UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Título: Human Computation en bioinformática. Desarrollo de un videojuego para la clasificación de proteínas.

Autores: Ezequiel Colautti – Martín Moro

Director: Dr. Julian Echave

Codirector: Ing. Armando De Giusti

Resumen

Una de las áreas de investigación más importantes de la biología son las proteínas. Para avanzar en su investigación, la clasificación de las más de 100.000 proteínas depositadas en el Banco de Datos de Proteínas en un conjunto relativamente menor de categorías es una tarea muy importante. Teniendo esto en cuenta, en este trabajo se propone aplicar el paradigma de Human Computation a la resolución de este problema de la Bioinformática. Para eso desarrollamos P3, un GWAP (del inglés, Game With A Purpose) que tiene como objetivo generar los datos necesarios para clasificar las proteínas a partir de la comparación de la representación tridimensional de las estructuras proteicas realizada por los jugadores. Por otro lado estudiamos la posibilidad de generar una clasificación de proteínas basadas en la información de los movimientos de las mismas además de su estructura.

Palabras Claves

- *Human Computation*
- *GWAP*
- *Proteínas*
- *Clasificación estructural de proteínas*
- *Clasificación de proteínas basada en el movimiento*
- *Bioinformática*

Trabajos Realizados

En primer lugar se realizó una investigación bibliográfica de los temas a desarrollar. Luego se programó una aplicación basada en el paradigma Human Computation, llamada "P3". P3 es un videojuego web en el que los jugadores deben identificar en un trío de proteínas aquella que consideren diferente de las dos restantes. Los jugadores avanzan por los distintos niveles a partir de la resolución correcta de una cantidad específica de tríos y son desafiados con niveles de dificultad progresiva para motivarlos a continuar jugando y respondiendo correctamente. Se realizaron pruebas con 30 jugadores para generar los datos y poder analizar la performance de la aplicación.

Conclusiones

Luego de realizada la aplicación y la generación de los datos a través de la misma, en función del análisis de los resultados obtenidos, puede observarse que es posible realizar una clasificación estructural de las proteínas que posea un alto grado de coincidencia con la clasificación SCOP. En cuanto al desarrollo de una clasificación basada en sus movimientos, si bien existen ciertas tendencias que indicarían que existe la posibilidad, se necesitarán más datos para verificar si agregar información sobre movimientos facilita la clasificación y/o conduce a una clasificación diferente que una puramente estructural.

Trabajos Futuros

En primer lugar sería interesante profundizar en la generación de una clasificación basada en el movimiento de las proteínas analizando los posibles cambios que deberían aplicarse a P3 para cumplir este objetivo y generando un volumen de datos mayor.

Por otro lado, sería útil desarrollar una mecánica de juego que permita resolver la clasificación de las más de 100.000 proteínas de una manera más eficiente y con la participación de una menor cantidad de jugadores.

Fecha de la presentación: **Noviembre de 2015**

Facultad de Informática UNLP

Tesina de Licenciatura en Informática

Human Computation en bioinformática. Desarrollo de un videojuego para la clasificación de proteínas.

Ezequiel Colautti

Martín Moro

Director:
Dr. Julián Echave

Co-director:
Ing. Armando De Giusti

5 de noviembre de 2015

1. Índice

2. Introducción

En este momento, millones de personas alrededor del mundo se encuentran jugando videojuegos en sus computadoras. Si sumamos las horas que cada uno de ellos dedica durante un año a esta actividad, obtendríamos un total de billones de horas hombre dedicadas a juegos electrónicos. ¿Qué pasaría si todo este tiempo y energía pudieran ser canalizados hacia trabajo útil? ¿Podríamos lograr que la gente, mientras juega videojuegos, resuelva problemas de gran escala de forma simultánea e inconsciente? [1]

Si bien, hoy en día, la tecnología permite que muchos problemas sean resueltos mediante la ejecución de un algoritmo, las computadoras aun no poseen la inteligencia conceptual básica y las capacidades perceptivas de las que la mayoría de los humanos disponen para resolver ciertos tipos de tareas de forma trivial. Entre estas tareas podemos nombrar a modo de ejemplo la obtención de información semántica de archivos de audio, video o imagen. El paradigma de *Human Computation* propone tratar los cerebros humanos como procesadores en un sistema distribuido, de forma tal que cada uno realice una pequeña parte de un procesamiento masivo. Esta propuesta tiene un enorme potencial para resolver problemas que aún no pueden ser resueltos computacionalmente.

Para lograr que las personas se interesen en participar en este proceso computacional colectivo, es necesario algún incentivo que los motive a colaborar. En este sentido, y teniendo en cuenta la enorme cantidad de horas hombre que se dedican anualmente a los videojuegos, podemos asumir que los juegos de computadora son un método eficiente para atraer participantes. Basándose en esta idea, surgen los juegos con un propósito (GWAP, del inglés *Games With A Purpose*) que se enfrentan al desafío de generar un mecanismo para atraer usuarios que, a medida que juegan para divertirse, contribuyan a realizar una parte del procesamiento de la solución. Los GWAP son una clase de sistemas de *Human Computation*.

En este trabajo intentaremos aprovechar las ventajas de *Human Computation* creando un GWAP para resolver un problema importante para la biología. Una de las áreas de investigación más importantes de la biología son las proteínas. Las proteínas son las biomoléculas responsables de la mayoría de las funciones de los organismos vivos. Su comprensión está en la base de la biofísica, la bioquímica, la biología evolutiva, la biotecnología y la biomedicina. Para avanzar en su investigación, la clasificación de las más de 100.000 proteínas depositadas en el Banco de Datos de Proteínas (PDB, *Protein Data Bank*) [2] en un conjunto relativamente menor de categorías se vuelve una tarea muy importante. Por esta razón, la clasificación de proteínas es uno de los temas más estudiados en las ciencias biológicas.

Actualmente, las dos clasificaciones más importantes son SCOP [3] y CATH [4]. También hay muchos otros esquemas de clasificación, la mayoría basados en métodos computacionales que analizan la estructura tridimensional o la cadena de aminoácidos de las proteínas. Es usual que se encuentren fallas o casos dudosos en estas

clasificaciones debido a que los métodos automáticos empleados se basan en definiciones diversas de similitud o disimilitud. Los criterios utilizados varían de acuerdo al grupo desarrollador, no habiendo consenso sobre cuál es la mejor medida.

Además de los sistemas que clasifican a las proteínas a partir de sus estructuras, existen unos pocos métodos para comparar la dinámica de las proteínas [5-8]. Estos métodos difieren mucho entre ellos y no existe una referencia que permita hacer estudios comparativos para poder decidir cuál es el mejor, lo que dificulta una clasificación automática confiable. Sin embargo, reconocer visualmente las similitudes y diferencias de los movimientos de las proteínas es relativamente fácil. El desarrollo de un sistema basado en *Human Computation* podría aprovechar lo mejor del procesamiento humano y del computacional para proveer un esquema de clasificación de proteínas por movimiento que sea capaz de convertirse en una referencia dentro del área. Una clasificación de este tipo representaría una innovación y sería de utilidad para muchas de las ciencias naturales como la medicina y la biología molecular.

Teniendo en cuenta lo mencionado previamente, en este trabajo desarrollamos P3, un GWAP que tiene como objetivo obtener dos tipos de clasificación de las proteínas. La primera clasificación se basa en las similitudes y diferencias entre las estructuras proteicas, mientras que la segunda surge a partir de la comparación de los movimientos que realizan. Para esto son las personas las que visualizando un modelo tridimensional de las proteínas deben compararlas y decidir cuales creen que pertenecen a la misma categoría.

En este capítulo solo presentamos una breve descripción del problema a resolver y los temas a tratar. En los capítulos 3, 4, 5 y 6 describimos los fundamentos teóricos de las diferentes áreas en los que se basa nuestro trabajo. En los capítulos 3 y 4 profundizamos en las técnicas informáticas utilizadas para desarrollar P3. Comenzamos explicando en detalle todos los aspectos de *Human Computation* para luego centrarnos en un tipo de aplicación de esta técnica como son los GWAP. En el capítulo 5 describimos brevemente la Bioinformática ya que el problema que intentamos resolver es abarcado por esta disciplina. En el capítulo 6 hablamos específicamente sobre las proteínas y la importancia de su clasificación, así como también de los distintos esquemas de clasificación que ya existen.

Una vez cubiertos los aspectos teóricos relacionados con este trabajo explicamos la solución implementada. En el capítulo 7 describimos P3 y las decisiones de diseño que tomamos. En el capítulo 8 discutimos las herramientas tecnológicas que utilizamos para desarrollar el juego.

En el capítulo 9 presentamos los resultados que obtuvimos luego de realizar las pruebas con 30 jugadores distintos. Finalmente en el capítulo 10 discutimos las conclusiones de esta investigación.

3. Human Computation

Gracias al avance de la tecnología, un gran número de problemas pueden resolverse mediante la ejecución de algoritmos computacionales. Sin embargo, hay problemas que son imposibles de resolver o cuyos resultados son insatisfactorios al resolverse computacionalmente. Algunos de estos casos resultan ser tareas sencillas para las personas ya que requieren de la creatividad y percepción humana para su solución. Pensemos, por ejemplo, en problemas que impliquen la identificación de las ideas principales de un texto, la realización de diagnósticos médicos, el reconocimiento de similitudes y diferencias entre modelos tridimensionales de objetos complejos del mundo real, la obtención de información semántica de archivos de audio, video o imagen, entre otros. Si bien las computadoras tienen el potencial de resolver muchos de estos problemas más rápidamente y con menor costo, la mayoría de estas soluciones automatizadas resultan ser de una calidad insatisfactoria. Incluso aunque las computadoras mejoran velozmente, siempre podremos pensar en problemas más difíciles.

La estrategia de *Human Computation* consiste en combinar las fortalezas de las computadoras y de los humanos delegando partes del problema a una gran cantidad de personas. El sistema computacional se diseña teniendo en cuenta los aspectos globales del problema a resolver y se generan subproblemas menores que se resuelven mediante las habilidades particulares de los humanos. *Human Computation* invierte la relación entre las personas y las computadoras, ya que son los humanos los que ayudan a las computadoras a realizar sus tareas. Esto es diferente de la idea tradicional de cooperación entre personas y computadoras donde las máquinas son las que facilitan las tareas de los usuarios.

3.1. Definición

El término *Human Computation* fue utilizado por primera vez en 1838 dentro de la literatura filosófica y psicológica. Sin embargo a nosotros nos interesa el uso moderno de este término que fue inspirado por Luis Von Ahn y utilizado en el año 2005 como título de su tesis doctoral. En este trabajo el autor presenta la siguiente definición:

"Human Computation es un paradigma que utiliza el poder de procesamiento de los humanos para solucionar problemas que las computadoras aún no puedan resolver".

Esta definición es bastante compatible con las definiciones dadas por el resto de los autores [9-14]. Al ser un área de investigación relativamente nueva donde sus aspectos teóricos están comenzando a desarrollarse, es importante considerar las distintas definiciones que han dado los investigadores para poder obtener una visión global y comprender las características más importantes de esta técnica. A partir del estudio de las mismas, identificamos dos características claves de *Human Computation*:

- Los problemas se ajustan al paradigma general de la computación y por lo tanto podrían algún día ser solucionados por las computadoras.

- La participación se solicita de forma proactiva con el propósito de ejecutar un cálculo. Esto excluye métodos como *data mining* donde se analizan los datos generados por los humanos de manera retrospectiva.

3.2. Aspectos de diseño en Human Computation

El denominador común entre la mayoría de los sistemas de *Human Computation* es que confían en los usuarios para proveer resultados. Estos resultados son agregados para obtener una base sobre la cual las computadoras deberán realizar ciertas tareas para resolver el problema global. A continuación describiremos brevemente los principales problemas que surgen al diseñar un sistema de *Human Computation* y presentaremos algunas de las soluciones típicas que se suelen utilizar [15].

3.2.1. Motivación

Uno de los desafíos más difíciles para el desarrollo de sistemas de *Human Computation* consiste en encontrar la forma de motivar a las personas a participar. Este problema se simplifica un poco por el hecho de que la mayoría de los sistemas de *Human Computation* se basan en redes de personas sin relación entre ellas que se encuentran conectados mediante sus computadoras en sus hogares. De esta forma no hay necesidad de motivarlos a concurrir a algún lugar o de realizar tareas demasiado distintas a las que realizan habitualmente. Sin embargo, como en general las acciones que deben realizar no benefician directamente a los participantes, es necesario presentar algún tipo de motivación para que la gente participe.

Algunos de los métodos más utilizados para generar esta motivación son los siguientes:

- Pagar: Es una de las formas más fáciles de motivar a los usuarios, ya que si se dispone de un poco de dinero para pagarles, resulta fácil transformar ese dinero en usuarios utilizando el sistema. Uno de los problemas con este enfoque es que cuando se involucra al dinero la gente tiende a intentar engañar al sistema para incrementar su paga. Además los participantes pueden dar respuestas erróneas con el objetivo de aumentar la cantidad de tareas resueltas y de esta forma ganar más dinero.

- Altruismo: Puede parecer fácil confiar en el deseo de ayudar, pero esto requiere que los participantes realmente consideren que el problema que se está solucionando es interesante e importante. Efectivamente, se debe tratar de una tarea que muchas personas deseen ayudar a resolver. En la mayoría de los casos el número de usuarios tiende a ser limitado.

- Implícito: Si tenemos la suficiente suerte o ingenio como para encontrar una forma de embeber las tareas a resolver dentro de las actividades regulares de la gente, sería posible convertir la participación de las personas en una parte natural de algo que los usuarios ya estuviesen realizando en su vida cotidiana.

- Diversión: Si se logra crear un juego que la gente elija jugar porque lo disfruta, ya no existe la necesidad de pagarles para que participen. Además estos usuarios son

propensos a jugar por un periodo de tiempo mayor. Sin embargo, esto también es difícil porque es realmente un desafío convertir muchas tareas computacionales en un juego que sea divertido.

3.2.2. Control de calidad

Aunque los usuarios estén motivados, podrían intentar hacer trampa o engañar al sistema. También puede ser que aunque actúen de buena fe no comprendan correctamente las instrucciones y resuelvan el problema de forma incorrecta. Por estos motivos es importante contar con mecanismos de control para garantizar la calidad de los resultados y descartar la información errónea así como también evitar las trampas por parte de los usuarios. Algunos de los mecanismos más conocidos son:

- Coincidencia de resultados: Dos participantes trabajan de forma independiente y simultánea en diferentes lugares sobre el mismo problema. La respuesta no será considerada correcta a menos que ambos coincidan.
- Diseño de tareas defensivo: Este enfoque consiste en diseñar las tareas a realizar de manera que sea más fácil cumplirlas que hacer trampa.
- Redundancia: en este caso cada tarea deberá ser completada por múltiples participantes y luego se intentará identificar la mejor solución y aquella que más coincidencia tenga entre todos los participantes. De esta forma es posible descartar el trabajo realizado por las personas que suelen tener un rendimiento bajo.
- Revisión de expertos: un conjunto de reconocidos expertos debería realizar una revisión de las contribuciones por su relevancia y su aparente nivel de acierto.
- Revisión multinivel: Un primer conjunto de participantes realizan una serie de tareas, luego otro conjunto de personas deberá controlar las soluciones y puntuar la calidad de las mismas.
- Filtro estadístico: Se realiza una agregación de los datos de forma tal que las contribuciones irrelevantes puedan ser detectadas y eliminadas.

Cada una de estas técnicas tendrá sus diferentes formas de aplicación dependiendo del tipo de sistema que se busca controlar.

3.2.3. Habilidades humanas requeridas

Dependiendo del tipo de aplicación que se esté desarrollando, los tipos de habilidades que se requerirán pueden variar desde capacidades innatas compartidas por la mayoría de los humanos, hasta conocimientos específicos que solo poseen algunos. Cuando se diseña un sistema de *Human Computation* es útil ser específico en el tipo de capacidad que será utilizada para de esta forma identificar los aspectos del problema que pueden ser resueltos de forma fácil por la computadora.

3.3. Distintos tipos de Human Computation

En esta sección presentaremos algunos tipos de aplicaciones de *Human Computation*. Más allá de que la categoría que profundizaremos serán los juegos con un propósito,

es importante mencionar otras opciones para ver como el paradigma de *Human Computation* puede ser aplicado de diferentes formas [13].

- Trabajo mecanizado: Este tipo de aplicaciones se caracteriza por el uso de dinero como parte de la motivación. Participan voluntarios rentados para realizar tareas explícitamente definidas que suelen tomar un breve periodo de tiempo. El ejemplo más notable es "Amazon's Mechanical Turk" [16], que permite que los desarrolladores escriban programas que asignan de forma automática tareas pequeñas que deben ser resueltas por una red de trabajadores en el sitio web de Mechanical Turk.

- Tareas de doble propósito: Este tipo de aplicaciones se distingue por el uso de tareas que la gente realiza generalmente independientemente del objetivo del sistema de Human Computation. ReCAPTCHA [17] es un ejemplo muy inteligente de como trasladar una tarea computacional a una actividad que muchas personas realizan de manera frecuente. ReCAPTCHA utiliza las respuestas realizadas por las personas cuando responden un CAPTCHA para poder transcribir libros y diarios antiguos previamente escaneados que no fueron reconocidos por programas de reconocimiento de caracteres.

- Grandes Búsquedas: Se distingue de las demás aplicaciones mencionadas porque en lugar de realizar una agregación o colección de muchos resultados provistos por muchas personas, el objetivo es el de encontrar la única solución correcta para un problema. Dado un conjunto de elementos, por ejemplo imágenes, los participantes deberán buscar cada uno entre un modesto número de elementos aquel que cumpla con ciertas características para solucionar un problema.

- Juegos con un Propósito: Esta técnica consiste en realizar un videojuego que además de entretener a los participantes, produzca información útil para resolver un problema determinado a medida que es utilizado. En el capítulo 4 describiremos con detalle este tipo de aplicación de *Human Computation*.

4. Juegos con un Propósito (GWAP)

Un enfoque popular para motivar a las personas a participar en el proceso de *Human Computation* es el de expresar el problema en términos de un juego con un propósito (GWAP, *Games With A Purpose*). La idea se basa en que las personas invierten muchísimas horas jugando en línea, por lo que sería posible utilizar esta energía para algún objetivo en particular. A diferencia de otros trabajos que han intentado utilizar un conjunto de personas distribuidas para resolver un problema, el paradigma que describimos en esta sección no se basa en el altruismo o en incentivos económicos para motivar a los jugadores a realizar ciertas acciones, sino en las ganas de divertirse. En este tipo de juegos las personas realizan un procesamiento útil como un resultado secundario. Al crear un juego de este tipo es importante probar que el mismo sea correcto y que podrán obtenerse resultados de alta calidad.

El proceso de transformar un problema en un GWAP, implica crear un juego cuya estructura favorezca la resolución correcta y fácil para los jugadores. Un juego puede ser especificado mediante el objetivo que los jugadores tratan de cumplir y un conjunto de reglas que determinan lo que se puede y no se puede hacer durante su desarrollo. Las reglas de los GWAP deberían encaminar a los jugadores a realizar los pasos necesarios para resolver el problema computacional y en caso de ser posible, incluir ciertas garantías probabilísticas de que el resultado del juego será útil incluso si el jugador quisiera generar datos erróneos.

Otra consideración al desarrollar un GWAP es la necesidad de construir un sistema que cumpla con dos objetivos muy importantes: entretener a los jugadores y lograr obtener datos relevantes para el problema. El incumplimiento de alguno de estos objetivos podría provocar el fracaso del proyecto. Por un lado se podría obtener un juego que sea interesante para un gran número de jugadores pero que no obtenga información útil para el problema que se quiere resolver. La contraparte sería un sistema diseñado para recopilar datos relevantes pero cuya utilización no represente una tarea entretenida para los jugadores por lo que, probablemente, nadie estaría interesado en participar [18]. No entraremos en la discusión filosófica de la definición de un juego “divertido” o “agradable”, si no que consideraremos que un juego es exitoso si logra las horas hombre deseadas.

4.1. GWAPs en la actualidad

Desde el desarrollo del primer GWAP hasta la actualidad varios han tenido un gran éxito a nivel internacional, como por ejemplo:

- ESP Game: fue el primer GWAP desarrollado. Su objetivo es la identificación y etiquetado de imágenes. En este juego, dos jugadores anónimos se encuentran con una misma imagen y se les pide que describan su contenido. Los jugadores no pueden comunicarse entre sí y se considerará que resuelven el problema cuando haya consenso en las descripciones de ambos [19].

- Foldit: se utiliza para predecir la estructura de proteínas a partir de los datos recolectados en el juego y se basa en la capacidad de las personas para manipular objetos en 3D. Los datos obtenidos a partir de este juego han sido de mucha ayuda en la investigación de las funciones de las proteínas de ciertas enfermedades, como el HIV/SIDA, Alzheimer o cáncer [20].

- EteRNA: tiene como objetivo la predicción del plegamiento de las moléculas del ARN. Los investigadores esperan capitalizar las ventajas de la inteligencia colectiva de los jugadores de EteRNA para responder preguntas fundamentales sobre los mecanismos de plegamiento del ARN. Los diseños más votados son sintetizados en un laboratorio de bioquímica de la Universidad de Stanford para evaluar los patrones de plegamiento de estas moléculas y luego ser comparados directamente con predicciones realizadas por computadoras. De esta forma también se busca mejorar los modelos computacionales actuales.

- Play to Cure: Genes in Space: los jugadores analizan información genética real para ayudar a encontrar curas contra el cáncer.

Si bien son pocos los juegos existentes basados en *Human Computation*, la mayoría de ellos han obtenido excelentes resultados en las diferentes áreas de investigación para las que fueron desarrollados. Es importante destacar algunos aportes significativos que han logrado varios de ellos para las ciencias biológicas:

- En el 2011 los jugadores de Foldit ayudaron a descifrar la estructura cristalina de la proteasa retroviral del virus Mason-Pfizer de los monos (M-PMV), un virus causante del SIDA en monos. Científicos especializados no habían podido descifrar la estructura de esta proteína en 15 años [21-22].

- En Enero del 2012 los jugadores de Foldit lograron rediseñar la estructura de la enzima encargada de la reacción Diels-Alder utilizada en química inorgánica. Los jugadores mejoraron la enzima ya existente, mejorando su productividad en más de 18 veces [23].

- En marzo del 2014 la asociación Cancer Research UK, encargada de analizar los datos obtenidos del juego Play to Cure, informó que, en un mes, los jugadores habían logrado clasificar más de 1.5 millones de patrones de la base de datos. Este logro le hubiese costado seis meses de análisis exhaustivo a cualquier científico especializado [24].

4.2. Diseño de un GWAP

Una vez que se identifica el problema que se quiere resolver y cuál debería ser la interacción humana para realizar dicha tarea, debemos trasladarlo al formato de un juego. Esto se logra creando una sesión de juego, que consiste en una serie de interacciones entre el usuario y el sistema para generar información. Lo primero que debe establecerse es cuál es el tiempo aproximado que debería durar cada sesión. En los juegos serios creados hasta el momento, generalmente las sesiones de juego duran

entre tres y seis minutos. Esto es una ventaja ya que motiva a las personas que cuentan con poco tiempo o que solo quieren tomarse un breve descanso para jugar.

Luego de definir el tiempo promedio que se desea utilizar, el diseñador debe determinar cuántas instancias del problema pueden ser realizadas por el jugador en dicho periodo de tiempo. Esto requiere un estudio preliminar para evaluar el tiempo promedio que les toma a las personas resolver una de las tareas deseadas.

A continuación enumeraremos algunos de los factores más importantes que se deben tener en cuenta a la hora de diseñar el juego para generar una experiencia agradable en el usuario [1]:

- Puntaje: en el caso de los juegos con un propósito, probablemente el método más directo para presentar un desafío a los jugadores es el de asignar puntos por cada instancia correcta lograda. Por ejemplo, en el caso de ESP Game, se le otorga un punto a cada jugador cuando hay coincidencia en la descripción provista. Con este sistema además se podrán generar retos como intentar superar un puntaje de un juego anterior o completar todos los niveles con cierto puntaje.

- Nivel de habilidad de los jugadores: otra forma de establecer metas es el uso de niveles de acuerdo a la habilidad del jugador. Por ejemplo, en el juego ESP Game existen cinco niveles de habilidad en los que se distribuyen de acuerdo a la cantidad de puntos que hayan acumulado. Cada jugador nuevo ingresa en el nivel de principiante y debe ganar una cierta cantidad de puntos para avanzar al siguiente nivel.

- Ranking: otra manera de generar desafíos para los jugadores es la presencia de un ranking. Esto es, una lista que presenta el nombre de usuario y los puntos de los jugadores con los puntajes más altos. El ranking puede construirse considerando los puntos obtenidos en distintos periodos de tiempo, por ejemplo, en la última hora, el último día, el mes corriente o incluso históricamente, lo que motiva a los jugadores a intentar aparecer en alguno de ellos.

- Aleatoriedad: en general, todos los juegos basados en Human Computation tienen un aspecto aleatorio. Esto se debe a que los datos de entrada que se otorgan en una sesión de juego generalmente se toman al azar a partir de un conjunto de entradas posibles. El hecho de que los datos sean aleatorios por sí mismo es un factor que genera motivación, ya que la dificultad de cada juego varía en cierta medida y esto mantiene entretenidos tanto a los jugadores principiantes como a los expertos.

5. Bioinformática

Las tecnologías de la información se han vuelto herramientas indispensables para las investigaciones en el campo de la biología, donde los científicos las utilizan cada vez más para resolver problemas biológicos. A esta aplicación de la informática, se la conoce como bioinformática [25].

La bioinformática es una ciencia relativamente nueva que ha guiado a muchos descubrimientos biológicos relevantes. Provee bases de datos centralizadas globalmente accesibles que permiten a los científicos buscar, agregar y analizar información. También brinda software para el análisis, la comparación, el modelado, la visualización y la interpretación de información biológica. Por ejemplo, gracias a la bioinformática es posible visualizar estructuras invisibles, como las proteínas, lo que facilita el estudio de su funcionamiento y estructura.

La cantidad de información biológica aumenta considerablemente año tras año. Por ejemplo, consideremos la base de datos de secuencias genéticas del NIH (*National Institutes of Health* de Estados Unidos) conocida como GenBank [26]. Esta disponía de aproximadamente 11.546.000 entradas en abril de 2001, que aumentaron a 56.620.500 en el 2006 y en la actualidad ya dispone de más de 182.188.700 secuencias distintas. Por esta razón es indispensable contar con técnicas computacionales que permitan entender y organizar grandes cantidades de información biológica.

5.1. Objetivos de la Bioinformática

En general, la bioinformática tiene tres objetivos principales [27]. En primer lugar, organizar los datos de manera que los investigadores puedan acceder a la información existente, así como también agregar nuevas entradas; por ejemplo, el banco de datos de proteínas que almacena información de estructuras macromoleculares. Esta información es inútil mientras que no sea analizada, por eso el propósito de la bioinformática es mucho mayor que el almacenamiento de información.

El segundo objetivo es desarrollar herramientas y recursos que sirvan para analizar los datos disponibles. Para estos desarrollos es necesario disponer de conocimientos expertos sobre la teoría computacional y de un conocimiento profundo de la biología.

Finalmente, como última meta, debemos mencionar la tarea de utilizar estas herramientas para analizar los datos e interpretar los resultados de manera significativa para la biología. Tradicionalmente, los estudios biológicos examinaron sistemas individuales en detalle, y frecuentemente los comparaban con unos pocos sistemas relacionados. Gracias a la bioinformática, actualmente se puede realizar un análisis global de todos los datos disponibles con el objetivo de descubrir los principios comunes a muchos sistemas [28].

La bioinformática provee las herramientas necesarias para aplicar el método científico a una gran cantidad de datos y debería ser vista como un enfoque científico para poder generar nuevos y diferentes tipos de preguntas biológicas.

5.2. Principales áreas de aplicación

Existen muchas áreas de la biología donde se aplica la bioinformática para entender el funcionamiento de sistemas biológicos complejos. Cualquier sistema donde la información pueda ser representada digitalmente ofrece una potencial aplicación de la bioinformática. De esta forma, puede aplicarse en el estudio de una simple célula o de un ecosistema completo.

Dos de las ciencias más importantes que utilizan la bioinformática son la genómica y la proteómica. La bioinformática provee una plataforma tecnológica que permite que los científicos trabajen y puedan analizar la gran cantidad de datos producidos en estas dos áreas de investigación.

Genómica es el conjunto de ciencias y técnicas dedicadas al estudio integral del funcionamiento, el contenido, la evolución y el origen de los genomas. Un genoma puede ser pensado como el conjunto completo de las secuencias de ADN que constituyen el material hereditario transmitido de generación en generación. Estas secuencias de ADN incluyen todos los genes (las unidades hereditarias funcionales y físicas que se trasmiten de los progenitores a su descendencia) y los transcriptomas (las copias de ARN que son el paso inicial para decodificar la información genética) incluidos en un genoma [29].

La Proteómica, por su parte, comprende el estudio a gran escala de las proteínas, en particular de su estructura y función. Las proteínas son partes vitales de los organismos vivos, ya que son los componentes principales de las rutas metabólicas celulares [30].

Los principales esfuerzos de investigación dentro de la bioinformática incluyen el alineamiento de secuencias de ácidos nucleicos y proteínas, la predicción de genes, el montaje del genoma, el alineamiento estructural de proteínas, la predicción de las estructuras de las proteínas, la predicción de la expresión génica, las interacciones proteína-proteína, y el modelado de la evolución.

Al entender las interacciones que ocurren entre todas las áreas de un genoma o una proteína, la bioinformática tiene el potencial de ofrecer claves para el entendimiento y modelado de cómo se manifiestan ciertas enfermedades humanas o estados saludables específicos.

6. Proteínas

Las proteínas son moléculas esenciales para el correcto funcionamiento de los organismos y ocupan un lugar de máxima importancia entre las moléculas constituyentes de los seres vivos (biomoléculas). Esto se debe a que son el elemento básico de construcción de los tejidos que forman nuestro cuerpo y regulan numerosas funciones vitales. Prácticamente todos los procesos biológicos dependen de la presencia o la actividad de este tipo de moléculas. Bastan algunos ejemplos para dar idea de la variedad y trascendencia de las funciones que desempeñan. Son proteínas [31]:

- Casi todas las enzimas, catalizadores de reacciones químicas en organismos vivientes.
- Muchas hormonas, reguladoras de actividades celulares.
- La hemoglobina y otras moléculas con funciones de transporte en la sangre.
- Los anticuerpos, encargados en parte de la defensa natural contra infecciones o agentes patógenos.
- Los receptores de las células, a los cuales se fijan moléculas capaces de desencadenar una respuesta determinada.
- La actina y la miosina, responsables del acortamiento del músculo durante la contracción.
- El colágeno, integrante de fibras altamente resistentes en tejidos de sostén.

Las proteínas poseen una estructura química central que consiste en una cadena lineal de aminoácidos plegada, formando una estructura tridimensional, lo que les permite realizar sus funciones. Existen 20 aminoácidos diferentes que se combinan entre ellos de múltiples maneras para formar cada proteína.

La estructura de las proteínas puede jerarquizarse en:

- Estructura primaria: secuencia lineal de aminoácidos.
- Estructura secundaria: plegamiento regular local causado por las interacciones entre los aminoácidos.
- Estructura terciaria: modo en que la cadena polipeptídica se pliega en el espacio.
- Estructura cuaternaria: deriva de la asociación de varias cadenas peptídicas para formar un multímero, que posee propiedades distintas a la de sus monómeros componentes.

6.1. Base de datos de proteínas

Desde 1971, el *Protein Data Bank* (Banco de Datos de Proteínas - PDB) ha servido como el único repositorio de información acerca de la estructura tridimensional de

proteínas, ácidos nucleicos y otros compuestos complejos. La información del PDB se mantiene actualizada gracias a la colaboración internacional de investigadores, educadores y estudiantes. La organización *Worldwide PDB* (wwPDB) se encarga de administrar el archivo PDB y se asegura de que el mismo continúe estando disponible para la comunidad global de forma gratuita y pública.

La información contenida en el archivo incluye coordenadas atómicas, factores estructurales cristalográficos y datos de NMR (*Nuclear Magnetic Resonance spectroscopy*) experimental. Además de coordenadas, cada entrada contiene los nombres de las moléculas, información de su estructura primaria, estructura secundaria, información del ensamblaje biológico, de ligandos y más.

6.2. Clasificación estructural de las proteínas

La clasificación estructural de las proteínas es un problema de la bioinformática muy importante en diversas áreas de investigación de proteínas. Dentro de estas áreas se incluyen la predicción de la estructura y las funciones de las proteínas, el estudio de la relación estructural y evolutiva entre proteínas, y la identificación de potenciales sitios de unión. Existen muchos esquemas de clasificación de proteínas, de los cuales los más reconocidos y utilizados son SCOP y CATH. Ambos dividen las proteínas en dominios que a su vez son clasificados de manera jerárquica.

6.2.1. SCOP

SCOP ordena los dominios de proteínas en *clases, folds, superfamilias y familias*. En su primer nivel define cuatro *clases* conocidas como *all α*, *all β*, *α/β* y *α + β* según la estructura secundaria de la proteína.

Primero cada dominio es clasificado dentro de una de las cuatro clases mencionadas previamente. En el nivel de *fold*, las similitudes entre las proteínas a nivel estructural representan el principal factor de agrupamiento, aunque pueden existir ciertos enlaces evolutivos entre ellas. En la clasificación de *superfamilias* se define más claramente la ascendencia evolutiva común, ya que se considera que las proteínas con estructuras y/o características funcionales similares poseen un origen evolutivo común. Las proteínas con secuencias de aminoácidos similares o estructuras y funciones muy similares implican una relación evolutiva más cercana y son agrupadas dentro de las *familias*. De esta forma, los miembros de las mismas familias o superfamilias en SCOP comparten la misma ascendencia u origen evolutivo.

El nivel de superfamilia de SCOP es especialmente interesante, porque las homologías remotas que implican que dos proteínas comparten funciones o un origen evolutivo común son difíciles o imposibles de detectar usando solo la información de la secuencia. Los intentos de automatizar la clasificación de las estructuras de las proteínas han desarrollado un interés particular en las medidas de similitud estructurales basándose en la alineación geométrica de las cadenas proteicas. En este sentido, SCOP ha sido frecuentemente considerada como la clasificación estándar o de referencia para comparar los métodos automáticos de clasificación de conjuntos de estructuras relacionados estructural y funcionalmente.

6.2.2. CATH

Los cuatro niveles principales de CATH son, *Clases, Arquitecturas, Topologías y Superfamilias Homologas* (en inglés *Homologous superfamilies*). De estos cuatro niveles proviene el nombre de la clasificación (C-A-T-H).

El proceso de creación de CATH contiene más pasos automáticos y menos intervención humana en comparación con SCOP. De forma análoga a SCOP, CATH comienza en el nivel de *clase* definiendo tres clases mayores según su estructura secundaria, estas son *all α*, *all β*, *α/β*.

Luego de obtener las estructuras de las proteínas de la PDB, se realizan comparaciones automáticas para agruparlas de acuerdo al grado de similitud de sus secuencias. Primero se define la *clase* de cada dominio para evitar la realización de comparaciones innecesarias entre las estructuras de distintas clases. En general este proceso es automático aunque en los casos muy difíciles podría haber intervención humana. Las comparaciones automáticas entre las proteínas de cada clase permiten producir los niveles de *topología* y *superfamilias homologas*. El último paso es la asignación manual de las arquitecturas usando la inspección visual y las referencias a la literatura. Los dominios que se encuentran dentro del mismo nivel de superfamilia homologa comparten el origen evolutivo.

Es importante remarcar que al utilizar métodos diferentes para clasificar las estructuras, en algunos casos se obtendrá una clasificación diferente de la misma proteína. Estas diferencias pueden ser encontradas tanto en la forma de particionar el dominio como también en la clasificación de un dominio a su clase correspondiente. Las diferencias y similitudes entre SCOP y CATH han sido evaluadas [32-33] y esos análisis permitieron obtener información valiosa de los problemas y desafíos que se presentan cuando se intenta clasificar las estructuras de las proteínas.

7. P3

Para lograr nuestro objetivo de realizar un GWAP capaz de generar datos que permitan clasificar proteínas hemos desarrollado una aplicación basada en el paradigma *Human Computation*, llamada "P3". P3 es un videojuego web en el que los jugadores deben identificar en un trío de proteínas aquella que consideren diferente de las dos restantes. Los jugadores avanzan por los distintos niveles a partir de la resolución correcta de al menos una cantidad específica de tríos y son desafiados con niveles de dificultad progresiva para motivarlos a continuar jugando y respondiendo correctamente.

7.1. Funcionalidad de P3

P3 es un juego web que puede ejecutarse en cualquier dispositivo (tanto móviles como de escritorio) que cuente con un navegador y acceso a internet. Está diseñado para ser utilizado por un solo jugador que deberá superar ocho niveles para completarlo. Una vez que haya completado los ocho niveles podrá volver a jugar desde el principio con diferentes proteínas. Más adelante explicaremos detalladamente como se compone cada nivel y las distintas dificultades de los mismos. Por el momento nos alcanzará con conocer que en cada nivel el jugador será desafiado con la resolución de diez tríos de proteínas de dificultad similar entre ellos. La resolución correcta de un trío otorgará un punto al jugador; si consigue obtener los puntos necesarios, el jugador pasará al siguiente nivel.

Para poder ingresar al juego, un jugador debe tener una cuenta de usuario. Una vez que se registre podrá jugar cuando lo desee y se mantendrá el estado de su juego a lo largo del tiempo. Esto quiere decir que podrá dejar de jugar en cualquier momento y la siguiente vez que se conecte a la aplicación, continuará desde el lugar en que lo dejo, sin importar si lo hace desde el mismo dispositivo o no.

Un trío consiste en la presentación en la pantalla de la representación gráfica tridimensional de tres proteínas seleccionadas de acuerdo a su clasificación en SCOP. Los parámetros utilizados para la selección de las proteínas se explicarán detalladamente en la sección "Niveles". Lo importante es entender que al seleccionar los tríos de acuerdo a la clasificación de SCOP, lo que el sistema hace es elegir dos proteínas que se encuentran clasificadas de forma similar en alguno de los niveles jerárquicos de SCOP (dependiendo el nivel se buscaran proteínas de la misma familia, superfamilia, fold o clase) y la última proteína es seleccionada de una clasificación de SCOP distinta de acuerdo al nivel en el que se encuentre el jugador. Esto significa que hay dos proteínas que tendrán cierto grado de similitud (de acuerdo a la clasificación de SCOP) y una tercera que será más diferente a las otras dos. La tarea que debe resolver el jugador consiste en identificar, a partir de la representación gráfica de las tres proteínas, cual es la que difiere más de las otras dos, es decir, cual es la que SCOP clasifica distinto de las tres.

En la pantalla del jugador se muestran las tres proteínas y al lado de cada una de ellas se encuentra un botón que puede ser clickeado por el usuario para indicar que es la proteína que considera diferente. En caso de que lo indicado por el jugador coincida con la clasificación de SCOP se considera que el trío se resolvió de forma correcta y se le otorga un punto. Si al finalizar los diez tríos del nivel el jugador tiene el puntaje necesario, avanzará al siguiente nivel. Si por el contrario no alcanza este objetivo, deberá jugar nuevamente con tríos de la misma dificultad.

Una característica de P3 es que dispone de dos tipos de niveles que se presentan de forma intercalada. En uno de los dos tipos, solo se muestra en pantalla la representación tridimensional de la estructura de las proteínas; en el otro tipo de nivel además de mostrar la estructura de las proteínas, se representan los movimientos de las mismas, de forma tal que el jugador pueda diferenciar cuales se mueven de forma similar.

Más adelante explicaremos más detalladamente como está compuesta la interfaz del usuario, pero por el momento y a modo de introducción diremos brevemente que además de la representación de las tres proteínas, el jugador cuenta con un conjunto de opciones de configuración como habilitar o deshabilitar el sonido, poner el juego en pantalla completa, hacer girar las proteínas o solicitar ayuda en caso de que se le presenten dudas. Además en otra parte de la pantalla se mantiene información del estado del juego como el puntaje del jugador, su eficacia, el nivel en que se encuentra, la cantidad de tríos resueltos, etc.

7.2. Interfaz del usuario

Además del juego propiamente dicho, P3 se presenta como un sitio web básico compuesto por algunas páginas más que el juego en sí. Al ingresar la URL de P3 (actualmente alojada en http://p3tesina.hclass.webfactional.com/) se presenta la página inicial donde se puede ver una breve descripción del juego con imágenes y un video explicativo, además en la esquina superior derecha se encuentra la opción de Entrar al juego (Imagen 1).

Imagen 1. Página principal de P3

Si el usuario presiona el botón “Entrar” se muestran dos formularios que podrá completar dependiendo de su situación (Imagen 2). Si nunca había ingresado al juego, podrá utilizar el formulario de Registro para crear su cuenta y acceder al juego. En el caso de que ya disponga de un usuario podrá ingresar a su cuenta mediante su nombre de usuario y contraseña.

Imagen 2. Pantalla de Acceso y Registro

Una vez dentro del juego podemos ver en la parte superior de la pantalla una barra con el nombre del juego, el nombre del usuario y las opciones del menú del jugador. Estas opciones comprenden: mostrar el juego en pantalla completa, habilitar o deshabilitar el sonido, abrir el panel de ayuda al jugador, ver el ranking o salir del juego. Si el jugador presiona el botón de ayuda, se desplegará un panel gris en el sector izquierdo de la pantalla con el texto necesario para brindar la ayuda al jugador (Imagen 3).

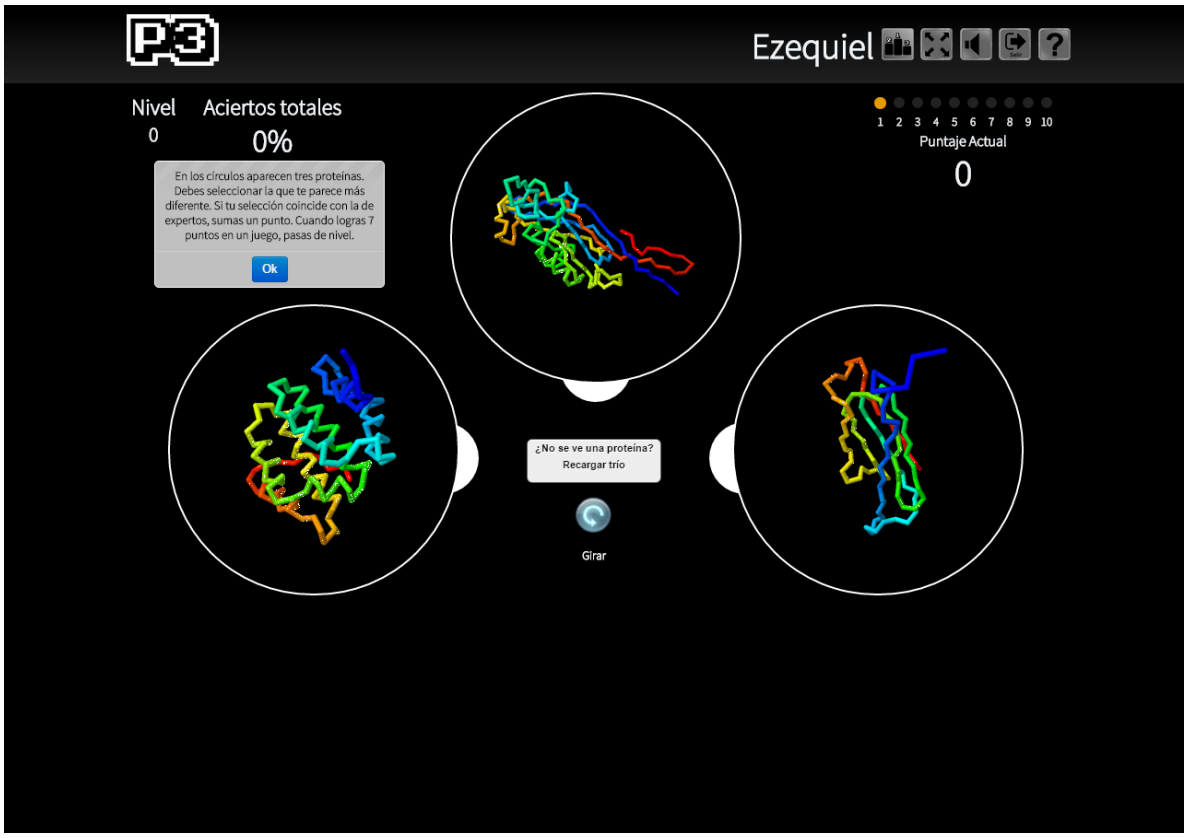
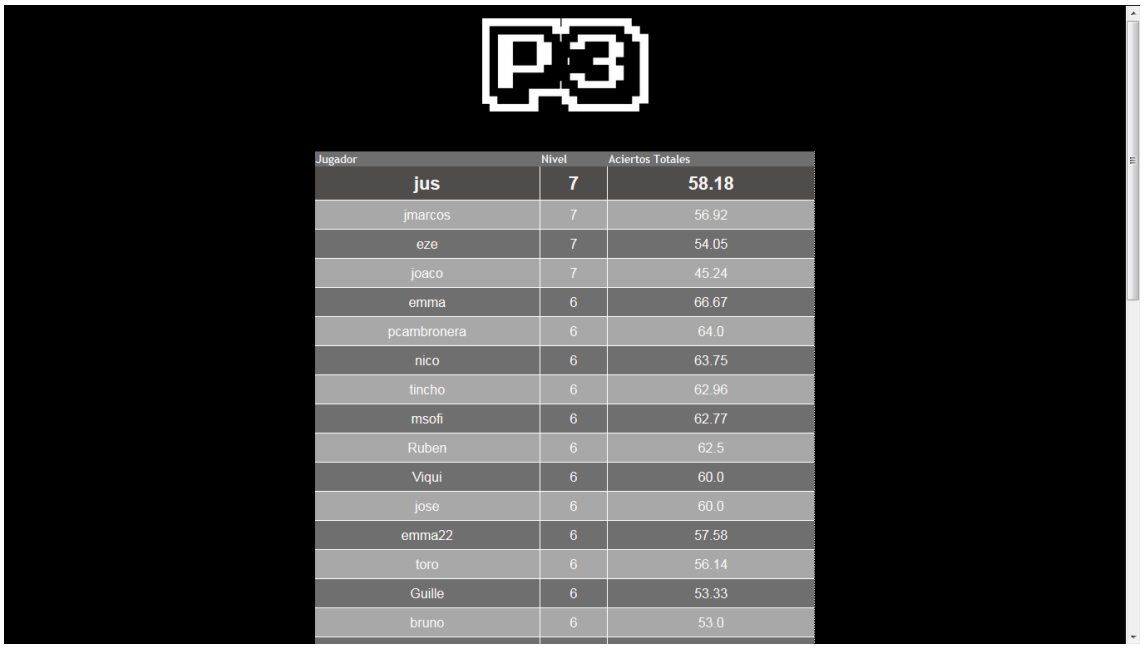


Imagen 3. Pantalla de juego (1)

Si el jugador presiona el botón de ranking, accederá a la página del "Ranking" global de P3, dónde encontrará un listado de jugadores con sus nombres, nivel máximo alcanzado y porcentaje total de aciertos, siendo el primer jugador aquel con el mayor nivel y mejor porcentaje (Imagen 4).



Jugador	Nivel	Aciertos Totales
jus	7	58.18
jmarcos	7	56.92
eze	7	54.05
joaco	7	45.24
emma	6	66.67
pcambronera	6	64.0
nico	6	63.75
tincho	6	62.96
msofi	6	62.77
Ruben	6	62.5
Viqui	6	60.0
jose	6	60.0
emma22	6	57.58
toro	6	56.14
Guille	6	53.33
bruno	6	53.0

Imagen 4. Pantalla de Ranking

Por debajo de la barra de la parte superior podemos encontrar dividido entre el sector izquierdo y derecho la información del estado del juego y del puntaje del jugador. A la izquierda se muestra el nivel en el que se encuentra el jugador y el porcentaje de aciertos total que obtuvo. A la derecha se muestra la performance del jugador en el nivel actual. Se representan mediante círculos los 10 tríos distintos del nivel y se utilizan colores para indicar el estado de cada uno. Los círculos en color gris son aquellos que aún no han sido jugados, el círculo naranja es el trío que se está jugando actualmente, los círculos verdes son aquellos que el jugador respondió correctamente y los rojos son los que tuvieron una respuesta errónea. Además, debajo de los círculos se indica de forma numérica el puntaje obtenido en el nivel actual.

Finalmente, en el centro de la pantalla, se encuentran dentro de tres círculos las proteínas que componen el trío representadas mediante JSmol. Esta herramienta permite al usuario acercar, alejar y rotar en todos sus ejes las proteínas utilizando el mouse. Cada uno de los círculos posee un botón blanco a su lado para que el usuario pueda seleccionar la proteína que considere correcta.

Una vez seleccionada una proteína, se actualiza la interfaz para mostrar si la decisión del jugador fue correcta o no. Para esto se marca con color verde el círculo correspondiente a la proteína correcta (diferente según SCOP). Cuando la selección del jugador no es correcta, se remarca con color rojo la proteína elegida para indicar el error (Imagen 4).

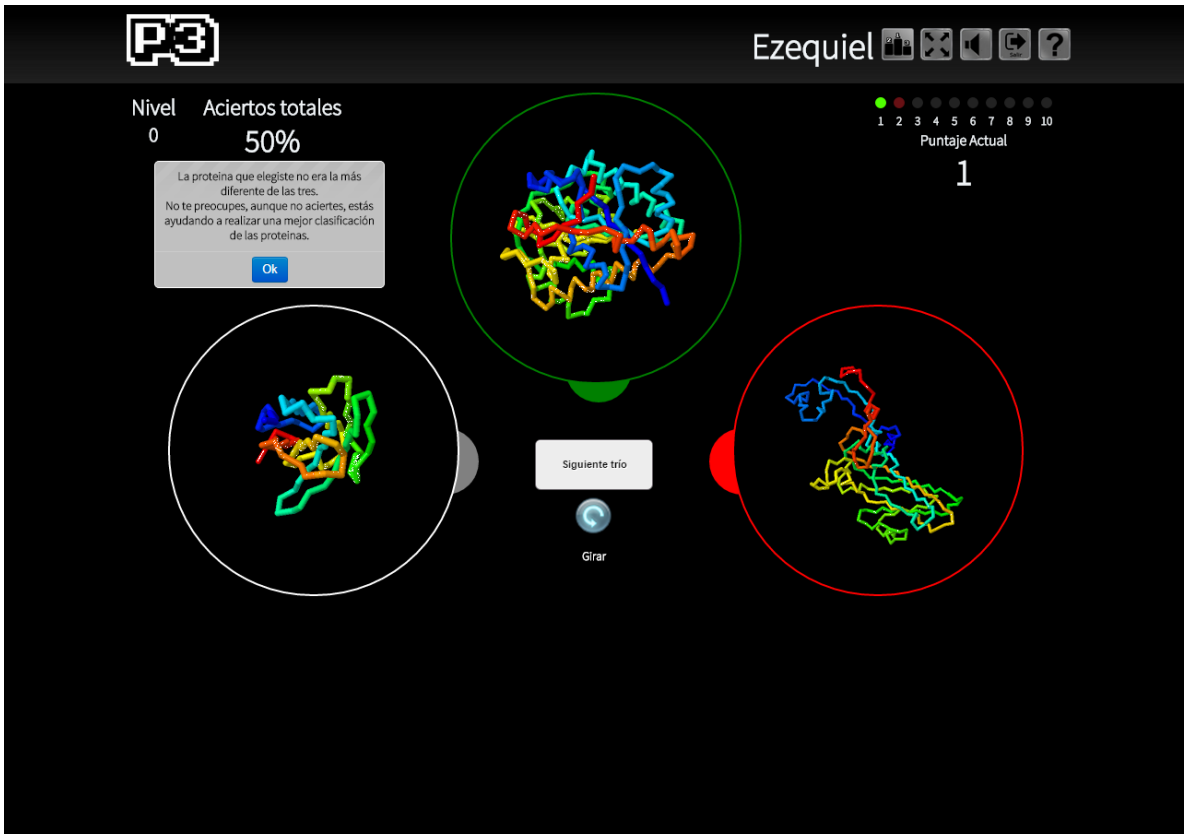


Imagen 5. Pantalla de juego (2)

7.3. ¿Por qué tríos?

Cuando comenzamos a pensar como diseñar un juego que nos permita realizar una clasificación de las proteínas mediante la interacción de los jugadores se nos ocurrieron muchas formas distintas de realizarlo. Las mismas variaron desde disponer de solo una proteína e ir comparándola contra todos los grupos de proteínas de los distintos niveles, hasta disponer de una cantidad de proteínas mayor (por ejemplo 20) y agruparlas de acuerdo a su similitud.

Sería demasiado largo y no es el objetivo de este trabajo contar específicamente todas las opciones que tuvimos en mente y prototipamos hasta definirnos por el modo de tríos, pero podría ser interesante nombrar algunos de los problemas por los cuales descartamos otras opciones y consideramos que esta es la mejor forma de obtener los datos. En cuanto a los modos de juego en donde cada jugador obtiene una proteína y debe ir seleccionando primero la clase, luego el fold, luego la superfamilia y finalmente la familia, el problema que se nos presento es que la cantidad de elementos contra los cuales había que compararlo en cada nivel sería demasiado alta, lo que llevaría demasiado tiempo y además es humanamente imposible en casos donde se dispongan por ejemplo de 200 folds realizar una comparación exacta de todos ellos.

Por otro lado en los modos de juego donde se disponga de un número mayor de proteínas (por ejemplo 20) y se proponga agruparlas de acuerdo a su similitud se nos presentan varios inconvenientes. Uno de los problemas que encontramos es que no es sencillo presentar todas en una pantalla de forma que tengan el tamaño suficiente como para poder comparar sus representaciones (pensando también en la idea de que sea compatible con tablets y celulares). Además al comparar entre una cantidad mayor de objetos, la comparación se dificulta notablemente y la tarea pasa a ser más subjetiva en el sentido de que el límite para considerar a las proteínas como parte de un grupo depende del grado de similitud que considere necesario cada jugador.

Otras opciones como puntuar el grado de similitud entre dos proteínas en un rango (por ejemplo de 1 a 5) también fueron descartadas porque no es fácil mantener una lógica para comparar a todas las proteínas con el mismo criterio.

Estos son solo algunos de los diseños que pensamos y algunas de las razones por las cuales los descartamos. Entre todas estas propuestas, la idea de presentar un trio de proteínas y seleccionar aquella que el jugador considere diferente de las dos restantes nos pareció la más fácil de entender y resolver para el usuario y la más simple de realizar a nivel técnico. Además cada comparación nos dará un dato exacto de la relación entre las tres proteínas que luego será utilizado para clasificar este trío en base al consenso y comparar con la clasificación de SCOP.

7.4. Niveles

Para poder cumplir con los objetivos de lograr un juego atractivo y a la vez producir información que nos ayude a resolver el problema de la clasificación de las proteínas, dividimos el juego en 8 niveles de dificultad. Según Federico Peinado (2012), de la

Universidad Complutense de Madrid, cada nivel es una porción del juego con 1 o más objetivos claros, pensado para jugarse como una unidad, en una sola sesión de juego. En el caso de P3 el objetivo de cada nivel es claro: resolver correctamente una cierta cantidad de los 10 tríos que se presentan en pantalla. La decisión de presentar 10 tríos por nivel se corresponde con el tiempo promedio que tarda un jugador en resolver cada trío, que multiplicado por 10 nos da un número aproximado de entre 5 y 10 minutos, que es el tiempo que nos parece oportuno para cada sesión de juego. De esta forma, terminar con una sesión de juego es una tarea relativamente corta y esto permite una mayor motivación para que los jugadores continúen jugando incluso cuando no dispongan de mucho tiempo.

En P3 utilizamos dos parámetros para definir los niveles y ordenarlos dependiendo de su dificultad. En primer lugar nos basamos en la clasificación estructural provista por SCOP para seleccionar las proteínas del trío de forma tal que sean más o menos parecidas entre ellas y de esta forma generar niveles de distintas dificultades. Particularmente en P3 dispondremos de cuatro niveles de dificultad según este parámetro. Más adelante explicaremos este proceso en mayor detalle. El segundo parámetro utilizado para dividir los niveles es la decisión de representar los movimientos de las proteínas o solamente su estructura.

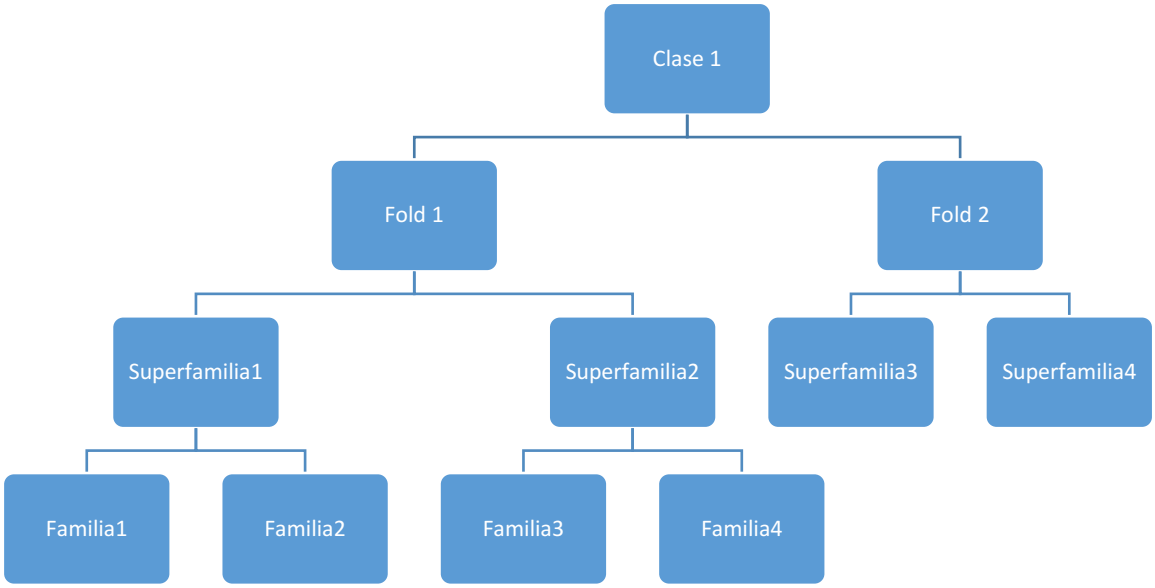


Gráfico 1. Jerarquía de clasificación de proteínas según SCOP

Como ya explicamos, SCOP es un esquema de clasificación de proteínas jerárquico de cuatro niveles: clase, fold, superfamilia y familia. Cada proteína clasificada por SCOP se encontrará catalogada dentro de cada uno de ellos. Por el momento olvidemos la posibilidad de representar los movimientos de las proteínas y consideremos que a partir de estos cuatro niveles de clasificación especificados por SCOP, en P3 diseñamos los niveles de dificultad que se le presentarán al jugador. Para entenderlo de forma más sencilla, comenzaremos por explicar el último nivel de dificultad (llamémosle nivel

4) utilizando el Gráfico 1 como ejemplo. En dicho nivel, el objetivo que nos planteamos es que el jugador pueda identificar y diferenciar las proteínas en el nivel de familia. Para esto a la hora de generar el trío, lo primero que hacemos es tomar del conjunto de proteínas disponibles una proteína al azar, por ejemplo una correspondiente a la Familia1. Una vez que seleccionamos la primera, debemos tomar otra proteína que corresponda a la misma familia que la anterior. Por último, el sistema seleccionará una proteína que pertenezca a la misma superfamilia que las dos previamente elegidas (es decir la Superfamilia1), pero cuya familia sea distinta (en este caso debería ser si o si de la Familia2). Esto nos da como resultado un trio donde dos proteínas pertenecen a la misma familia (Familia1) y la última comparte la misma clasificación hasta el nivel de superfamilia (Superfamilia1), pero pertenece a una familia distinta (Familia2). En lo que concierne al usuario, en general este es el nivel más complicado ya que al estar las tres proteínas dentro de la misma superfamilia, el grado de similitud estructural es alto.

El nivel de dificultad inmediatamente anterior, consiste en lograr diferenciar las proteínas en el nivel de superfamilia. En este caso la primera proteína también será tomada de forma aleatoria; a modo de ejemplo supongamos que la proteína obtenida al azar pertenece a la Familia2. La diferencia será que en este nuevo nivel la segunda proteína deberá pertenecer a la misma superfamilia que la primera (es decir la Superfamilia1) pero a una familia distinta (en este caso deberá pertenecer a la Familia1). Finalmente, la tercer proteína deberá ser tomada del mismo fold que las dos anteriores (Fold1) pero de una superfamilia distinta (necesariamente la Familia2).

Continuando con la misma lógica, en el nivel 2 el objetivo será diferenciar las proteínas según su fold. Para esto comenzaremos de igual forma que en los niveles previos tomando una proteína al azar; supongamos que obtenemos una proteína de la Familia4. En este caso, la siguiente proteína deberá corresponder al mismo fold que la primera (es decir al Fold1) pero a una superfamilia distinta (en el ejemplo solo quedaría disponible la Superfamilia1). Finalmente la proteína que será considerada más diferente a las otras dos, deberá ser seleccionada dentro del conjunto de las proteínas que pertenecen a la misma clase que las dos anteriores (Clase1) pero de un fold distinto al Fold1.

Finalmente llegamos al nivel 1 donde el jugador deberá diferenciar entre las proteínas al nivel de clase. Para esto, la primera proteína será tomada nuevamente al azar, para continuar con el ejemplo de la figura 2 supongamos que corresponde a la Familia2, dentro de la Superfamilia1 y el Fold1. La siguiente proteína deberá ser seleccionada de forma tal que cumpla que corresponde a la misma clase que la primera (Clase1) pero a un fold distinto (solo quedaría disponible el Fold2). Por último, la proteína distinta deberá ser seleccionada de cualquier otra clase que no corresponda a la Clase1.

Al comienzo de esta sección mencionamos que el otro parámetro utilizado para generar los niveles es la posibilidad de incluir la representación de los movimientos de las proteínas. Retomaremos esta idea que nos permitirá explicar la existencia de los 8 niveles disponibles en P3. La decisión que tomamos, es que si bien a partir de la

clasificación de SCOP logramos generar cuatro niveles de dificultad, cada nivel deberá ser jugado por los jugadores en dos modos distintos: solo estructura o estructura y movimientos. En el primer modo, los jugadores verán el trío de proteínas representado a partir de su estructura de forma tridimensional como un cordón multicolor plegado sobre sí mismo; el segundo modo es similar pero además se agrega la representación de los movimientos internos de cada una de ellas.

Como resultado P3 dispone de 8 niveles: los dos primeros niveles seleccionan los tríos de forma tal que el usuario deba diferenciar a nivel de clases; el tercer y cuarto nivel aumenta un poco la dificultad de la tarea ya que el jugador deberá identificar las diferencias a nivel de fold; en los niveles quinto y sexto la tarea consiste en diferenciar las proteínas a nivel de superfamilia; por ultimo en los niveles séptimo y octavo la dificultad aumenta ya que las tres proteínas pertenecen a la misma superfamilia y el jugador deberá identificar las diferencias a nivel de familia. En cada uno de los pares mencionados, el jugador completará uno de los dos niveles visualizando solo la estructura de las proteínas y en el otro se le permitirá ver los movimientos de las mismas. Por motivos de estudio y comparación de los resultados, nos interesa que algunos jugadores jueguen primero los modos de movimientos y luego los de estructura y otros jugadores lo hagan de forma inversa.

7.5. Selección del conjunto de datos

Como mencionamos anteriormente, la PDB posee actualmente la información estructural de más de 100.000 proteínas. Si bien en un futuro P3 podría extenderse para pensar en realizar una clasificación de todas ellas, esto requeriría de una gran cantidad de horas de juego y un número enorme de personas jugando. A los fines de este trabajo, consideramos suficiente realizar una prueba con un conjunto de datos pequeño que nos permita obtener información estadística y resultados a partir de la participación de un menor número de jugadores. Para esto decidimos realizar una selección de proteínas normalizada de forma tal que se presenten proteínas de distintas clases, folds, superfamilias y familias. Para cumplir con las condiciones necesarias para formar un trío es necesario disponer de al menos dos clases, donde cada clase disponga de al menos dos folds, cada fold de al menos dos superfamilias, cada superfamilia al menos dos familias y cada familia al menos dos proteínas. Realizamos una selección aleatoria de proteínas de forma tal que se cumplan estos requisitos mínimos pero asegurándonos de que hayan proteínas de las cuatro clases principales de SCOP.

7.6. Diagrama de clases

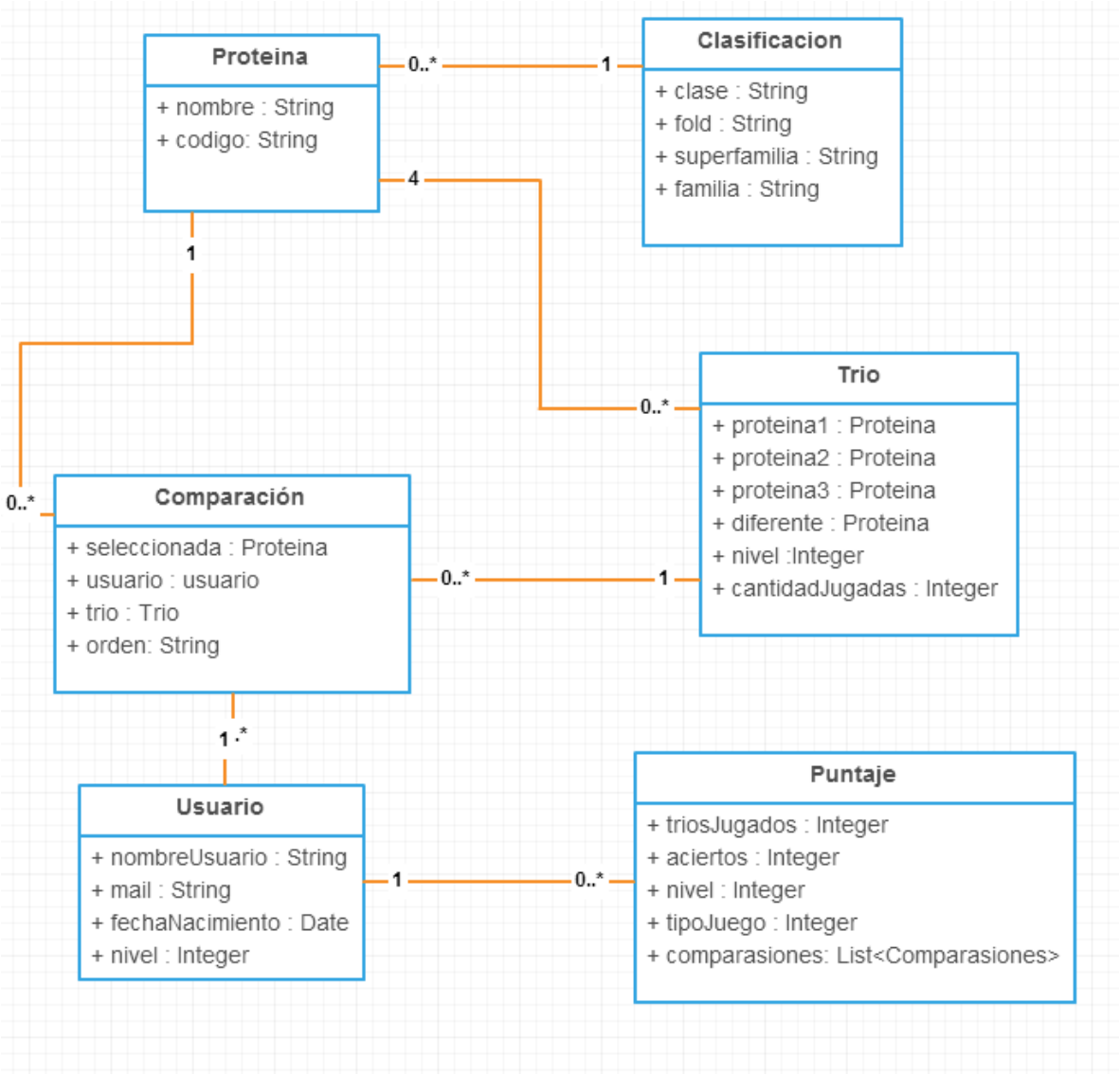


Gráfico 2. Modelo de clases de P3

El modelo de clases de P3 puede verse en el Gráfico 2. A continuación haremos una breve descripción de cada una de las clases para una mejor comprensión de su funcionamiento:

- Usuario: Representa la información del jugador. Almacena sus datos personales y se relaciona con otras entidades para guardar el estado del juego actual y el historial del jugador.
- Proteína: la utilizamos para representar una proteína. Solo almacenamos su nombre y descripción. La información de la secuencia de la misma se encuentra guardada en archivos de texto en el servidor.
- Clasificación: representa la clasificación de SCOP para una proteína dada.

- Trio: representa a un trío. Registra las 3 proteínas que componen el trio (Proteina1, Proteina2, Proteina3), cual es la diferente, a qué nivel corresponde y cuantas veces fue resuelto.
- Comparación: representa lo que seleccionó un jugador en un trío determinado. Tiene un trio, un usuario, una fecha, la proteína seleccionada y el orden en que se mostraron.
- Puntaje: la usamos para guardar los puntos de los jugadores en un nivel.

8. Tecnologías utilizadas

Decidimos implementar un juego disponible en línea para facilitar el acceso de los jugadores desde cualquier lugar que disponga de internet y un navegador web. Además la utilización de tecnologías web como HTML5 y javascript permiten el funcionamiento de la aplicación en la mayoría de los dispositivos modernos. Estas condiciones son importantes debido a que la relación del juego con *Human Computation* implica la necesidad de un número de usuarios relativamente grande para poder resolver el problema de la clasificación de las proteínas; por lo tanto es necesario brindar las mayores comodidades a los jugadores para que el acceso al juego sea rápido y ampliamente disponible.

Las aplicaciones web se basan en el paradigma de cliente-servidor y como su nombre lo indica constan de dos partes. Por un lado, el usuario, quien ejecuta una aplicación en su ordenador local: el denominado programa cliente. Este programa cliente se encarga de ponerse en contacto con una computadora remota (generalmente mediante internet) para solicitar el servicio deseado. Por otro lado, el servidor remoto responderá a lo solicitado mediante un programa que está ejecutando. Este último se denomina programa servidor. En P3 desarrollamos el programa del lado del servidor utilizando el lenguaje de programación Python, en particular, el framework para desarrollo web llamado Django. Para el lado del cliente aprovechamos las ventajas de las tecnologías más modernas del desarrollo web como son HTML5 y javascript que dispone de varias librerías para facilitar el desarrollo, especialmente JQuery y JSmol. JSmol nos permitió realizar la visualización de las estructuras de las proteínas a partir de los archivos obtenidos de la PDB. A continuación explicaremos con mayor profundidad las herramientas utilizadas en el desarrollo de nuestro juego.

8.1. Python

Para el desarrollo del lado del servidor usamos el lenguaje de programación Python. En nuestro caso particular, la buena experiencia previa programando con Python fue una de las razones que nos llevó a elegirlo sobre otros lenguajes.

Python es un lenguaje de programación cuya filosofía hace hincapié en una sintaxis que favorece un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos). La más clara ventaja es que se consiguen buenos resultados con pocas líneas de código, se trata de un lenguaje bastante poderoso.

Python es simple y fácil de entender, tanto para escribir programas como para leerlos posteriormente. A diferencia de lenguajes como C++ o Java la sintaxis de Python es bastante similar al lenguaje natural y por lo tanto mucho más fácil de entender. Esto fue una ventaja fundamental para trabajar en equipo y poder entendernos mutuamente. Por otro lado es sumamente compatible con la arquitectura utilizada y el hecho de que sea multiplataforma nos sirvió para poder trabajar tanto en entornos con sistema operativo Windows como Linux. También es importante destacar la facilidad para desarrollar y realizar las pruebas.

En resumen, Python nos pareció un lenguaje simple, rápido, flexible y legible. Además de todo esto es portable y posee una gran comunidad en internet que facilita la resolución de los problemas que se puedan presentar.

8.2. Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–controlador. La meta fundamental de Django es facilitar la creación de sitios web complejos. Algunas de las características que posee Django y que facilitan el desarrollo son las siguientes:

- Un mapeador objeto-relacional (ORM, del inglés *Object-Relational Mapping*).
- Una API de base de datos robusta.
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URLs basado en expresiones regulares.
- Un sistema "middleware" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

En nuestro caso particular, al principio nos vimos atraídos principalmente por dos características: la posibilidad de desarrollar nuestra aplicación utilizando el paradigma de MVC y la disponibilidad de un ORM que facilite la programación orientada a objetos. Luego de haber utilizado el concepto de MVC en distintos trabajos de la facultad y de experimentar sus ventajas en cuanto a mantener separados los datos, la lógica y la visualización de los mismos, nos pareció la forma más conveniente para desarrollar nuestra aplicación. Junto con esta característica, el hecho de disponer de

un ORM nos pareció una ventaja fundamental para poder centrarnos en el diseño de la aplicación utilizando el paradigma de orientación a objetos. El diseño de la base de datos y el acceso a la misma resultó ser una tarea realmente sencilla gracias a las posibilidades que brinda Django.

8.3. HTML5, JavaScript, AJAX y CSS

Si bien las herramientas que utilizamos para desarrollar el cliente de nuestra aplicación son ampliamente conocidas, queremos realizar una breve descripción de las mismas. HTML5 es la última versión de Hypertext Markup Language, el código utilizado para crear páginas web. Se creó para resolver los problemas de compatibilidad que afectan al estándar HTML4. Fue diseñado para lograr ejecutar prácticamente todo lo que pueda ejecutarse online sin requerir ningún programa adicional como extensiones del navegador. HTML5 puede encargarse de ejecutar desde animaciones, música, películas hasta aplicaciones realmente complejas que corran en el navegador. Otra ventaja es que es libre y multiplataforma, lo que significa que no importa si se ejecutara en una tablet, un celular, una notebook o un smartTV: si el navegador soporta HTML5 debería funcionar correctamente.

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor. Para facilitar aún más la implementación del código en Javascript utilizamos la librería JQuery, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web [34].

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

CSS es un lenguaje de estilos que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas. Además con CSS3 es posible realizar animaciones dentro de las páginas web.

8.4. Jmol y JSmol

Jmol es un visor de estructuras químicas. Jmol devuelve una representación tridimensional de una molécula y puede usarse como herramienta de enseñanza o

para la investigación. Es software libre y de código abierto, escrito originalmente en Java. Su particularidad más importante para nuestros fines es que se puede integrar en páginas web para mostrar las moléculas de muchas formas. JSmol es un framework para Javascript que permite ejecutar una versión de Jmol implementada en HTML5 en lugar de Java. Esto permite representar las estructuras moleculares incluso en los dispositivos que no tienen instalado java y que no lo pueden instalar.

9. Resultados Obtenidos

Para analizar la performance de P3 realizamos pruebas con 30 jugadores de diferentes edades (entre 15 y 50 años) y de ambos sexos, la mayoría de ellos nunca había visto una proteína. Las pruebas fueron supervisadas por nosotros pero sin realizar ninguna intervención. En primer lugar se les dio una breve explicación sobre el objetivo y el funcionamiento del juego resaltando que presten atención a los carteles de ayuda que aparecen en los primeros minutos. Los jugadores comenzaron creando un nuevo usuario y luego se dedicaron a jugar a P3 durante aproximadamente una hora. Los resultados de las pruebas serán analizados a continuación. En el anexo 1 se pueden observar las tablas generadas para analizar los datos obtenidos.

9.1. Análisis inicial de la performance de P3

Consideremos un trio de proteínas (p1, p2, p3) y supongamos que estas fueron elegidas por los jugadores (n1, n2, n3) veces respectivamente, donde n1+n2+n3 es el número total de jugadores que jugaron este trío. Para crear una clasificación propuesta por P3 consideramos que la proteína más votada (*max*(n1, n2, n3)) es la más diferente de las tres. La clasificación de un trío será correcta si el consenso coincide con la clasificación de SCOP (*igualQueSCOP = VERDADERO*) e incorrecta si no coincide (*igualQueSCOP = FALSO*). En el gráfico 3 mostramos la proporción de tríos clasificados correcta e incorrectamente para los distintos niveles de clasificación (*clase*, *fold*, *superfamilia* y *familia*) según los jugadores hayan visto solo las estructuras (estático) o las estructuras moviéndose (movimiento).

Gráfico 3. Proporción de aciertos en comparación con la clasificación de SCOP

Podemos observar que excepto para el caso de familias, P3 coincide con SCOP más veces que las que se equivoca (las columnas verdes son más altas que las rojas). Más aún, en realidad la línea de base sería una elección aleatoria, lo que corresponde a una tasa de éxito de 33,33%. En tal caso la barra roja sería el doble de alto que la barra verde, por lo tanto P3 es mejor (mucho mejor) que una selección al azar.

El análisis anterior considera el acuerdo entre la proteína más votada en P3 con el valor correcto de SCOP sin tener en cuenta que tan bueno es el acuerdo (¿Es la más votada por el 100% de los jugadores o solo por el 50%?) ni la confianza estadística que podemos tener en el resultado, que depende del número de jugadores (un acuerdo del 100% entre 2 jugadores no es tan significativo como entre 20 jugadores). La proporción de votos y el número de jugadores determinan un *p-valor* que cuantifica la significación estadística (probabilidad de error). El *p*-valor se calcula, para el caso de P3, usando el test estadístico binomial. Si queremos ser exigentes, podemos considerar solo los casos en que la predicción de P3 tiene un *p-valor* mejor que 0.01 (que es un valor típico y significa que tenemos una confianza de 0.99 en la predicción). En tal caso el gráfico anterior queda de la siguiente manera (gráfico 4):

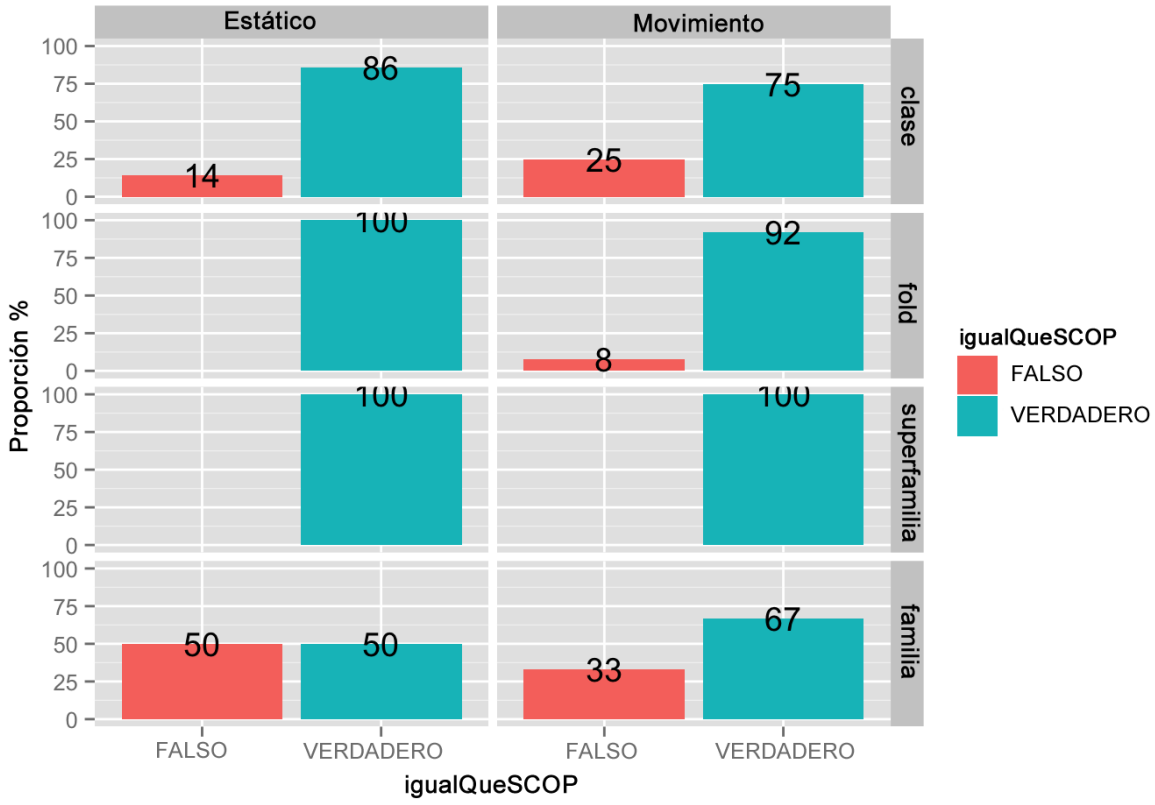


Gráfico 4. Proporción de aciertos en comparación con SCOP, para casos clasificados por P3 con *p-valor* < 0,01

Vemos que en general la clasificación es excelente. Notamos que los valores relativamente bajos para el nivel de clase (que son muy fáciles de clasificar) se deben a que, al ser parte de los primeros niveles, los jugadores todavía están aprendiendo. Lo que más nos interesa es la clasificación de superfamilias, ya que todos los otros niveles se pueden hacer automáticamente. El acuerdo en este caso es notable y muy prometedor.

9.2. Análisis riguroso: curvas ROC

En esta sección haremos un análisis más riguroso y detallado. Las curvas *Receiver Operative Caracteristic* (ROC) se usan para evaluar la performance de clasificadores. Antes de mostrar estas curvas, en el gráfico 5 mostramos, para cada uno de los 80 tríos considerados, el consenso alcanzado por los jugadores. En el gráfico, *consenso* representa la proporción de jugadores que seleccionaron la proteína más votada del trio y sus valores varían entre 0.33 (al ser la más votada al menos tiene un tercio de los votos) y 1 (para el caso en que todos los jugadores seleccionaron la misma proteína). Al igual que en el caso anterior utilizamos el *p*-valor para determinar la probabilidad de que la selección sea errónea. Además, el gráfico muestra en verde los casos en que la proteína del consenso de P3 coincide con SCOP y en rojo el caso en que no.

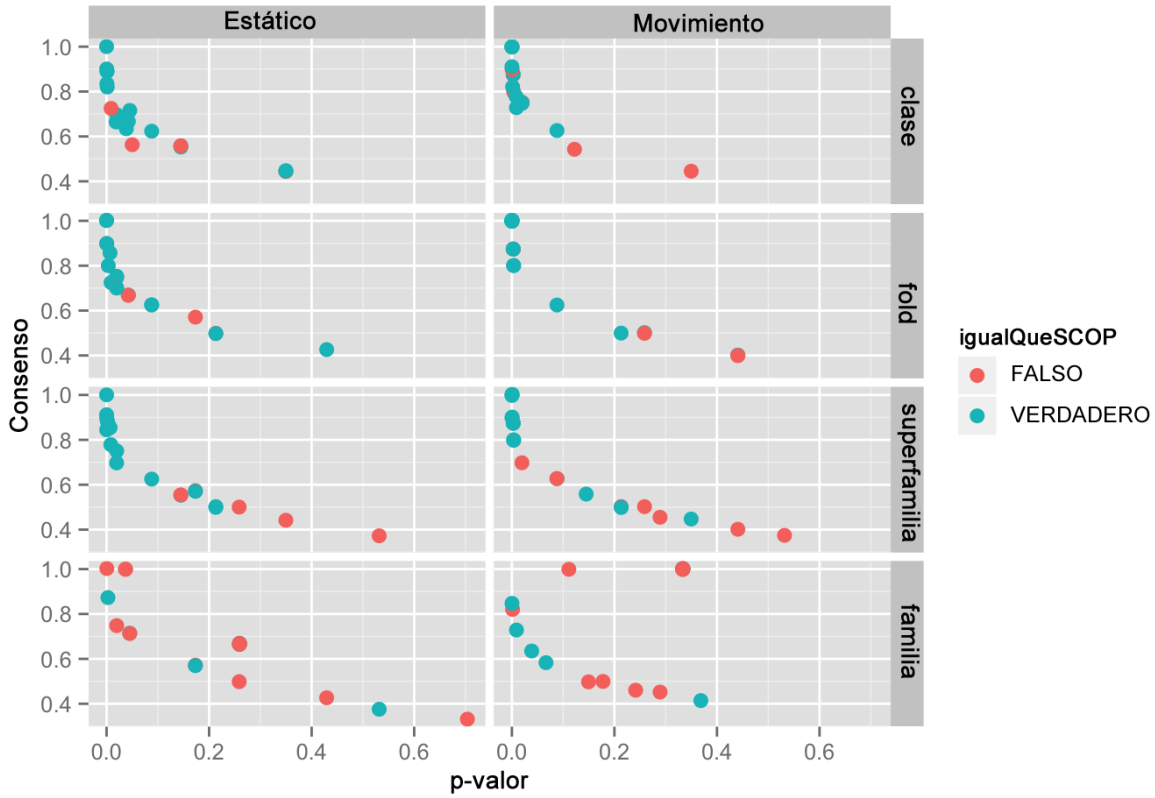


Gráfico 5. Curvas ROC del consenso en función del *p-valor*

A partir del gráfico podemos sacar varias conclusiones. Lo más importante es que a medida que aumenta la confianza (disminuye la probabilidad de error, *p-valor*), la proteína seleccionada por el consenso de P3 coincide cada vez más con la correcta según SCOP: los puntos para valores de *p-valor* cercanos a 0 son predominantemente verdes. Esto es así para las tareas de clasificación de clase, fold, y superfamilia. La excepción se da en el nivel de familia, donde hay muchos más errores.

La idea de las curvas ROC es que podemos variar la performance del clasificador, variando el *p-valor* máximo que toleraremos. Para un *p-valor* máximo determinado, los puntos a la derecha los consideramos "no clasificados" y los de la izquierda "clasificados". De los puntos "clasificados" tenemos casos en que el clasificador coincide con el valor correcto (puntos verdes del gráfico 5) y casos en el que el clasificador difiere del valor correcto (puntos rojos). Los puntos verdes se llaman *True Positive* (TP) y los rojos *False Positive* (FP). La curva ROC es la proporción TP/(TP+FP), a la que se llama *True Positive Rate* (TPR) o "Precisión". La TPR o Precisión es la proporción de predicciones correctas e indica la calidad del clasificador. Para P3 obtenemos las siguientes curvas ROC:

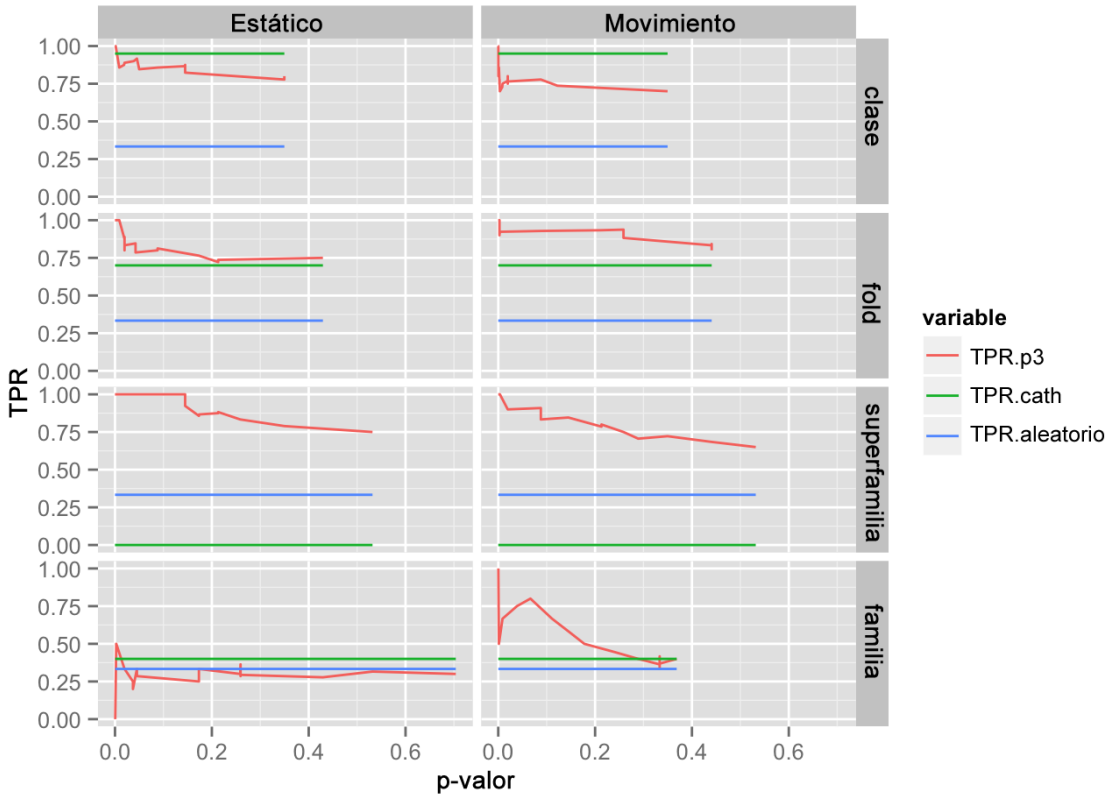


Gráfico 6. Curvas ROC del TPR en función del *p-valor*

Cuanto más arriba esté la curva (cuanto mayor sea el área bajo la curva), mejor es el clasificador. Como referencia pusimos los casos de CATH (utilizamos el acuerdo entre la clasificación de CATH y SCOP como clasificador) y "aleatorio" como líneas de base con las que comparar. Se ve que en general P3 está por encima de ambos (excepto en el caso de familias con tríos estáticos).

9.3. Performance global

Como vimos, la calidad de un clasificador, la Precisión, depende del *p-valor* que se use, lo que se representa en las curvas ROC (Gráfico 6). Un clasificador será mejor que otro si su curva ROC está por encima. Es conveniente contar con un valor único que cuantifique la calidad de un clasificador independientemente del *p-valor*. Para esto, se usa, típicamente, el área bajo la curva ROC (AUC: *Area Under the Curve*) que coincide con la precisión promedio: se calcula la precisión para cada *p-valor* y luego se promedian los valores. Los resultados se muestran a continuación.

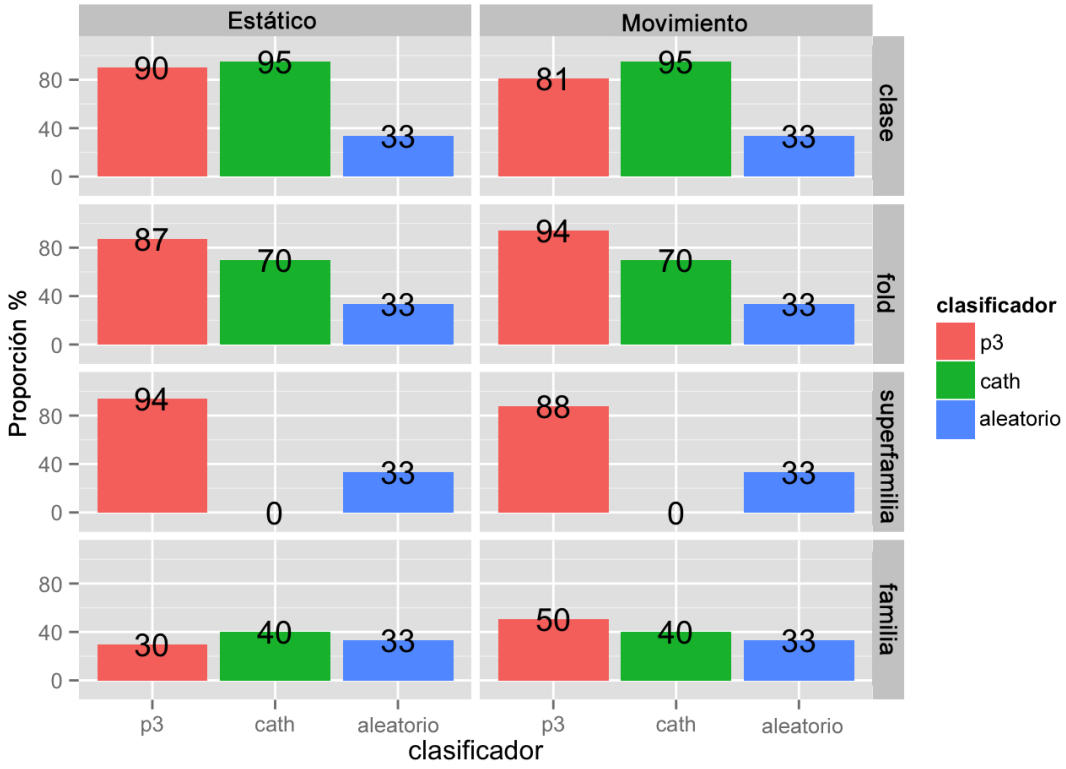


Gráfico 7. Precisión promedio de los clasificadores P3, CATH y aleatorio

En el gráfico 6 vemos que P3 es mejor que "aleatorio" en todos los casos (aún para familias) y mejor que CATH en casi todos los casos. Es notable la gran precisión de P3 en el caso de superfamilias, sobre todo comparada con CATH donde no coincide con SCOP en ningún trío. Más detalladamente, hacemos las siguientes observaciones:

- En el nivel de clase, los jugadores están aprendiendo a jugar, por lo que la performance de P3 es algo menor que la de CATH. De todos modos, este nivel es trivial, ya que es fácil de clasificar automáticamente mediante algoritmos y a partir de pruebas realizadas por más de 300 personas con una versión previa del juego, podemos probar que cuando los jugadores aprenden, clasifican el 100% de los tríos de este nivel de forma correcta.

- A nivel de fold, P3 supera tanto la selección aleatoria como la que se obtiene a partir de CATH. Los resultados son mejores cuando se muestran los movimientos, pero necesitamos más datos para poder verificar esta tendencia.

- A nivel de familias, no esperamos un porcentaje de acuerdo con SCOP muy alto porque este nivel está basado en la comparación de secuencias de amino ácidos, no de estructuras. Notamos sin embargo que cuando se usan movimientos, hay una precisión de 50%, superior al 33% correspondientemente a una selección puramente aleatoria. Será interesante ver si este es un efecto de los pocos datos (pocos jugadores llegaron a este nivel, por lo que los datos son más inciertos) o si se verifica la tendencia.

- Finalmente, lo que más nos interesa: A nivel de superfamilias, la precisión es excelente. Parece haber una mejor performance al usar estructuras estáticas (pero serán necesitamos más jugadores para verificar la tendencia).

10.Conclusión

Los resultados preliminares que fueron analizados en el capítulo anterior demuestran, cómo fue planteado en la hipótesis de este proyecto, que es posible realizar una clasificación de las proteínas con un altísimo grado de acuerdo con SCOP (considerada actualmente como el *gold standard*) a partir del agregado de las selecciones de los jugadores. Esto es particularmente notable porque los expertos de SCOP tienen conocimiento no solo de las estructuras proteicas sino también de la secuencia de aminoácidos y de las funciones biológicas, datos que no estuvieron disponibles para los jugadores de P3 (la mayoría nunca había visto la representación de una proteína).

Una de las claves del éxito de este trabajo es el hecho de haber aplicado prácticamente las técnicas de *Human Computation* al desarrollar P3 como un GWAP. Al ser un juego, brinda la motivación proveniente del desafío de pasar todos los niveles y de la superación de otros contrincantes, la cual aprovechamos para mantener interesados a nuestros jugadores en la resolución correcta de los tríos y de esta forma obtener datos significativos.

En cuanto a la idea de si existe alguna ventaja al agregar información sobre los movimientos aparte de la estructura, las tendencias no están claras y hace falta recopilar más datos. Por el momento no podemos afirmar que a partir de P3 se pueda obtener una clasificación basada en el movimiento de las proteínas y que difiera en gran medida de la clasificación estructural existente, sin embargo sería interesante ahondar en este tema y es una posible línea de investigación para trabajos futuros.

Sería interesante pensar como una tarea a futuro la idea de desarrollar una mecánica de juego que permita resolver la clasificación de las más de 100.000 proteínas de una manera más eficiente y con la participación de una menor cantidad de jugadores. Si bien en el capítulo 9 queda demostrado que la performance de P3 es muy buena y que sería posible realizar una clasificación de todas las proteínas existentes, es verdad que sería necesario un número muy alto de jugadores, por lo que resulta atrayente la idea de realizar una mejora en este sentido.

11.Bibliografía

[1] L. von Ahn y L. Dabbish, «Designing games with a purpouse,» *Communications of the ACM,* vol. 51, nº 8, pp. 58-67, Agosto 2008.

[2] F. Bernstein, T. Koetzle, G. Williams, E. Meyer, M. Brice, J. Rodgers, O. Kennard, T. Shimanouchi y M. Tasumi, «The Protein Data Bank: a computer-based archival file for macromolecular structures,» *Journal of Molecular Biology,* vol. 112, nº 3, pp. 535-542, Mayo 1977.

[3] A. Murzin, S. Brenner, T. Hubbard y C. Chothia, «SCOP: A structural classification of proteins database for the investigation of sequences and structures,» *Journal of Molecular Biology,* vol. 247, nº 4, pp. 536-540, 7 Abril 1995.

[4] L. H. Greene, T. E. Lewis, S. Addou, A. Cuff, T. Dallman, M. Dibley, O. Redfern, F. Pearl, R. Nambudiry, A. Reid, I. Sillitoe, C. Yeats, J. M. Thornton y C. Orengo, «The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution.,» *Nucleic Acids Research,* vol. 35, nº 1, pp. 291-297, Enero 2007.

[5] S. Maguid, S. Fernandez-Alberti, L. Ferrelli y J. Echave, «Exploring the Common Dynamics of Homologous Proteins. Application to the Globin Family.,» *Biophysical Journal,* vol. 89, nº 1, pp. 3-13, Julio 2005.

[6] S. Maguid, S. Fernandez-Alberti y J. Echave, «Evolutionary conservation of protein vibrational dynamics.,» *Gene,* vol. 442, nº 1-2, pp. 7-13, Octubre 2008.

[7] E. Fuglebakk, J. Echave y N. Reuter, «Measuring and comparing structural fluctuation patterns in large protein datasets,» *Bioinformatics,* vol. 28, nº 19, pp. 2431-2440, Octubre 2012.

[8] C. Micheletti, «Comparing proteins by their internal dynamics: Exploring structure–function relationships beyond static structural alignments,» *Physics of Life Reviews,* vol. 10, nº 1, pp. 1-26, Marzo 2013.

[9] E. Law y L. von Ahn, «Input-Agreement: A New Mechanism for Collecting Data Using Human Computation Games,» de *CHI '09 proceedings of the 27th conference on human factors in computing systems*, Boston, 2009.

[10] R. Chandrasekar, E. Chi, M. Chickering, P. G. Ipeirotis, W. Mason, F. Provost, J. Tam y L. von Ahn, «Front matter,» de *KDD '10 The 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , Washington DC, 2010.

[11] M. Yuen, L. Chen y I. King, «A Survey of Human Computation Systems,» de *Computational Science and Engineering, 2009. CSE '09. International Conference*

on , Vancouver, BC , 2009.

[12] K. T. Chan, I. King y M. Yuen, «Mathematical Modeling of Social Games,» de *Computational Science and Engineering, 2009. CSE '09. International Conference on* , Vancouver, BC , 2009.

[13] A. Quinn y B. Bederson, «A Taxonomy of Distributed Human Computation,» de *CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, 2011.

[14] E. Law, K. West, M. Mandel, M. Bay y J. S. Downie, «Evaluation of algorithms using games: The case of music tagging,» de *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, Kobe, 2009.

[15] A. J. Quinn y B. B. Bederson, «Human Computation: Charting The Growth Of A Burgeoning Field,» de *Proceedings of the SIGCHI conference on human factors in computing systems*, Paris, 2011.

[16] Amazon.com, «Mechanical Turk.,» Amazon.com, Inc., 5 Marzo 2015. [En línea]. Available: http://www.mturk.com.. [Último acceso: 13 Marzo 2015].

[17] G. Inc, «reCAPTCHA,» 9 Febrero 2015. [En línea]. Available: https://www.google.com/recaptcha/intro/index.html. [Último acceso: 14 Marzo 2015].

[18] R. Hunicke, M. LeBlanc y R. Zubek, «MDA: A Formal Approach to Game Design and Game Research,» de *Game Design and Tuning Workshop at the Game Developers Conference*, San Jose, 2004.

[19] L. Von Ahn, *Human computation,* Pittsburgh: School of Computer Science, Carnegie Mellon University, 2005.

[20] Center for Game Science at University of Washington; UW Department of Biochemistry, «Fold It: Solve Puzzles for Science,» UW Center for Game Science, UW Department of Computer Science and Engineering, UW Baker Lab, DARPA, NSF, HHMI, Microsoft, y Adobe, 3 Marzo 2015. [En línea]. Available: https://fold.it/portal/. [Último acceso: 4 Marzo 2015].

[21] F. Khatib, F. Dimaio, S. Cooper, M. Kazmierczyk, M. Gilski, S. Krzywda, H. Zabranska, I. Pichova, J. Thompson, Z. Popović, M. Jaskolski y D. Baker, «Crystal structure of a monomeric retroviral protease solved by protein folding game players,» *Nature Structural & Molecular Biology,* vol. 18, nº 10, p. 1175–1177, 2011.

[22] D. Praetorius, «Gamers Decode AIDS Protein That Stumped Researchers For 15 Years In Just 3 Weeks,» *The Huffington Post,* 19 Septiembre 2011.

[23] C. Eiben, J. Siegel, J. Bale, S. Cooper, F. Khatib, B. Shen, P. Foldit, B. Stoddard, Z. Popovic y D. Baker, «Increased Diels-Alderase activity through backbone remodeling guided by Foldit players,» *Nature Biotechnology,* vol. 30, nº 2, p. 190–192, Febrero 2012.

[24] R. Weber, «Play to Cure has already analysed 6 months worth of cancer data,» 7 Marzo 2014. [En línea]. Available: http://www.gamesindustry.biz/articles/2014-03-17-play-to-cure-has-already-analysed-6-months-worth-of-cancer-data. [Último acceso: 20 Abril 2015].

[25] Swiss Institute of Bioinformatics, «What is Bioinformatics,» 23 Marzo 2015. [En línea]. Available: http://www.isb-sib.ch/what-is-bioinformatics.html. [Último acceso: 23 Marzo 2015].

[26] D. A. Benson, I. Karsch-Mizrachi, K. Clark, D. J. Lipman, J. Ostell y E. W. Sayers, «GenBank,» *Nucleic Acids Research,* vol. 40, nº 1, pp. 48-53, Enero 2012.

[27] N. M. Luscombe, D. Greenbaum y M. Gerstein, «What is Bioinformatics? A Proposed Definition and Overview of the Field,» *Methods of Information in Medicine,* vol. 40, nº 4, pp. 346-358, 2001.

[28] N. Ahmad, A. Bind y S. Maheshwari, «Bioinformatics New Era: Introduction and Overview,» *Journal of Computational Intelligence in Bioinformatics,* vol. 4, nº 1, p. 7, 2011.

[29] S. B. Primrose y R. M. Twyman, Principles of Genome Analysis and Genomics, Tercera ed., Malden, MA: Blackwell Publishing, 2003.

[30] W. P. Blackstock y M. P. Weir, «Proteomics: quantitative and physical mapping of cellular proteins,» *Trends in Biotechnology,* vol. 17, nº 3, pp. 121-127, 1 Marzo 1999.

[31] Wikipedia, «Wikipedia: Proteína,» 17 Mayo 2015. [En línea]. Available: https://es.wikipedia.org/wiki/Prote%C3%ADna. [Último acceso: 17 Mayo 2015].

[32] C. Hadley y D. T. Jones, «A systematic comparison of protein structure classifications: SCOP, CATH and FSSP,» *Structure,* vol. 7, nº 9, pp. 1099-1112, 15 Septiembre 1999.

[33] R. Day, D. A. Beck, R. S. Armen y V. Daggett, «A consensus view of fold space: combining SCOP, CATH, and the Dali Domain Dictionary,» *Protein Science,* vol. 12, nº 10, pp. 2150-2160, Octubre 2003.

[34] Wikipedia, «Wikipedia: JQuery,» 20 Enero 2014. [En línea]. Available: https://es.wikipedia.org/wiki/JQuery. [Último acceso: 14 Septiembre 2015].

12. Anexo 1: tablas de datos

12.1. Datos de cada jugador

La tabla 1 contiene los datos de los jugadores. Esta tabla se muestra ordenada según *meanScore*, que es el puntaje promedio de cada jugador, que coincide con la proporción de selecciones del jugador que concuerdan con la clasificación de SCOP.

	player	age	knewProteins	nMoves	maxLevel	meanScore
8	felipe	18	FALSE	70	6	0.46
4	anita	20	FALSE	80	6	0.49
5	bruno	15	TRUE	70	6	0.53
10	joaco	16	FALSE	100	7	0.53
15	sofi	19	FALSE	80	6	0.56
1	Guille	26	FALSE	70	6	0.57
12	jus	21	FALSE	80	7	0.59
6	emma22	26	FALSE	70	6	0.60
9	jmarcos	23	FALSE	80	7	0.60
11	jose	28	FALSE	70	6	0.60
18	toto	20	FALSE	70	6	0.61
13	msofi	24	FALSE	64	6	0.62
2	Ruben	22	FALSE	70	6	0.63
17	toro	16	FALSE	70	6	0.63
7	eze	23	TRUE	81	7	0.63
3	Viqui	24	FALSE	70	6	0.66
16	tincho	26	FALSE	70	6	0.67
14	pcambronera	35	FALSE	70	6	0.69

Tabla 1. Tabla de jugadores de P3

Variables de la tabla de jugadores:

- player: nombre del jugador
- age: edad
- knewProteins: FALSE equivale a que el jugador nunca vio una estructura proteica anteriormente, VERDADERO es el caso contrario
- nMoves: cantidad de tríos jugados
- maxLevel: máximo nivel alcanzado
- meanScore: puntaje promedio de cada jugador, que coincide con la proporción de selecciones del jugador que concuerdan con la clasificación de SCOP

Vemos de la tabla anterior que solo dos jugadores habían visto proteínas con anterioridad y que estos no parecen tener una ventaja competitiva (mejor *meanScore*) que quienes no conocían las proteínas.

12.2. Datos de cada trío

La tabla 2 tiene los datos de cada uno de los tríos. Las primeras filas se muestran a continuación para aclarar el formato.

trio_id	level	type	task	p0	p1	p2	scop	cath	consensus	p.value	n0	n1	n2	score.scop	score.consensus
47	4	estatico	superfamily	d1ej2a_	d1k4ma_	d1f4la2	d1f4la2	4	d1f4la2	0.09	3	0	5	0.62	0.62
46	4	estatico	superfamily	d1krha3	d1n62a2	d1l5pa_	d1l5pa_	4	d1n62a2	0.26	2	4	2	0.25	0.50
45	4	estatico	superfamily	d1hkqa_	d1repc1	d1fnna1	d1fnna1	4	d1hkqa_	0.17	4	1	2	0.29	0.57
44	4	estatico	superfamily	d1dr9a1	d1l6za1	d1hdma1	d1hdma1	4	d1hdma1	0.02	1	1	6	0.75	0.75
43	4	estatico	superfamily	d1k8rb_	d1lfda_	d1ogwa_	d1ogwa_	4	d1lfda_	0.53	2	3	3	0.38	0.38

Tabla 2. Tabla de tríos

Variables de la tabla de tríos:

- trio_id: identificador único del trío
- level: nivel del juego donde se mostró el trío
- type: si las proteínas se mostraron de forma estática o en movimiento
- task: si el trío pertenece al nivel de clase, fold, superfamilia o familia
- p0, p1, p2: proteínas pertenecientes al trío
- scop: proteína correcta según la clasificación SCOP
- cath: proteína correcta según la clasificación CATH
- consensus: consenso de los jugadores de P3 (proteína más votada)
- p-value: *p-valor* del trío
- n0, n1, n2: cantidad de veces que fueron seleccionadas las proteínas p0, p1, p2 respectivamente
- score.scope: proporción de aciertos en comparación con SCOP
- score.consensus: proporción de aciertos en comparación con el consenso

12.3. Datos de cada jugada

La tabla 3 tiene todas las jugadas de todos los jugadores. Las primeras filas se muestran a continuación para aclarar el formato.

	player	type	level	trio_id	p0	p1	p2	scop	cath	selected	task
67	Guille	estatico	4	47	d1ej2a_	d1k4ma_	d1f4la2	d1f4la2	4	d1f4la2	superfamily
68	Guille	estatico	4	46	d1krha3	d1n62a2	d1l5pa_	d1l5pa_	4	d1l5pa_	superfamily
69	Guille	estatico	4	45	d1hkqa_	d1repc1	d1fnna1	d1fnna1	4	d1hkqa_	superfamily
70	Guille	estatico	4	44	d1dr9a1	d1l6za1	d1hdma1	d1hdma1	4	d1dr9a1	superfamily
71	Guille	estatico	4	43	d1k8rb_	d1lfda_	d1ogwa_	d1ogwa_	4	d1ogwa_	superfamily

Tabla 3. Tabla de jugadas

Variables de la tabla de tríos:

- player: nombre del jugador
- type: si las proteínas se mostraron de forma estática o en movimiento
- level: nivel del juego donde se realizó la jugada
- trio_id: identificador único del trío
- p0, p1, p2: proteínas mostradas en la jugada
- scop: proteína correcta según la clasificación SCOP
- cath: proteína correcta según la clasificación CATH
- selected: proteína seleccionada por el jugador
- task: si el trío pertenece al nivel de clase, fold, superfamilia o familia