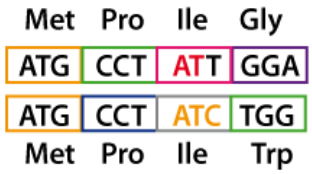

Ilustración 84 Indel con frameshift, caso especial

Este caso lo que tiene de especial es que la inserción de nucleótidos coincide con el primer aminoácido afectado por el cambio, y por lo tanto este no se ve afectado en lo que al significado de la proteína se refiere. Es por ello que se indica el siguiente cambio que se encuentra, que es el de la Glicina por el Triptófano, que en nomenclatura HGVS se describe como NP_000050.2:p.Gly4Trpfs*10, encontrándose el nuevo codón de parada diez nucleótidos a la derecha del cambio.

Conclusión

En estos casos el algoritmo a seguir es bastante sencillo. Lo que se debe de hacer es los mismo que se hacía en el paso uno del algoritmo anterior (Ilustración 80), en el momento en que los aminoácidos no coincidan se deberá mostrar dicho cambio como una sustitución de esos

aminoácidos. Esto es debido a que a partir de ese momento ninguno de los aminoácidos que le siguen se mantendrá igual, debido al cambio en la pauta de lectura.

8.3. Lecciones aprendidas

En el caso del diagnóstico genético hasta ahora cada biólogo ofrecía una descripción propia, según sus criterios, a la hora de describir las variaciones encontradas en sus estudios. Esto conllevaba una mayor desestructuración de los datos antes comentados. Es necesaria, por tanto, la descripción formalizada de las variaciones que se estudian. Actualmente se disponen de las guías dadas por el consorcio HGVS para la nomenclatura de las variaciones. Es imprescindible de cara al estudio de las variaciones que éstas estén bien identificadas.

Añadir en este aspecto en los GeIS supone una gran ventaja a la hora de poder extraer la información de las variaciones y, obtener los datos necesarios para poder llevar a cabo una descripción formalizada de acuerdo a la nomenclatura HGVS.

En este capítulo se ha estudiado cómo transformar las variaciones almacenadas en un GeIS a dicha nomenclatura HGVS. Las variaciones de tipo indel, inserción y borrado han sido analizadas con el fin de obtener los pasos convenientes para su traducción a los tres niveles necesarios de descripción: nivel ADN, nivel mRNA y nivel proteínico.

La transformación de una descripción de un nivel genético a un nivel de ADN codificante resulta trivial, y tan sólo se corresponde a un traslado de un sistema de posicionamiento a otro. En cambio, debido a que un aminoácido puede derivarse a partir de diferentes combinaciones de aminoácidos, la transformación de una descripción de un nivel de mRNA a una descripción a nivel proteínico no es en absoluto trivial. Es necesario un estudio más detenido, estudio que se ha llevado a cabo en este capítulo ofreciendo una serie de algoritmos en pseudocódigo para poder llevar a cabo dicha transformación.

Prueba de conceptos

Como primer resultado del esquema conceptual CSHG, se ha creado una base de datos genómica (HGDB) [37, 38] para almacenar la información sobre las variaciones. Dicha base de datos es utilizada por el presente proyecto GeIS. Gracias a la conceptualización del dominio tanto los problemas de la heterogeneidad y la dispersión de datos como la automatización de tareas bioinformáticas, como puede ser el análisis de secuencias genéticas, quedan resueltos.

Teniendo dichos problemas resueltos y una base de datos cargada con la información necesaria, el siguiente paso a seguir para completar el estudio del GeIS es el la generación de un artefacto software encargado de la explotación de los datos almacenados en HGDB.

Las lecciones aprendidas, durante la elaboración del estudio preliminar del desarrollo de un sistema de información para el diagnóstico genético, se ponen en práctica mediante la implementación de un prototipo funcional capaz de llevar a cabo un diagnóstico génico a partir de una secuencia de ADN de un paciente. La elección de esta funcional viene dado por el hecho de que una de las tareas más comunes y que más tiempo requieren en el mundo de la genómica es el del análisis de secuencias genéticas, tal y como se indicó en la motivación de este trabajo, más atrás indicado.

Esta herramienta se encargará de recibir dicha secuencia y proporcionar un informe que ayude al doctor a diagnosticar una determinada enfermedad. El bioinformático tan sólo tendrá que introducir la muestra en el formato adecuado y revisar los resultados proporcionados por el sistema, olvidándose así de todo lo relacionado con el tratamiento manual de las secuencias, de las búsquedas infinitas entre la bibliografía disponible y de la traducción de dichas variaciones al formato HGVS. Esta automatización se ha podido llevar a cabo gracias a la conceptualización del dominio mediante el modelo conceptual presentado en esta tesis. La información sobre genes, variaciones, fenotipos y referencias bibliográficas se representa ahora como entidades conceptuales perfectamente definidas. Gracias a esta conceptualización, los problemas de la heterogeneidad y dispersión de la información han sido solucionados, evitando el preprocesado manual de parte de la información que no es legible por los ordenadores y garantizando así la calidad de los datos almacenados.

9.1. Proceso de análisis

El proceso de análisis que este prototipo lleva a cabo se resume en la **¡Error! No se encuentra el origen de la referencia.**. Algunas de las entidades del esquema conceptual se utilizan en diferentes pasos y están representadas en la figura mediante cajas rectangulares blancas. El proceso se divide en cinco diferentes pasos:

1. **Introducción de los datos**. El biólogo selecciona un gen de la lista proporcionada por la herramienta, por ejemplo el gen BRCA2, e introduce la muestra de ADN que desea analizar. La introducción de la muestra puede realizarse tanto manualmente como mediante la carga de un archivo en formato FASTA.
2. **Informe de alineamiento**. De acuerdo con el gen seleccionado, la herramienta localiza la secuencia de referencia adecuada mediante el uso de la entidad *Allelic Reference*. Después de esto, el proceso de alineamiento, que se comentó en el capítulo de

Alinemiento más atrás, se lleva a cabo encontrando las variaciones existentes en la muestra estudiada. Mediante la utilización del esquema conceptual, cada diferencia descubierta se formaliza como una instancia de la entidad *Variation*. Esta formalización, que no está presente en otras herramientas o bases de datos actualmente, es independiente de cualquier salida dada por una herramienta de alineamiento y proporciona una forma útil para el intercambio de variaciones. Se genera por tanto un informe que resuma todos los cambios mediante el uso de ésta entidad.

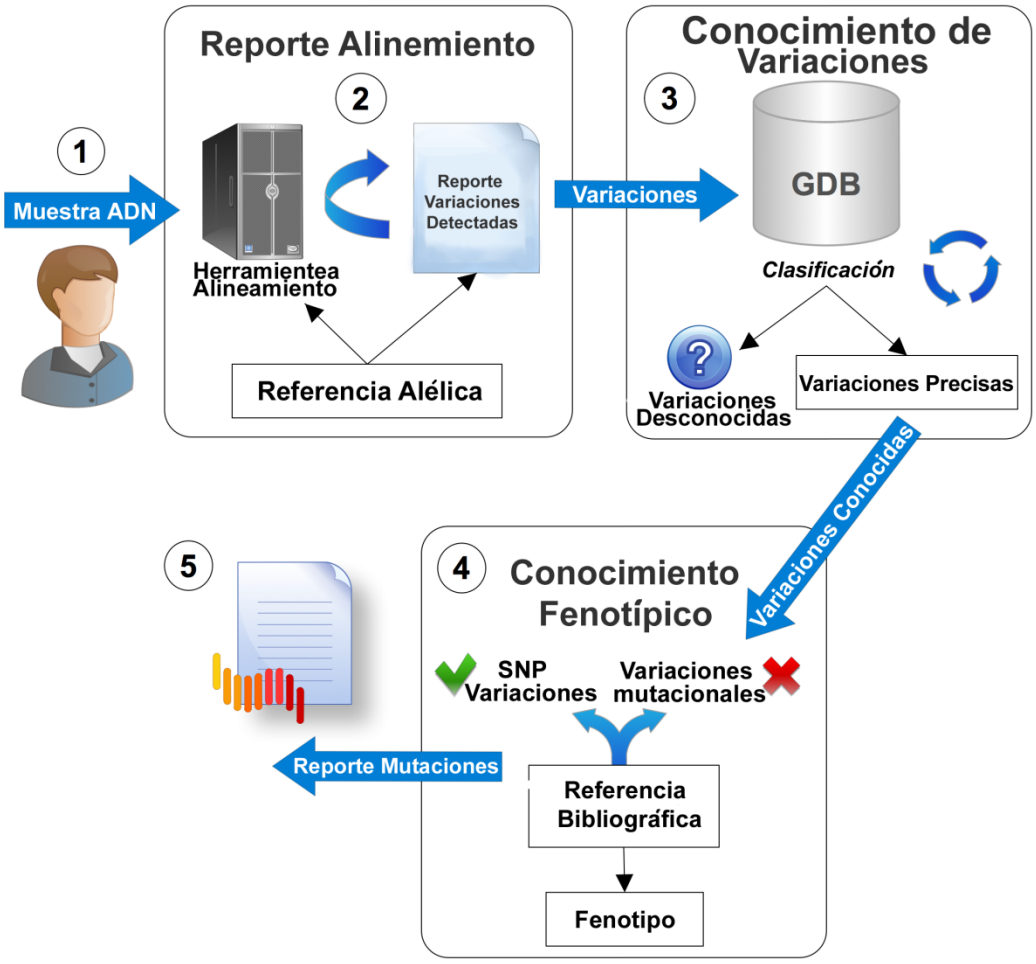

Ilustración 85 Herramienta de análisis de mutaciones basada en CSHG

3. **Conocimiento sobre las variaciones**. Gracias al reporte generado en la fase previa, el problema de clasificación se simplifica. Las variaciones se localizan de acuerdo a una secuencia de referencia ampliamente conocida y sus posiciones coinciden con los datos almacenados en HGDB. Por tanto, para cada variación se realiza una consulta a HGDB para determinar si dicho cambio ha sido definido como una variación precisa. Si una variación no puede encontrarse en nuestro HGDB, entonces ésta es clasificada como desconocida. Llegados a este punto, las variaciones conocidas se clasifican según el determinado cambio que originan.
4. **Conocimiento fenotípico**. Las variaciones que son clasificadas como conocidas pueden tener asociado un fenotipo. Para poder evaluar si un fenotipo está relacionado con una enfermedad específica, es necesario que se proporcione una publicación de investigación como prueba de confianza. Por estas razones, el esquema conceptual

describe la referencia bibliográfica que confirma el fenotipo para una determinada variación. En el contexto de este trabajo, las variaciones con un fenotipo patógeno son clasificadas como mutaciones, mientras que el resto pueden ser clasificadas como SNP o como variaciones sin un fenotipo negativo asociado.

5. **Creación del informe.** Toda la información obtenida se recopila formando un informe genético. Este informe contiene información sobre las variaciones encontradas: mutaciones, variaciones cuyo fenotipo no tiene una enfermedad asignada, SNPs o variaciones desconocidas. Para cada variación se proporciona la siguiente información: la posición en la que fue encontrada en la secuencia estudiada, su tipo (*Insertion*, *Deletion*, *Indel* o *Inversion*) y el número de nucleótidos que han sido eliminados o insertados, todo esto junto a su descripción en formato HGVS. Para aquellas mutaciones encontradas en HGDB se añade también la información sobre su fenotipo asociado y su bibliografía. Por último, el informe puede ser guardado como un documento de texto.

9.2. Arquitectura y clases

En cuanto a la arquitectura seguida para llevar a cabo la implementación del prototipo aquí expuesto se ha optado por la elección de una arquitectura multicapa. Este tipo de arquitectura permite una separación lógica de los procesos relacionados con la presentación de la herramienta, el proceso interno de dicha aplicación, y la gestión que ésta hace de la información. La aplicación de arquitecturas N-capa proporciona un modelo a los desarrolladores para conseguir llevar a cabo aplicaciones que sean flexibles y reutilizables. Esto conlleva a que los desarrolladores tan sólo tengan que modificar o añadir una capa en lugar de tener que reescribir la aplicación entera de nuevo para introducir un cambio. Teniendo en cuenta que en el dominio de la bioinformática, y en concreto en el de la genómica, se están llevando a cabo muchas investigaciones teniendo como consecuencia la creación de nueva información para el dominio, es importante que se facilite la forma en la que el sistema desarrollado pueda ser mantenido, modificado o ampliado (Ilustración 86).

Este prototipo ha sido implementado como una herramienta web dada la tendencia actual utilizada por empresas como 23andme [45]. La tecnología utilizada ha sido C# y ASP.NET.

Otro punto a tratar en este apartado es el de la traducción a partir del modelo conceptual, expuesto en capítulos anteriores, al modelo de clases. La Ilustración 87 muestra un breve resumen de las clases más importantes de la herramienta de análisis de variaciones que han sido implementadas, así como algunos de sus atributos y servicios. Es importante tener claro que el objetivo de este prototipo es encontrar todas las variaciones que pertenecen a una determinada secuencia de ADN, consecuencia directa de esta restricción son las diferencias existentes entre el modelo representado en anteriores capítulos y el diagrama de clases aquí expuesto.

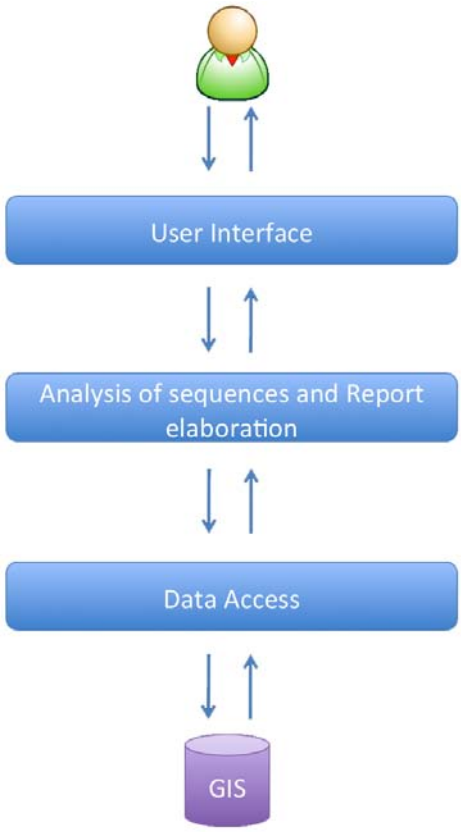

Ilustración 86 Arquitectura del prototipo

La entidad *Variation* es la clase principal y representa las variaciones que han sido encontradas después de haber llevado a cabo el análisis de la secuencia de ADN. Es por ello que tan sólo se analizaran las variaciones precisas, dejando fuera del objetivo las variaciones imprecisas. El atributo *type* de esta entidad indica el tipo de variación que se ha encontrado: inserción, borrado, indel o sustitución. Los servicios de esta clase están relacionados con el acceso y gestión de sus propios atributos.

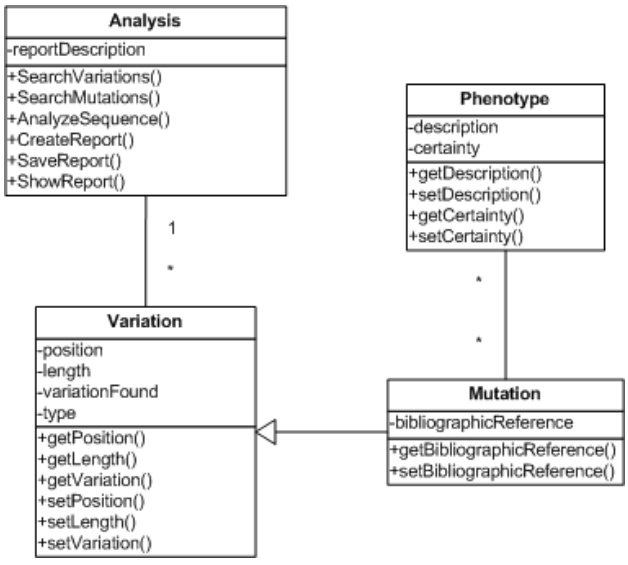

Ilustración 87 Resumen del diagrama de clases

La especialización *Mutation* caracteriza aquellas variaciones que tienen asociada algún tipo de enfermedad. Como se ha indicado a lo largo de este trabajo, para garantizar la fiabilidad de la relación entre una variación y su consecuente fenotipo es necesario un artículo que lo verifique. Por tanto a la especialización *Mutation* se le ha añadido un atributo *bibliographicReference*, en el que se guarda dicha referencia bibliográfica. En lugar de añadir una nueva clase para representar la propiedad de referencia bibliográfica se ha decidido añadir como atributo a esta especialización ya que este tipo de información tan sólo va a ser leída de la base de datos como una propiedad de la mutación y no va a ser gestionada en otro modo. De hecho, el método *setBibliographicReference* es privado y tan sólo es usado para proporcionar el valor al atributo a través del método de extracción de información de la base de datos.

La clase *Phenotype* representa la forma en la que se expresa una variación de ADN. Esta clase está enlazada a la entidad *Mutation*, ya que sólo las variaciones que son de tipo mutación tienen una expresión fenotípica.

Finalmente la clase *Analysis* es la clase que se encarga de llevar a cabo el análisis de las secuencias y crear el reporte final, con la respectiva información en formato HGVS de las variaciones encontradas y sus fenotipos asociados, que será mostrado al biólogo. La visibilidad de los métodos *SearchVariations*, *SearchMutations*, *AnalyzeSequence* y *ShowReport* es privada. El usuario, en esta fase del proyecto, sólo puede hacer uso de los servicios de creación y almacenamiento del reporte, responsables de llamar al resto internamente.

9.3. Análisis de secuencias

El proceso del análisis de las secuencias y extracción de las variaciones se hace en la clase *Analysis*. De todas las herramientas estudiadas en el capítulo de Alineamiento la que escogida para ser utilizada por el prototipo ha sido BLAT. Dicha elección ha sido llevada a cabo por la necesidad de realizar alineamientos locales en lugar de alineamientos globales, dado que las dos secuencias que se van a alinear en este prototipo son muy similares. La velocidad de escaneo que BLAT ofrece para los emparejamientos relativamente cortos, su tendencia a realizar alineamientos grandes y los diferentes formatos de salida de datos han sido las principales razones para su elección.

El formato de salida elegido para extraer la información de los datos es el PSL, y para llevar a cabo este proceso se utiliza la fórmula que se mostró en la Ilustración 48 en el capítulo de alineamiento, así como el algoritmo mencionado también en éste. Cabe recordar que esta forma de salida está limitada para encontrar inserciones y borrados, y tan sólo indica la existencia de indels. En este prototipo, en el caso de que se verifique la existencia de algún indel, se lanza de nuevo la herramienta BLAST con otro tipo de salida para encontrar mediante una búsqueda lineal este tipo de variaciones.

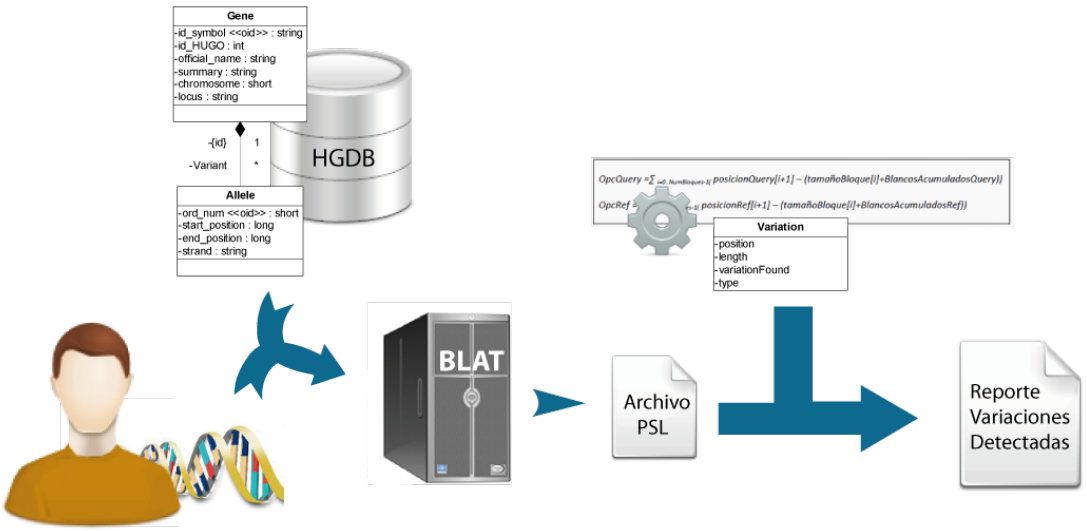

Ilustración 88 Proceso de alineación

Así pues en este apartado se lleva a cabo, tal y como se ve en la Ilustración 88, la recogida de la información dada por el usuario, el gen y la secuencia de ADN. Mediante la información del gen a estudiar, se extrae del Sistema de Información la secuencia de referencia que deberá compararse con la introducida por el usuario. Una vez obtenida, se pasa a lanzar la herramienta BLAST y se consigue el archivo de salida PSL. Éstos se analizan mediante las técnicas explicadas anteriormente, y sus resultados pasan a formar parte de la entidad *Variation* de modo que dicha información pueda ser manejada en procesos futuros por otro tipo de herramientas.

9.4. Filtrado de variaciones

Una vez se tienen las variaciones formateadas según la entidad *Variation*, la siguiente etapa a seguir es la del filtrado de variaciones para escoger a cuáles de ellas se les deberá adjuntar una expresión fenotípica y una referencia bibliográfica que lo constate. Como se ha comentado a lo largo de este trabajo esta búsqueda tan sólo se hará en aquellas variaciones precisas, es decir, aquellas de las cuales se tiene su posición concreta.

Sin embargo, antes de realizar la búsqueda de variaciones se debe tener en cuenta la existencia de posibles variaciones sinónimas, tal y como se observa en la Ilustración 89. Por tanto se pondrán en práctica los algoritmos explicados en el capítulo 7.1. Variaciones sinónimas

obteniendo de este modo para este tipo de variaciones el rango en el que deberá buscarse alguna variación del sistema de información.

Después de este paso lo que se tienen son las variaciones que se encuentran en el Sistema de información clasificadas por tipo y el resto de variaciones no encontradas.

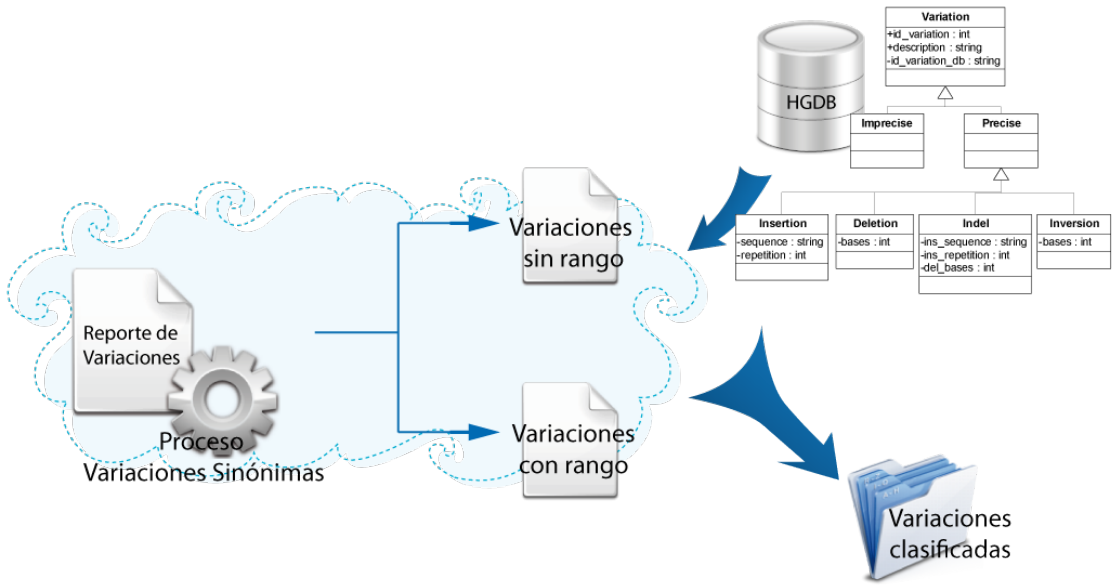

Ilustración 89 Proceso de clasificación de variaciones

9.5. Conocimiento fenotípico y Formato HGVS

En esta fase se tiene tan sólo en cuenta aquellas variaciones que han sido encontradas en el Sistema de Información y han sido clasificadas como precisas. De todas éstas tan sólo parar aquellas que sean clasificadas como mutaciones (Ilustración 90) será necesario buscar si tiene un fenotipo asociado y añadir a éste la bibliografía necesaria para verificar dicha relación. Esto es posible gracias al uso del modelo conceptual CSHG mediante el cual se puede estudiar estas correspondencias y obtener los resultados de una manera fácil y rápida.

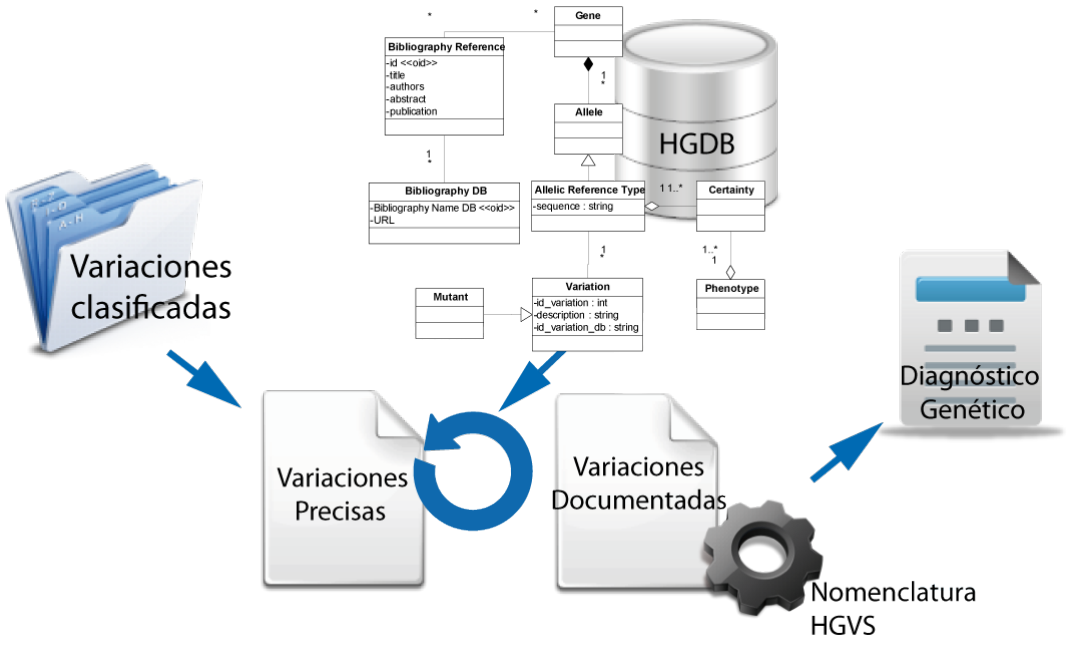

Ilustración 90 Proceso de introducción del fenotipo y descripción en formato de HGVS

Por último tan sólo queda recopilar toda la información que se ha obtenido del estudio para formar el informe que refleje el diagnóstico genético resultante. Las variaciones documentadas en dicho informe deben haber pasado por el proceso de formateado HGVS que se describió en el capítulo de Notación de variaciones

.

Conclusión

La aplicación de los Sistemas de Información en un dominio tan actual y relevante como lo es el de la genómica es poco frecuente. Desde esta perspectiva hablar del caos existente en los datos genómicos es algo inevitable. Existen toneladas de datos genómicos que se encuentran a disposición del público, cada uno con su correspondiente base de datos que está definida con un esquema concreto, un formato de datos determinado, de identificadores, etc. Por lo tanto, la integración de diferentes fuentes de datos se convierte en una tarea realmente complicada que a veces es simplemente imposible de llevar a cabo. Es necesidad urgente el realizar una correcta gestión de esta información, ya que la información científica disponible, los datos experimentales, el conocimiento sobre los productos génicos, la información sobre enfermedades, y muchos más datos relacionados con el ámbito de la genómica están creciendo día tras día y de manera desorbitada. Para comprender el conocimiento relevante que hay detrás de toda esta inmensa cantidad de datos, el modelado conceptual debería convertirse en la clave de los artefactos software.

La heterogeneidad de la información y de las estructuras de información en el dominio de la genómica es muy grande y presenta un gran impedimento a la hora de integrar y reutilizar bases de datos genómicas. Un esquema conceptual común es necesario, ya que esto reduce la redundancia de información que pueda haber en bases de datos autónomas, normaliza algunos aspectos del tratamiento de las bases de datos, como pueden ser las consultas SQL, y facilita la participación de la base de datos en subconjuntos, así como la duplicación de relaciones y su actualización dentro de un sistema de base de datos distribuido.

En esta tesis de máster se propone la utilización de Sistemas de Información en el dominio de la genómica, y más concretamente en el campo del diagnóstico genético, de modo que los problemas de heterogeneidad en este dominio queden resueltos.

La primera aportación de esta tesis es la definición formal de un GeIS. Esta idea no es nueva, sin embargo sí lo es el uso de modelos conceptuales que definan y organicen los datos genómicos de manera formal. En esta definición se indican las diferencias existentes entre un Sistema de Información normal y un Sistemas de información Genómico: (1) la necesidad de almacenar cantidades ingentes de datos como lo son las secuencias genómicas, (2) la desestructuración de la información existente en el campo y (3) la continua evolución del dominio.

Dado que el campo de la genética, al igual que el de la genómica, son dominios muy extensos esta tesis se ha centrado en el estudio de los diagnósticos genéticos. Gracias a este estudio preliminar se han detectado diferentes problemas a la hora de introducir los Sistemas de Información en este dominio.

La primera problemática existente viene dada por la dificultad a la hora de elegir qué herramienta usar para llevar a cabo un alineamiento de secuencias. Esto es debido a la cantidad de herramientas disponibles y a los criterios a los que cada una de ellas responde. Como segunda aportación, en este trabajo se han descrito los tres apartados que han de tenerse en cuenta a la hora de llevar a cabo dicha elección:

- El tipo de búsqueda que se desea realizar.
- El tiempo disponible para la obtención de resultados.
- Los formatos de salida ofertados.

Este último concepto es el que más problemas origina, debido a que cada herramienta devuelve la información resultante del estudio llevado a cabo según unos criterios y estándares propios. Además, cabe destacar que muchas veces no se tiene en consideración las necesidades de los biólogos a la hora de solventar los problemas, ya que estas herramientas están diseñadas desde el punto de vista del problema.

En esta tesis se ha llevado a cabo un estudio de las aplicaciones utilizadas para el análisis genético de secuencias, analizando sus pros y contras, así como sus formatos de salida, y ofreciendo una solución de cara a la implementación de un GeIS como tercera contribución.

La cuarta contribución de esta tesis es la incorporación de un nuevo concepto, el de variación sinónima. Debido a la baja variabilidad del código genético y a las diferentes técnicas de alineamiento disponibles, se puede dar el caso en que cambios de nucleótidos que referencian a distintas variaciones tienen como resultado una misma mutación. Esto es un problema de ambigüedad y no se dispone de una solución exacta. Por tanto, a causa de la existencia de este tipo de variaciones pueden surgir problemas de redundancia o pérdida de información. En esta tesis se ofrece una posible solución práctica a dicho problema.

En lo referente a la documentación de las variaciones viene existiendo un caos global, ya que cada biólogo las documenta según cree correcto. Actualmente existen unas guías de uso para realizar una nomenclatura estándar ade las variaciones, estas guías son las HGVS. Como quinta aportación de esta tesis de máster se ofrece un estudio de dichas guías, destacando los problemas encontrados a la hora de formalizar la descripción de las variaciones siguiendo dichas reglas. Se ofrece además una implementación práctica de la traducción automática de variaciones.

Como sexta aportación se ha implementado, a modo de prueba de concepto, un prototipo para llevar a cabo el informe de un diagnóstico genético realizado a partir de una muestra de ADN de un paciente. Para obtener los requisitos necesarios para llevar a cabo dicho prototipo se ha contado con la colaboración de la empresa IMEGEN.

Gracias a este trabajo se ha podido realizar un póster y un artículo corto *Facing the Challenges of Genome Information Systems: a Variation Analysis Prototype*, que fueron presentados en el CAiSE Forum del año 2010 en Hammamet, Túnez. Cabe destacar que la versión extendida de dicho artículo fue elegida entre todas las presentadas en dicha conferencia para formar parte del libro *Information Systems Evolution* publicado en 2011 por la editorial Springer.

Queda pues justificada la necesidad de la introducción de los Sistemas de Información al ámbito de la genómica, teniendo su base en el modelado conceptual. El caos de información existente en dicho ámbito lleva una considerable pérdida de tiempo en la realización de tareas biológicas. Los biólogos invierten mucho tiempo buscando información entre las diferentes fuentes de datos, cambiando el formato de la información encontrada para poder utilizarla en otras herramientas, etc. En otro ámbito traduciríamos tiempo por dinero, sin embargo en este dominio el tiempo se puede traducir en vidas. La necesidad de optimizar las tareas que

desarrollan los expertos genómicos es indiscutible, puesto que esto puede llevar a agilizar la creación de una nueva vacuna o la erradicación de alguna enfermedad.

Parte de este problema se basa en que la visión de biólogos e ingenieros informáticos no está unida, y el espacio interdisciplinar existente es en algunos casos mayor del deseado. Es por ello que el hecho de introducir el modelado conceptual en este ámbito es una apuesta segura, pues acerca ambas disciplinas y permite una colaboración mayor y más exitosa.

Teniendo este trabajo de base y habiendo recopilado el conocimiento necesario en el ámbito de la bioinformática, los siguientes pasos a seguir serán la transformación del prototipo en una herramienta funcional que solvente los problemas expuestos y las limitaciones encontradas en la implementación de éste. Del mismo modo, otras ramas de investigación que se podrán seguir son:

- Estudio del diseño de un informe genético, generado mediante modelos, teniendo en cuenta las necesidades del usuario. De este modo se permitiría al usuario elegir la estructura del informe, la información representada, etc.
- Estudio de la estandarización de los informes genéticos. El uso de un estándar en este campo podría ser muy útil teniendo en cuenta su posible introducción en el ámbito de la medicina personalizada y su relevancia actual.
- Estudio del diagnóstico genético mediante alineamiento de varias secuencias de ADN.
- Estudio del diagnóstico genómico de un paciente. Dentro de éste se englobaría el diagnóstico genético que sería extraíble a partir del genoma del paciente.

Como último punto se debe mencionar que este trabajo ha sido la base para la realización de una segunda tesis de máster llevada a cabo por Mª José Villanueva, TITULO.

Bibliografía

1. Venter, J., et al., *The sequence of the human genome.* Science, 2001. **291**(5507): p. 1304-1351.
2. Collins, F.S., Green, Eric D.,Guttmacher, Alan E.,Guyer, Mark S., *A vision for the future of genomics research.* Nature, 2003. **422**(6934): p. 835-847.
3. Stevens, R., et al., *A classification of tasks in bioinformatics.* Bioinformatics, 2001. **17**(2): p. 180.
4. Hamosh, A., et al., *Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders.* Nucleic Acids Research, 2005. **33**(suppl 1): p. D514.
5. Stenson, P., et al., *Human gene mutation database (HGMD®): 2003 update.* Human mutation, 2003. **21**(6): p. 577-581.
6. Stevens , R., Goble , C. ,Baker, P. , Brass, A., *A classification of tasks in bioinformatics.* Bioinformatics, 2001. **17**: p. 180.
7. Zachman, J., *A framework for information systems architecture.* IBM systems journal. **26**(3): p. 276-292.
8. Mardis, E., *Next-generation DNA sequencing methods.* Annual review of genomics and human genetics, 2008. **9**(1): p. 387.
9. Gilbert, D.G., *Eugenes: a eukaryote genome information system.* Nucleic Acids Research, 2002. **30**: p. 145–148.
10. Beaudet, A. and L. Tsui, *A suggested nomenclature for designating mutations.* Human mutation, 1993. **2**(4): p. 245-248.
11. Beutler, E., *The designation of mutations.* American journal of human genetics, 1993. **53**(3): p. 783.
12. Olivé, A., *Conceptual modeling of information systems*. 2007: Springer-Verlag New York Inc.
13. Fowler, M. and K. Scott, *UML distilled: a brief guide to the standard object modeling language*. 2000: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
14. Chen, P., *The entity-relationship model—toward a unified view of data.* ACM Transactions on Database Systems (TODS), 1976. **1**(1): p. 9-36.
15. Pastor, O., *Conceptual Modeling Meets the Human Genome.* Conceptual Modeling-ER 2008, 2008: p. 1-11.
16. Pastor, O., Levin, AM., Casamayor, J.C., Celma, M., Virrueta, A. , Eraso, L. , *Enforcing Conceptual Modeling to Improve the Understanding of Human Genome.* 6th Data Integration in the Life Sciences Workshop (DILS09) Manchester, UK, 2009.
17. Den Dunnen, J. and S. Antonarakis, *Nomenclature for the description of human sequence variations.* Human genetics, 2001. **109**(1): p. 121-124.
18. Richesson, R. and J. Turley, *Conceptual models: Definitions, construction, and applications in public health surveillance.* Journal of Urban Health, 2003. **80**: p. 128-128.
19. Lodish, H. and S.L. Zipursky, *Molecular cell biology.* Biochemistry and Molecular Biology Education, 2001. **29**: p. 126-133.
20. Shah, S., et al., *Atlas – a data warehouse for integrative bioinformatics.* BMC bioinformatics, 2005. **6**(1): p. 34.
21. Ng, P. and S. Henikoff, *SIFT: Predicting amino acid changes that affect protein function.* Nucleic Acids Research, 2003. **31**(13): p. 3812.
22. Tatusovaa, T.A., Maddena, T. L., *BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences*, in *FEMS Microbiology Letters*. 1999. p. 247-250.
23. Day, I., *dbSNP in the detail and copy number complexities.* Human Mutation, 2010. **31**(1): p. 2-4.

24. Bairoch, A., et al., *The universal protein resource (UniProt).* Nucleic Acids Research, 2005. **33**(Database Issue): p. D154.
25. Sayers, E., et al., *Database resources of the national center for biotechnology information.* Nucleic acids research, 2009.
26. Ramensky, V., P. Bork, and S. Sunyaev, *Human non synonymous SNPs: server and survey.* Nucleic acids research, 2002. **30**(17): p. 3894.
27. den Dunnen, J. and S. Antonarakis, *Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion.* Human mutation, 2000. **15**(1): p. 7-12.
28. Flicek, P., et al., *Ensembl 2011.* Nucleic acids research, 2010.
29. Grumbine, R., *The Natural History of the UC Santa Cruz Campus.* Western North American Naturalist, 2010. **70**(1): p. 130-131.
30. Venselaar, H., et al., *Protein structure analysis of mutations causing inheritable diseases. An e-Science approach with life scientist friendly interfaces.* BMC Bioinformatics. **11**(1): p. 548.
31. Newman, H., M. Ellisman, and J. Orcutt, *Data-intensive e-science frontier research.* Communications of the ACM, 2003. **46**(11): p. 68-77.
32. Dowell, R.D., Jokerst, R.M., Day, A., Eddy, S.R.,Stein, L., *The Distributed Annotation System.* BMC Bioinformatics, 2001. **2**.
33. Berman, H., et al., *The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data.* Nucleic Acids Research, 2006.
34. Hekkelman, M., et al., *WIWS: a protein structure bioinformatics Web service collection.* Nucleic Acids Research. **38**(suppl 2): p. W719.
35. Wildeman, M., et al., *Improving sequence variant descriptions in mutation databases and literature using the Mutalyzer sequence variation nomenclature checker.* Human mutation, 2008. **29**(1): p. 6-13.
36. Fokkema, I., J. den Dunnen, and P. Taschner, *LOVD: Easy creation of a locus specific sequence variation database using an ìLSDB in a boxî approach.* Human Mutation, 2005. **26**(2): p. 63-68.
37. van der Kroon, M., et al., *Mutational data loading routines for human genome databases: the BRCA1 case.* 2009. **4**: p. 291-312.
38. Lereu, I., *Diseño e implementación de un entorno de carga de datos genómicos para el gen NF1 centrado en esquemas conceptuales.* PROYECTO FIN DE CARRERA II-B-DSIC.
39. Kent, W., *BLAT - the BLAST-like alignment tool.* Genome research, 2002. **12**(4): p. 656.
40. Watson, J. and F. Crick, *A structure for deoxyribose nucleic acid.* A century of Nature: twenty-one discoveries that changed science and the world, 2003: p. 82.
41. Venter, J., et al., *The sequence of the human genome.* Science, 2001. **291**(5507): p. 1304.
42. Hamosh, A., et al., *Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders.* Nucleic Acids Research, 2005. **33**(suppl 1): p. D514.
43. Stenson, P.D., et al., *The human gene mutation database: 2008 update.* Genome medicine, 2009. **1**(1): p. 13.
44. Pruitt, K.D., T. Tatusova, and D.R. Maglott, *NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins.* Nucleic Acids Research, 2006.
45. 23andme. Available from: https://www.23andme.com/.

ANEXO: Código del prototipo

Inicio.aspx

```
namespace prototipo {


    public partial class inicio {

        /// <summary>
        /// Control Head1.
        /// </summary>
        /// <remarks>
        /// Campo generado automáticamente.
        /// Para modificarlo, mueva la declaración del campo del
archivo del diseñador al archivo de código subyacente.
        /// </remarks>
        protected global::System.Web.UI.HtmlControls.HtmlHead Head1;

        /// <summary>
        /// Control form1.
        /// </summary>
        /// <remarks>
        /// Campo generado automáticamente.
        /// Para modificarlo, mueva la declaración del campo del
archivo del diseñador al archivo de código subyacente.
        /// </remarks>
        protected global::System.Web.UI.HtmlControls.HtmlForm form1;
    }
}
```

Inicio.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace prototipo
{
    public partial class inicio : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Mutaciones.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="mutaciones.aspx.cs" Inherits="prototipo.mutaciones" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <meta http-equiv="content-type" content="text/html; charset=utf-8"
/>
    <title>Diagen</title>
    <meta name="keywords" content="" />
    <meta name="description" content="" />
    <link href="Stylesheet1.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
        <div class="logosFijos">

        </div>
        <div id="header">
            <div id="logo">
                <h1>
                    diagen</h1>
                <ul>
                    <li><a href="inicio.aspx" accesskey="1"
title="">Home</a></li>
                    <li><a href="mutaciones.aspx" accesskey="2"
title="">Reports</a></li>
                </ul>
                <img src="images/losDosLogos3.png" alt="Centro de
Investigación en Métodos de Producción de Software & Universidad
Politecnica de Valencia"/>
            </div>
        </div>
        <div class="logosFijos"></div>
        <hr />
        <div id="page">
            <div id="bg">
                <div id="content">
                    <div class="post" style="padding-top: 57px;">
                        <h2 class="title">
                            Prototype</h2>
                        <div class="entry">
                            <b>Selected Gene </b>
                            <asp:DropDownList ID="ddl1" runat="server"
AutoPostBack="True">
                            </asp:DropDownList>
                            <br />
                            <br />
                            <b>Reference Sequence</b>
                            <br />
                            <asp:TextBox ID="tbSecRef" runat="server"
Height="170px" ReadOnly="True" Width="340px"
                                TextMode="MultiLine">Reference
Sequence</asp:TextBox>
                            <div class="botonesLargos">
                                <asp:Button ID="bMostrarSecRef"
OnClick="bMostrarSecRef_Click" runat="server" Text="View Reference
Sequence"
```

```
                                    CssClass="botonesLargos" />
                            </div>
                            <br />
                            <br />
                            <b>Compared Sequence</b>
                            <br />
                            <asp:TextBox ID="tbSecComp" runat="server"
Height="170px" Width="340px" TextMode="MultiLine"
                                BackColor="#F1EADA">Write the sequence
that you want to compare...</asp:TextBox>
                            <br />
                            <br />
                            <b>Load FASTA file containing the sequence
to be compared</b>
                            <br />
                            <input id="uplTheFile" type="file"
runat="server" lang="en" /><br />
                            <br />
                            <br />
                        </div>
                    </div>
                </div>
                <div id="sidebar">
                    <div id="about-box">
                        <br />
                        <br />
                        <asp:Button ID="bInforme" runat="server"
CssClass="botones" Text="Show Report"
                            OnClick="bInforme_Click" />
                        <asp:Button ID="bLimpiar" CssClass="botones"
runat="server" Text="Clean" OnClick="bLimpiar_Click" />
                        <asp:Button ID="bSave" CssClass="botones"
runat="server" Text="Save" OnClick="bSave_Click" />
                        <br />
                        <br />
                        <h2>
                            Report</h2>
                        <asp:Label  Width="450px" ID="lbResultado"
runat="server">Here the report will be showed...</asp:Label>
                        <br />
                    </div>
                </div>
                <div style="clear: both;">
                     </div>
            </div>
        </div>
        <!-- end page -->
        <hr />
        <div id="footer">
            <p>
                © Version 1.0 February 2010 All rights reserved.
Centro de Métodos en
                Métodos de Producción de Software (Universidad
Politecnica de Valencia).</p>
        </div>
    </form>
</body>
</html>
```

Mutaciones.aspx.cs

```
using System;
using System.Collections;
using System.Data;
using System.Data.OracleClient;
using System.Diagnostics;
using System.IO;

namespace prototipo
{
    public partial class mutaciones : System.Web.UI.Page
    {
        bool hayInyDel = false;
        ArrayList variaciones = new  ArrayList();
        private static string idGen;
        private static int idAllele;
        /// <summary>
        /// Es el número de nucleótidos del los indels encontrados
        /// que tienen el mismo número de nucleótidos eliminados e
        /// insertados.
        /// </summary>
        private int nucIndelIguales = 0;
        /// <summary>
        /// Indica el número de indels que ha encontrado en los que
        /// el número de nucleótidos eliminados es igual al número
        /// de nucleótidos
        /// </summary>
        private int numIndelDif = 0;


        protected void Page_Load(object sender, EventArgs e)
        {
             if(!Page.IsPostBack)
             {
                 //cargar el dropdownlist
                 ArrayList genes = ObtenerNombreGenes();

                 ddl1.DataSource = genes;
                 ddl1.DataBind();
             }
        }

        /// <summary>
        /// Muestra la secuencia de referencia del gen seleccionado.
        /// Por ahora tan sólo la del NF1
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        protected void bMostrarSecRef_Click(object sender, EventArgs
e)
        {
           RecogerSecuenciaReferencia();

        }


        /// <summary>
        /// Redacta el informe de lo que ha encontrado y lo muestra
por pantalla
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
```

```
        protected void bInforme_Click(object sender, EventArgs e)
        {
            bool esFichero = false;
            {

                string sample =
@"C:\\Proyectos\Publicaciones\prototipo\Documentos\Sample.txt";

                //preguntar si realmente ha seleccionado un archivo:
                if (null != uplTheFile.PostedFile &&
uplTheFile.PostedFile.FileName != string.Empty)
                {
                    try
                    {
                        esFichero = true;
                        //Es en la propiedad "postedFile" donde se
guarda la ruta del archivo
                        //Con el método "SaveAs" lo guardas en la ruta
que tú el digas
                        uplTheFile.PostedFile.SaveAs(sample);
                    }
                    catch (Exception ex)
                    {
                        lbResultado.Text = "Error al guardar el
archivo: <b>" +
                          uplTheFile.PostedFile.FileName + "</b><br>"
+ ex.ToString();

                    }
                }

                // recoger secuencia de referencia
                string secRef = RecogerSecuenciaReferencia();

                // path del documento BD

                string path =
@"C:\Proyectos\Publicaciones\prototipo\Documentos\DataBase.fa";

                //Creamos el fichero que nos servirá como BD para el
uso del BLAT
                CreateFileBD(path, secRef);

                StreamWriter sw;

                if ((tbSecComp.Text.Trim() != string.Empty &&
tbSecComp.Text != "Escriba la secuencia a comparar...") || esFichero )
                {
                    tbSecComp.ForeColor = System.Drawing.Color.Black;
                    string secComp;

                    if(!esFichero)
                    {
                        sample = CreaFileSample();
                        secComp = tbSecComp.Text;
                    }
                    else
                    {
                        StreamReader reader = new
StreamReader(sample);
                         secComp = "";
```

```
                        while(reader.ReadLine()!= null)
                        { secComp += reader.ReadLine(); }
                        reader.Close();

                    }
                    string salida =
@"C:\Proyectos\Publicaciones\prototipo\Documentos\Salida";
                    // Lanzar el BLAT
                    BLAT(path, sample, salida, false);

                    // Analizar el fichero generado
                    // nos quedamos con la última línea
                    StreamReader sr = new StreamReader(salida);
                    string leido = sr.ReadLine();
                    sr.ReadLine();
                    sr.ReadLine();
                    sr.ReadLine();
                    sr.ReadLine();
                    leido = sr.ReadLine();
                    sr.Close();
                    //represnta el valor que devuelve blast
                    bool ok = true;

                    if (leido != null)
                    {
                        // mirar que tipo de cambio hay, si es que lo
hay
                        string[] datos = leido.Split('\t');

                        #region [missmatch = INDELS de igual longitud]

                        if (datos[1] != "0")
                        {
                            // Se han encontrado indels en los que el
número
                            // de nucleótidos eliminados e insertados
es el mismo.
                            nucIndelIguales =
Convert.ToInt32(datos[1]);
                            salida =
@"C:\Proyectos\Publicaciones\prototipo\Documentos\SalidaBlast";
                            // Lanzar BLAT en forma de BLAST
                            BLAT(path, sample, salida, true);

                            // Hay que llamar al método que nos dae
las variaciones para
                            IndelSameLenght(salida, idGen, idAllele,
variaciones, tbSecRef.Text, secComp);// tbSecComp.Text);
                        }
                        #endregion

                        /*
                         * 17 NUM BLOQ
                         * 18 TAMAÑo
                         * 19 QUERYºººº
                         * 20 REf
                         */
                        // número de bloques en los apartados
siguientes
                        int numBloques = Convert.ToInt32(datos[17]);
```

```
                        string[] posicionesQuery =
datos[19].Split(',');
                        string[] posicionesRef = datos[20].Split(',');
                        string[] tamanyoBloques =
datos[18].Split(',');
                        // variables que guardan el número de
nucleótidos
                        // afectados y/o espacios en blanco de cada
cadena
                        int blancosQacc = 0;
                        int blancosRacc = 0;
                        // variables auxiliares que actúan de
acumuladores
                        // de los sumatorios que se deberían hacer de
las posiciones
                        // desde 0 a i
                        int auxacc =
Convert.ToInt32(tamanyoBloques[0]);
                        //int auxRacc = 0;
                        // Recorrer los bloques para obtener qué
variaciones tenemos
                        // y saber sus tipos
                        for(int i = 0; i < numBloques-1; i++)
                        {
                            int opcQ =
Convert.ToInt32(posicionesQuery[i+1]) - (auxacc + blancosQacc);
                            int opcR = Convert.ToInt32(posicionesRef[i
+ 1]) - (auxacc + blancosRacc);

                            if(opcQ == 0)
                            {
                                #region [DELECION]
                                if (opcR != 0)
                                {
                                    Variacion variacion = new
Variacion();
                                    variacion.IdGen = idGen;
                                    variacion.IdAllele = idAllele;

                                    // rellenar el campo de la cadena
de referencia
                                    if (esFichero)
                                        variacion.CadenaQuery = "";
                                    else
                                        variacion.CadenaQuery =
secComp;//  tbSecComp.Text;
                                    // rellenar el campo de la
posición donde se produce el cambio en la secuencia de referencia
                                    variacion.Posicion =
Convert.ToInt32(posicionesRef[i + 1]);
                                    // rellenar el campo de la cadena
de referencia
                                    variacion.CadenaRef =
tbSecRef.Text;
                                    // rellenar el campo de la cadena
de cambio
                                    variacion.CadenaEliminada =

secRef.Substring(variacion.Posicion - 1, opcR);
```

```
                                    variacion.Tipo =
(int)Variacion.TipoVariacion.Deletion;
                                    variacion.NumBasesDel = opcR;
                                    variacion.NumBasesIns = 0;
                                    // añadir la variación a la
colección
                                    variaciones.Add(variacion);

                                    //aumentar el acumulador
                                    blancosRacc += opcR;
                                    auxacc +=
Convert.ToInt32(tamanyoBloques[i + 1]);
                                }
                                #endregion
                            }
                            else
                            {
                                #region [INSERCION]
                                if (opcR == 0)
                                {
                                    Variacion variacion = new
Variacion();
                                    variacion.IdGen = idGen;
                                    variacion.IdAllele = idAllele;

                                    // rellenar el campo de la cadena
de referencia
                                    variacion.CadenaQuery = secComp;//
tbSecComp.Text;
                                    // rellenar el campo de la
posición donde se produce el cambio en la secuencia de referencia
                                    variacion.Posicion =
Convert.ToInt32(posicionesRef[i + 1]);
                                    // rellenar el campo de la cadena
de referencia
                                    variacion.CadenaRef =
tbSecRef.Text;
                                    // rellenar el campo de la cadena
de cambio
                                    variacion.CadenaInsertada =

secComp.Substring(variacion.Posicion, opcQ);

//secComp.Substring(variacion.Posicion - 1, opcQ);
                                    variacion.Tipo =
(int)Variacion.TipoVariacion.Insertion;
                                    variacion.NumBasesDel = 0;
                                    variacion.NumBasesIns = opcQ;
                                    // añadir la variación a la
colección
                                    variaciones.Add(variacion);


                                    //aumentar el acumulador
                                    blancosQacc += opcQ;
                                    auxacc +=
Convert.ToInt32(tamanyoBloques[i + 1]);
                                }
                                #endregion
                                #region [INDEL DIF]
                                else
```

```
                                {
                                    Variacion variacion = new
Variacion();
                                    variacion.IdGen = idGen;
                                    variacion.IdAllele = idAllele;

                                    // rellenar el campo de la cadena
de referencia
                                    variacion.CadenaQuery = secComp;
// tbSecComp.Text;
                                    // rellenar el campo de la
posición donde se produce el cambio en la secuencia de referencia
                                    variacion.Posicion =
Convert.ToInt32(posicionesRef[i + 1]);
                                    // rellenar el campo de la cadena
de referencia
                                    variacion.CadenaRef =
tbSecRef.Text;
                                    // rellenar el campo de la cadena
de cambio
                                    variacion.CadenaInsertada =

secComp.Substring(variacion.Posicion - 1, opcQ);
                                    variacion.CadenaEliminada =
secRef.Substring(variacion.Posicion - 1, opcR);
                                    variacion.Tipo =
(int)Variacion.TipoVariacion.InDel;
                                    variacion.NumBasesDel = opcR;
                                    variacion.NumBasesIns = opcQ;
                                    // añadir la variación a la
colección
                                    variaciones.Add(variacion);

                                    //aumentar el acumulador
                                    blancosQacc += opcQ;
                                    blancosRacc += opcR;
                                    auxacc +=
Convert.ToInt32(tamanyoBloques[i + 1]);
                                }
                                #endregion
                            }
                        }

                        lbResultado.Text = "";
                        // Contrastar las variaciones con la BD
                        if (variaciones.Count == 0)
                        {
                            if (!ok)
                            {
                                lbResultado.Text = "No hay ninguna
variación. ";
                            }
                            else
                            {
                                lbResultado.Text =
                                    "Se ha producido un error.
Compruebe que la secuencia introducida para ser comparada es
correcta.";
                            }
```

```
                        }
                        else
                        {
                            lbResultado.Text += "Encontradas " +
variaciones.Count + " variaciones en el gen " +
((Variacion)variaciones[0]).IdGen + ". ";
                            lbResultado.Text += "<br>";

                            // BUSQUEDA EN BD
                           foreach (Variacion variacion in
variaciones)
                            {
                                string fenotipo;
                                //Buscar la variación en lal base de
datos
                                VariacionBD(variacion);
                                // Si no es mutación no necesita que
se busque un fenotipo, puesto que no tendrá
                                // ninguno asignado...

                                if (variacion.IdVariation != -1)
                                {

                                    RellenarTbResultado(variacion);

                                    if (variacion.Efecto ==
(int)Variacion.SpecializationEffect.Mutant)
                                    {
                                        //Es mutación -> hay que
encontrar su fenotipo

                                        fenotipo =
Phenotype(variacion.IdVariation);
                                        if (fenotipo == string.Empty)
                                        {
                                            fenotipo = "Mutant
variation. It doesn't have assigned phenotype yet. ";
                                        }
                                        lbResultado.Text += " And wich
phenotype is " + fenotipo + ". ";
                                    }
                                    else
                                    {
                                        lbResultado.Text += "
Variation type ";

                                        switch(variacion.Efecto)
                                        {
                                           case
(int)Variacion.SpecializationEffect.NeutralPolomorphism:
                                                lbResultado.Text += "
polimorphism. ";

                                                break;


                                           case
(int)Variacion.SpecializationEffect.UnkownConsequence:
                                                lbResultado.Text += "
unkown consequence. ";

                                                break;
                                        }
                                    }
```

```
                                    // Añadir las referencias si las
tiene
                                    if (variacion.Url != string.Empty)
                                    {
                                        //TODO: descomentar lo
comentado. Poner el título en lugar de la url
                                        lbResultado.Text += "<br>
Reference: <a href=\"" + variacion.Url + "\">" + variacion.Url +
"</a>";//+ variacion.NombreUrl + "</a>";
                                    }
                                    lbResultado.Text += "<br>";

                                }
                                else
                                {
                                    lbResultado.Text += "<br>
Variation not found in the database. ";
                                    RellenarTbResultado(variacion);
                                }

                            }
                        }


                    }
                }

            }
        }

        /// <summary>
        /// Crea el fichero que se utilizará como fichero de ejemplo
        /// que se contrastará contra el fichero BD mediante el BLAT.
        /// Se crea con la cadena que recoje por pantalla.
        /// </summary>
        /// <returns></returns>
        private string CreaFileSample()
        {
            StreamWriter sw;
            string secComp = tbSecComp.Text;
               string sample =
@"C:\Proyectos\Publicaciones\prototipo\Documentos\Sample.txt";
            sw = new StreamWriter(sample);
            sw.WriteLine("> sequence to compare of the "+
ddl1.SelectedValue+" gene. " );
            sw.WriteLine(secComp);
            sw.Close();
            return sample;
        }

        /// <summary>
        /// Crea el fichero en la ruta path que se utiliza como BD por
        /// el programa BLAT con la secuencia de referencia que hay
        ///  en secRef. Si el fichero existe, lo elimina primero.
        /// </summary>
        /// <param name="path">path donde se creará el fichero BD
</param>
        /// <param name="secRef">secuencia de referencia</param>
        private void CreateFileBD(string path, string secRef)
        {
            // eliminar el fichero si existe
```

```
            if (File.Exists(path))
            {
                File.Delete(path);
            }

            // Eliminación de ficheros viejos
            DriveInfo di =
                 new
DriveInfo(@"C:\Proyectos\Publicaciones\prototipo\Documentos");
            DirectoryInfo dirInfo = new
DirectoryInfo(@"C:\Proyectos\Publicaciones\prototipo\Documentos");//
di.RootDirectory;
            FileInfo[] fileNames = dirInfo.GetFiles("*.*");
            foreach (FileInfo fi in fileNames)
            {
                if (fi.Name.Split('.')[0] == "DataBase")
                {
                     string pathFich =
@"C:\Proyectos\Publicaciones\prototipo\Documentos\" + fi.Name;
                    File.Delete(pathFich);
                }
            }

            // creación del fichero que funcionará como BD
            StreamWriter sw = new StreamWriter(path);
            sw.WriteLine(">DataBase");
            sw.WriteLine(secRef);
            sw.Close();
        }

        /// <summary>
        /// lanzamiento del BLAT
        /// </summary>
        /// <param name="secRef">nombre y ruta del fichero del fichero
que contiene la secuencia de referencia</param>
        /// <param name="secSample">nombre y ruta del fichero del
fichero que contiene la secuencia a comparar</param>
        /// <param name="salida">nombre y ruta del fichero de salida
que se creará</param>
        /// <param name="blast">true -> salida en modo BLAST , false -
> salida en modo BLAT</param>
        private void BLAT(string secRef, string secSample, string
salida, bool blast)
        {
            Process compiler = new Process();
            compiler.StartInfo.FileName = @"C:\blatSuite\blat.exe";
            if(blast)
                compiler.StartInfo.Arguments = secRef + " " +
secSample + " " + salida + " -out=blast"; //  C:\blatSuite\Salida";
            else
                compiler.StartInfo.Arguments = secRef + " " +
secSample + " " + salida;
            compiler.StartInfo.UseShellExecute = false;
            compiler.StartInfo.RedirectStandardOutput = true;
            compiler.Start();
            compiler.WaitForExit();

        }

        /// <summary>
        /// Mostrado de los resultados por pantalla
```

```
        /// </summary>
        /// <param name="variacion"></param>
        private void RellenarTbResultado(Variacion variacion)
        {
            if (variacion.IdVariation != -1)
                lbResultado.Text += "<br>Variation found. ";
            lbResultado.Text += "<br>Sequence ";
            switch (variacion.Tipo)
            {
                case (int) Variacion.TipoVariacion.Deletion:
                    lbResultado.Text += "deleted  " +
variacion.CadenaEliminada + " in the position " + variacion.Posicion
                        + " of the gene " + variacion.IdGen + ".
Deletion type. ";
                    break;
                case (int) Variacion.TipoVariacion.InDel:
                    lbResultado.Text += " inserted  " +
variacion.CadenaInsertada +
                        ", deleted " + variacion.CadenaEliminada +
                        " in the position " + variacion.Posicion + "
of the gene " + variacion.IdGen +  ". InDel type. ";
                    break;
                case (int) Variacion.TipoVariacion.Insertion:
                    lbResultado.Text += "inserted " +
variacion.CadenaInsertada + " in the posistion " + variacion.Posicion
                        + " of the gene " + variacion.IdGen + ".
Insertion type. ";
                    break;
                case (int) Variacion.TipoVariacion.Inversion:
                    lbResultado.Text += "inversion " + " in the
position " + variacion.Posicion + " of the gene " + variacion.IdGen +
". Inversion type. ";
                    break;
            }

            if(variacion.Efecto != -1)
            {
                lbResultado.Text += " Expressing its effect as ";
                switch (variacion.Efecto)
                {
                    case (int)Variacion.SpecializationEffect.Mutant:
                        lbResultado.Text += "mutation. ";
                        break;
                    case
(int)Variacion.SpecializationEffect.NeutralPolomorphism:
                        lbResultado.Text += "neutral polimorphism. ";
                        break;
                    case
(int)Variacion.SpecializationEffect.UnkownConsequence:
                        lbResultado.Text += "unknown consequence. ";
                        break;
                }

            }
            lbResultado.Text += "<br>";
        }

        /// <summary>
        /// devuelve un string con la secuencia de referencia
        /// hay que buscar
        /// </summary>
```

```
        /// <returns></returns>
        private string RecogerSecuenciaReferencia()
        {
            // cadena de conexión
            string genomadb = "Data Source=genoma.dsic.upv.es;User
ID=hugo_v2;Password=hugo;Unicode=True";
            OracleConnection conn = new OracleConnection(genomadb);

            // abrir conexión
            conn.Open();

            // query
            string sql =
                "select allref.sequence, al.id_gene , al.allele_num
from ALLELIC_REFERENCE_TYPE allref, allele al where al.allele_num =
allref.allele_num and al.id_gene = '" + ddl1.SelectedValue + "'";
            OracleCommand cmd = new OracleCommand(sql, conn);
            cmd.CommandType = CommandType.Text;

            // lector
            OracleDataReader dr = cmd.ExecuteReader();
            dr.Read();

           // recoger el identificador de la secuencia
            idGen = dr.GetString(1);

            // recoger el identificador del alelo
            idAllele = dr.GetInt32(2);

            // recoger la secuencia leída
            tbSecRef.Text = dr.GetString(0);


            conn.Close();
            conn.Dispose();

            return tbSecRef.Text;
        }

        /// <summary>
        /// rellena los campos del identificador de lal variación
        /// y el del tipo de efecto que tiene.
        /// </summary>
        /// <param name="var"></param>
        /// <returns></returns>
        private void  VariacionBD(Variacion var)
        {
            // cadena de conexión
            string genomadb = "Data Source=genoma.dsic.upv.es;User
ID=hugo_v2;Password=hugo;Unicode=True";

            //bool encontrado = false;
            OracleConnection conn = new OracleConnection(genomadb);

            // abrir conexión
            conn.Open();
            //TODO: problema con CLOB -> preguntar a JC

            string tipo = string.Empty;
```

```
            string sql = "select p.id_variation,
v.specialization_effect from PRECISE p, Variation v where
v.id_allele_num_rt= '" + var.IdAllele +
                "' and p.position =" + var.Posicion + " and
p.id_variation = v.id_variation ";

            switch(var.Tipo)
            {
                case (int)Variacion.TipoVariacion.Insertion:
                    tipo = "IS";
                    sql += " and TYPE = '" + tipo + "' and
TO_CHAR(INS_SEQUENCE) = '" + var.CadenaInsertada + "'";
                    break;
                case (int)Variacion.TipoVariacion.Deletion:
                    tipo = "DE";
                    sql += " and TYPE = '" + tipo + "' and NUM_BASES =
'" + var.NumBasesDel/*var.CadenaCambio.Length */+ "'";
                    break;
                case (int)Variacion.TipoVariacion.InDel:
                    tipo = "ID";
                    sql += " and Type = '" + tipo + "' and
TO_CHAR(INS_SEQUENCE) = '" + var.CadenaInsertada +
                        "' and NUM_BASES = '" + var.NumBasesDel +
"'";
                    break;
                case (int)Variacion.TipoVariacion.Inversion:
                    tipo = "IN";
                    break;

            }

            OracleCommand cmd = new OracleCommand(sql, conn);
            cmd.CommandType = CommandType.Text;
            int idVariation = -1;
            string fenotipo = string.Empty;
            // lector
            OracleDataReader dr = cmd.ExecuteReader();
            try
            {
                dr.Read();

                idVariation = dr.GetInt32(0);
                var.IdVariation = idVariation;
                string efecto = dr.GetString(1);
                switch(efecto)
                {
                    case "M":
                        var.Efecto =
(int)Variacion.SpecializationEffect.Mutant;
                        break;
                    case "N":
                        var.Efecto =
(int)Variacion.SpecializationEffect.NeutralPolomorphism;
                        break;
                    case "U":
                        var.Efecto =
(int)Variacion.SpecializationEffect.UnkownConsequence;
                        break;
                }

                conn.Close();
```

```
                conn.Dispose();

            }
            catch(Exception ex )
            {
                conn.Close();
                conn.Dispose();

                // No se ha encontrado variación en la BD
                // Mirar las posibles...
                PosiblesVariaciones(var);


            }

            #region[carga URL]
            /// Si hemos encontrado la variación en la BD
            /// entonces buscaremos si tiene una url asociada

            if (var.IdVariation != -1)
            {
                // cadena de conexión
                string genomadb1 = "Data
Source=genoma.dsic.upv.es;User ID=hugo_v2;Password=hugo;Unicode=True";

                //bool encontrado = false;
                OracleConnection conn1 = new
OracleConnection(genomadb1);

                // abrir conexión
                conn1.Open();

                sql = "select  b.url, bd.title from
bibliography_db_location b, bibliography_reference bd " +
                      "where b.ID_BIB_REF =(select id_bib_ref from
reference_variation r where r.id_variation = "+ var.IdVariation + ")"
+
                      "and b.id_bib_ref = bd.id_bib_ref";


                OracleCommand cmd1 = new OracleCommand(sql, conn1);
                cmd1.CommandType = CommandType.Text;
                OracleDataReader dr1 = cmd1.ExecuteReader();
                try
                {
                    //TODO: Y si hay más de una referencia
bibliográfica???
                    dr1.Read();

                    var.Url = dr1.GetString(0);
                    var.NombreUrl = dr1.GetString(1);
                    conn1.Close();
                    conn1.Dispose();
                }
                catch (Exception ex)
                {
                    conn1.Close();
                    conn1.Dispose();
                }
            }
            //return fenotipo;
```

```
            #endregion
```

Web.Config

```
<?xml version="1.0"?>
<configuration>
      <configSections>
            <sectionGroup name="system.web.extensions"
type="System.Web.Configuration.SystemWebExtensionsSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
                  <sectionGroup name="scripting"
type="System.Web.Configuration.ScriptingSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
                        <section name="scriptResourceHandler"
type="System.Web.Configuration.ScriptingScriptResourceHandlerSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                        <sectionGroup name="webServices"
type="System.Web.Configuration.ScriptingWebServicesSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
                              <section name="jsonSerialization"
type="System.Web.Configuration.ScriptingJsonSerializationSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="Everywhere"/>
                              <section name="profileService"
type="System.Web.Configuration.ScriptingProfileServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                              <section name="authenticationService"
type="System.Web.Configuration.ScriptingAuthenticationServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                              <section name="roleService"
type="System.Web.Configuration.ScriptingRoleServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
                        </sectionGroup>
                  </sectionGroup>
            </sectionGroup>
      </configSections>
      <appSettings/>
      <connectionStrings/>
      <system.web>
    <customErrors mode="Off"/>

    <!--
            Establezca debug="true" en la compilación para insertar
símbolos
            de depuración en la página compilada. Dado que este
            proceso afecta al rendimiento, debe establecer este valor
como true
            durante la depuración.
```

```
        -->
            <compilation debug="true">
                  <assemblies>
                        <add assembly="System.Core, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                        <add assembly="System.Data.DataSetExtensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                        <add assembly="System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                        <add assembly="System.Xml.Linq,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
                  </assemblies>
            </compilation>
            <!--
            La sección <authentication> habilita la configuración
            del modo de autenticación de seguridad utilizado por
            ASP.NET para identificar a un usuario entrante.
        -->
    <!--authentication mode="Windows"/-->
    <anonymousIdentification />
            <!--
            La sección <customErrors> habilita la configuración de
            las acciones que se deben realizar si un error no
controlado tiene lugar
            durante la ejecución de una solicitud. Específicamente,
            permite a los desarrolladores configurar páginas de error
html
            que se mostrarán en lugar de un seguimiento de pila de
errores.

        <customErrors mode="RemoteOnly"
defaultRedirect="GenericErrorPage.htm">
            <error statusCode="403" redirect="NoAccess.htm" />
            <error statusCode="404" redirect="FileNotFound.htm" />
        </customErrors>
        -->
            <pages>
                  <controls>
                        <add tagPrefix="asp" namespace="System.Web.UI"
assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
                        <add tagPrefix="asp"
namespace="System.Web.UI.WebControls" assembly="System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
                  </controls>
            </pages>
            <httpHandlers>
                  <remove verb="*" path="*.asmx"/>
                  <add verb="*" path="*.asmx" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
                  <add verb="*" path="*_AppService.axd"
validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
                  <add verb="GET,HEAD" path="ScriptResource.axd"
type="System.Web.Handlers.ScriptResourceHandler,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" validate="false"/>
```

```
            </httpHandlers>
            <httpModules>
                  <add name="ScriptModule"
type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
            </httpModules>
      </system.web>
      <system.codedom>
            <compilers>
                  <compiler language="c#;cs;csharp" extension=".cs"
warningLevel="4" type="Microsoft.CSharp.CSharpCodeProvider, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">
                        <providerOption name="CompilerVersion"
value="v3.5"/>
                        <providerOption name="WarnAsError"
value="false"/>
                  </compiler>
            </compilers>
      </system.codedom>
      <!--
        La sección system.webServer es necesaria para ejecutar
ASPN.NET AJAX en Internet
        Information Services 7.0. No es necesaria para la versión
anterior de IIS.
    -->
      <system.webServer>
            <validation validateIntegratedModeConfiguration="false"/>
            <modules>
                  <remove name="ScriptModule"/>
                  <add name="ScriptModule"
preCondition="managedHandler" type="System.Web.Handlers.ScriptModule,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
            </modules>
            <handlers>
                  <remove name="WebServiceHandlerFactory-Integrated"/>
                  <remove name="ScriptHandlerFactory"/>
                  <remove name="ScriptHandlerFactoryAppServices"/>
                  <remove name="ScriptResource"/>
                  <add name="ScriptHandlerFactory" verb="*"
path="*.asmx" preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
                  <add name="ScriptHandlerFactoryAppServices" verb="*"
path="*_AppService.axd" preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
                  <add name="ScriptResource"
preCondition="integratedMode" verb="GET,HEAD"
path="ScriptResource.axd"
type="System.Web.Handlers.ScriptResourceHandler,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
            </handlers>
      </system.webServer>
      <runtime>
            <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
                  <dependentAssembly>
```

```
                        <assemblyIdentity name="System.Web.Extensions"
publicKeyToken="31bf3856ad364e35"/>
                        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
newVersion="3.5.0.0"/>
                    </dependentAssembly>
                    <dependentAssembly>
                        <assemblyIdentity
name="System.Web.Extensions.Design"
publicKeyToken="31bf3856ad364e35"/>
                        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
newVersion="3.5.0.0"/>
                    </dependentAssembly>
                </assemblyBinding>
        </runtime>
</configuration>
```

Variacion.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace prototipo
{

    public class Variacion
    {
        private int posicion = 0;
        private string idGen = string.Empty;
        private int idAllele = -1;
        private string cadenaQuery = string.Empty;
        private string cadenaRef = string.Empty;
        private string cadenaInsertada = string.Empty;
        private string cadenaEliminada = string.Empty;
        private int numBasesDel = 0;
        private int numBasesIns = 0;
        private int tipo = -1;
        private string url = string.Empty;
        private string nombreUrl = string.Empty;
        private int efecto = -1;
        private int idVariation = -1;
        private int idCadenaRef = -1;
        private bool esSnp = false;


        public enum TipoVariacion { Insertion, Deletion, InDel,
Inversion };

        public enum SpecializationEffect
        {
            Mutant,
            NeutralPolomorphism,
```

```
            UnkownConsequence
        } ;

        /// <summary>
        /// Indica la posición de la variación en la cadena de
referencia
        /// </summary>
        public int Posicion
        {
            get { return posicion; }
            set { posicion = value; }
        }

        /// <summary>
        /// Indica el gen en el que se produce la variación
        /// </summary>
        public string IdGen
        {
            get { return idGen; }
            set { idGen = value; }
        }

        /// <summary>
        /// cadaena insertada por el usuario
        /// </summary>
        public string CadenaQuery
        {
            get { return cadenaQuery; }
            set { cadenaQuery = value; }
        }

        /// <summary>
        /// indica el valor que debería tener de no haberse
        /// producido ninguna variación con respecto a la
        /// cadena de referencia
        /// </summary>
        public string CadenaRef
        {
            get { return cadenaRef; }
            set { cadenaRef = value; }
        }

        /// <summary>
        /// Indica de qué tipo de variación se trata
        /// inserción, deleción, inversión, indel
        /// para eso está la enumeración TipoVariacion
        /// </summary>
        public int Tipo
        {
            get { return tipo; }
            set { tipo = value; }
        }



        /// <summary>
        /// Número de bases que han sido borradas
        /// </summary>
        public int NumBasesDel
        {
            get { return numBasesDel; }
```

```
            set { numBasesDel = value; }
        }

        /// <summary>
        /// dirección url donde está la referencia bibliográfica
        /// </summary>
        public string Url
        {
            get { return url; }
            set { url = value; }
        }

        /// <summary>
        /// nombre que se verá en la url para la referencia
bibliográfica
        /// </summary>
        public string NombreUrl
        {
            get { return nombreUrl; }
            set { nombreUrl = value; }
        }

        /// <summary>
        /// Indica el efecto que tiene la variación,
        /// si es una mutación, un polimorfismo neutro
        /// o si aún no se conocen sus consecuencias
        /// </summary>
        public int Efecto
        {
            get { return efecto; }
            set { efecto = value; }
        }

        /// <summary>
        /// Contiene el identificador de la variación
        /// en la base de datos. En la tabla Variation.
        /// </summary>
        public int IdVariation
        {
            get { return idVariation; }
            set { idVariation = value; }
        }

        /// <summary>
        /// Contiene el identificador de la cadena de
        /// referencia que tiene asociada.
        /// </summary>
        public int IdCadenaRef
        {
            get { return idCadenaRef; }
            set { idCadenaRef = value; }
        }

        /// <summary>
        /// Número de bases que han sido insertadas
        /// </summary>
        public int NumBasesIns
        {
            get { return numBasesIns; }
            set { numBasesIns = value; }
        }
```

```
        /// <summary>
        /// Indica si la variación encontrada es un snp
        /// </summary>
        public bool EsSnp
        {
            get { return esSnp; }
            set { esSnp = value; }
        }

        public string CadenaInsertada
        {
            get { return cadenaInsertada; }
            set { cadenaInsertada = value; }
        }

        public string CadenaEliminada
        {
            get { return cadenaEliminada; }
            set { cadenaEliminada = value; }
        }

        public int IdAllele
        {
            get { return idAllele; }
            set { idAllele = value; }
        }
    }
}
```