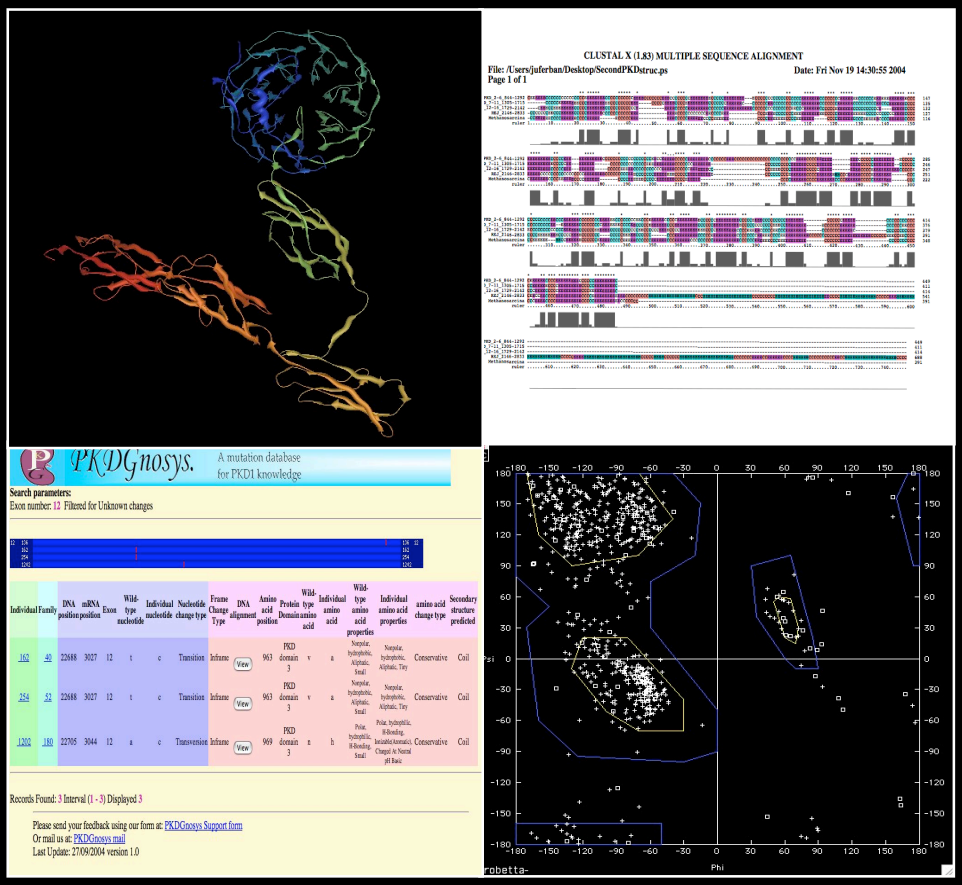
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

“Polycystin 1, PKDGnosys and 3D design”



Julio Fernández Banet

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTAD DE MEDICINA
DEPARTAMENTO DE MEDICINA

"Polycystin 1, PKDGnosys and 3D design"

Memoria presentada por el Licenciado Julio Fernández Banet para optar al Grado de Doctor por la Universidade de Santiago de Compostela.

Noviembre, 2004

El Dr. Xosé Manuel Lens Neo, Profesor Asociado del Departamento de Medicina de la Universidade de Santiago de Compostela

CERTIFICA

que la presente tesis titulada "*Polycystin 1, PKDGnosys and 3D design*" que se presenta para optar al grado de Doctor fue realizada bajo mi dirección por el Licenciado D. Julio Fernández Banet y está en condiciones de ser leída ante el tribunal correspondiente.

Y, para que así conste, firma la presente en Santiago de Compostela a 9 de Diciembre de 2004.

VºBº
Dr. Xosé Manuel Lens Neo

VºBº

Dr. José Ramón González Juanatei

VºBº

Julio Fernández Banet

A mis padres, sin los cuales

nada de esto hubiese sido posible.

¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida mas fácil, nos aporta tan poca felicidad? La repuesta es ésta, simplemente: porque aún no hemos aprendido a usarla con tino.

Albert Einstein

SUMMARY

Autosomal Dominant Polycystic Kidney Disease (ADPKD) is one of the most frequent monogenic inherited diseases.

PKD1 is a complex gene with 46 exons and ~50kb. The protein that it encodes (polycystin 1) contains transmembrane, extra and intracellular domains.

A set of tools was developed to study PKD1 gene and its protein. First step was to implement the PERL-BioPERL program to automate the analysis of PKD1 gene sequences from ADPKD affected individuals, and to annotate every change found within the sequences. Second step was to integrate genotypic information into a relational database. By the use of these tools it was possible to annotate more than 384 nucleotide changes. It was also possible to identify new sequence variants.

Tertiary structure prediction methods were used for a better understanding of the role of polycystin 1. Due to its complexity (36 different domains) the protein was modeled using “ab initio” methods and with the obtained structures it was possible to suggest at least two different functions for polycystin 1.

The extracellular part was suggested to play a role in cell adhesion. This was supported by the folds obtained for the different PKD domains as they appear to be organized in 7-bladed beta-propellers, structure that was found in many proteins with cell adhesion and binding functions.

For the transmembrane domains it was possible to find similarities with proteins that form cation channels and leads to the suggestion that polycystin 1 could be part of a multimeric non specific cation channel.

INDEX

FIGURE INDEX

TABLE INDEX

INTRODUCTION

Las enfermedades quísticas renales comprenden un amplio abanico de desórdenes que varían en la forma de transmisión, anormalidades asociadas, y en el impacto sobre la salud del individuo. La enfermedad quística renal puede ser congénita, heredada o adquirida (GILBERT-BARNES y col., 1990). El número de enfermedades asociadas a quistes en el riñón sugiere la existencia de una respuesta inespecífica de dicho órgano a una gran variedad de alteraciones genéticas y no genéticas. Los quistes renales pueden ser una característica aislada o formar parte de una enfermedad sistémica.

Fenotipo.

La poliquistosis renal autosómica dominante (ADPKD) es la enfermedad monogénica dominante más frecuente en la población, afectando aproximadamente a uno de cada 800 individuos, sin distinción de sexo o raza. Esto significa que en la actualidad en España existen más de 40.000 pacientes con esta enfermedad.

La enfermedad consiste en una progresiva expansión de quistes en ambos riñones, cuyo incremento en tamaño conduce inevitablemente a su pérdida de funcionalidad, llevando, en aproximadamente el 50% de los pacientes, a una terapia de reemplazamiento renal (*Pirson y col*., 1998), alcanzando la insuficiencia renal crónica terminal (ESRD, siglas correspondientes a End Stage Renal Disease) entre la quinta y séptima década de vida (GONZALO y col.,1996)

ADPKD es la causante del 13% de los pacientes que entran en diálisis y necesitan transplante renal, presentando una alta mortalidad y un elevado perjuicio socio-económico debido al elevado coste de los tratamientos sustitutivos. En Galicia, al igual que en el resto del mundo, el número de pacientes que comienzan un tratamiento sustitutivo representan el 13% de los enfermos (*Lens*, 1993).

El principal aspecto clínico de la Poliquistosis Renal Autosómica Dominante es la aparición y crecimiento de un alto número de quistes epiteliales a partir de túbulos renales en los riñones afectos. (Figura 1)

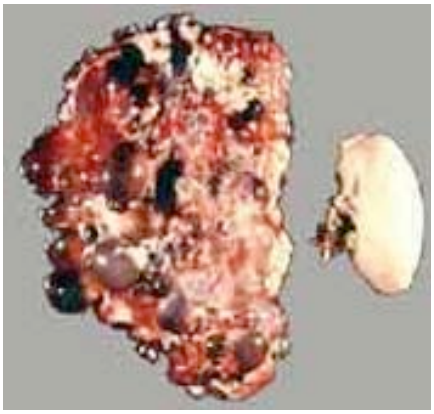


Figura 1 : *Riñón Poliquístico versus Riñón Normal*

ADPKD es una enfermedad sistémica que presenta, además, otras manifestaciones extrarenales: quistes en el hígado, aneurismas intracraneales, anormalidades en las válvulas cardíacas y divertículos de colon. Otras manifestaciones vasculares son: aneurismas torácicos, ilíacos, coronarios y abdominales. (*Torres V.E.,*1999).

Genotipo.

La Poliquistosis Renal Autosómica Dominante es una enfermedad genéticamente heterogénea, habiéndose demostrado, por estudios de ligamiento, la existencia de al menos dos loci, PKD1 (cromosoma 16) y PKD2 (cromosoma 4) clonados en 1995 y 1998, y un hipotético tercer gen todavía por identificar.

El gen PKD1 es el responsable del 85% de los casos de ADPKD (TORRA y col.;1996, PETERS y col.,1992). Dicho gen se ha localizado en el cromosoma 16p13.3. (REEDERS y col.; 1985). El 15% restante es causado por el gen PKD2, localizado en el cromosoma 4q13-23 (PETERS y col, 1993; KIMBERLING y col, 1993). Un porcentaje de pacientes de ADPKD no presentan asociación con mutaciones en PKD1 ni en PKD2 (DAOUST y col, 1995), aunque hasta la fecha no se ha identificado un posible tercer locus PKD3.

La heterogeneidad genética de la poliquistosis autosómica dominante plantea la cuestión de que los fenotipos y presentaciones clínicas de los tres genes sean diferentes. Con esta heterogeneidad parte de la variabilidad interfamiliar existente puede ser explicada por las diferencias en el genotipo. De hecho se ha demostrado diferencia en la edad de aparición de la enfermedad así como en la severidad de ésta, existiendo un pronóstico más favorable en los enfermos con el gen PKD2 alterado frente a los afectos con el gen PKD1 dañado (TORRA y col., 1996; RAVINE y col., 1992).

Descripción de PKD1 y su producto génico la proteína poliquistina 1.

El gen PKD1 presenta una longitud aproximada de 50kb divididas en 46 exones. Su transcrito contiene 14 kb aprox. (The European Polycystic Consortium, 1994).

PKD1 presenta varios genes homólogos, los cuales se encuentran agrupados en el mismo cromosoma, en la región 16p13.1, compartiendo aproximadamente un 95% de homología con el gen PKD1 (Hughes y col, 1995).

La región duplicada del gen PKD1 abarca el 70% de la longitud total del gen, desde el extremo 5' y comprende un total de 33 exones. *Bogdanova y col* (2001) realizaron un estudio sobre la expresión de 5 de estos genes homólogos y observaron que el mRNA producido por estos genes presentaba codones de iniciación subóptimos que no eran transcritos, con lo que concluían que en realidad se trataba de pseudogenes.

La presencia de duplicados, unido al gran tamaño del gen de PKD1 dificulta en gran medida la detección de mutaciones.

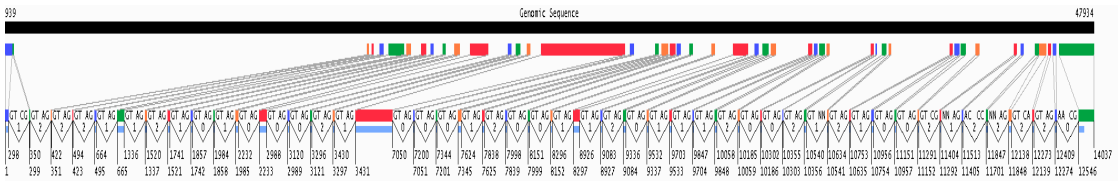


Figura 2 : *Estructura del gen PKD1.*

Aunque en la actualidad todavía no se ha descrito la función del producto génico de PKD1, la poliquistina 1, se han producido avances en el estudio de su estructura y propiedades bioquímicas.

La poliquistina 1, la cual presenta un tamaño de 460 kDa y 4302 amino ácidos, es una glicoproteína asociada a membrana cuyo extremo N-terminal es extracelular y el extremo C-terminal es intracelular. (The International Kidney Disease Consortium, 1995).(Figura 3).

La estructura de la poliquistina 1 incluye en su parte extracelular: 1 amino-terminal, 2 dominios flanqueantes ricos en cisteína, 2 dominios LRR ricos en leucina, 1 dominio WSC y dominio lectina-C-type, un LDL-A, 16 dominios PKD, 1 dominio REJ y un dominio GPS; presenta además 11 dominios transmembrana; en su parte intracelular presenta: 1 dominio PLAT/LH2, 1 dominio coiled-coil y varios dominios protein-quinasa A (Satayner and Zhou,2001).

Se ha encontrado que la poliquistina 1 interactúa con la poliquistina 2, con un regulador de señal de proteínas G, y que puede activar la transcripción de la protein-quinasa C, la proteína de activación 1 y estabilizar la beta-catenina.

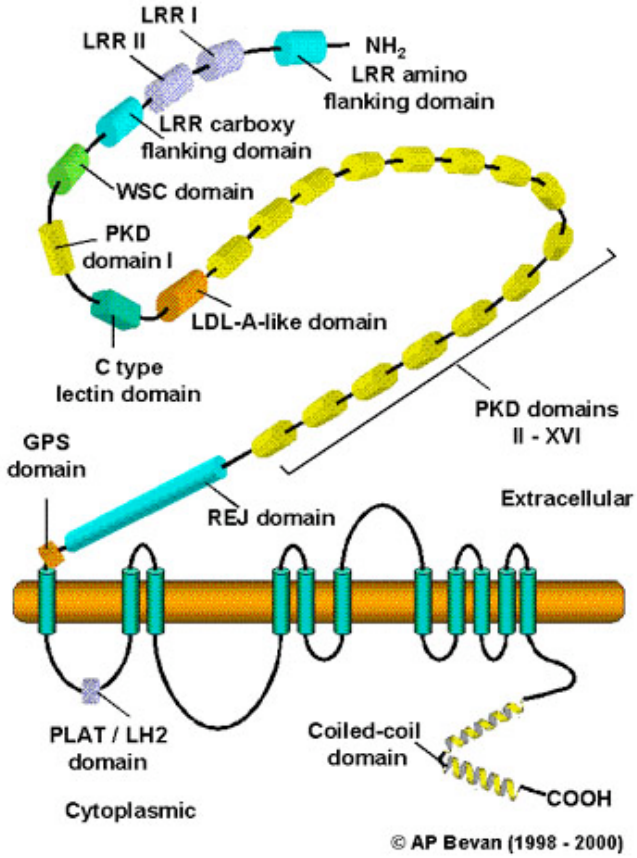


Figura 3 : *Esquema teórico de la Poliquistina 1 y sus dominios.*

1. Leucine-rich repeats (72-125): Repeticiones ricas en leucina indicativas de interacciones proteína-proteína. Recientemente se ha demostrado que están también involucradas en la unión a proteínas de la matriz extracelular (colágeno I, laminina y fibronectina). (Malhas AN, y col. 2002).

2. WSC (177-240): Su nombre viene de las proteínas de la levadura *S. Cerevisiae* donde intervienen en el mantenimiento de la integridad de la pared celular y la respuesta al estrés ambiental. Se piensa que este dominio está involucrado en la unión a carbohidratos (Ponting CP, y col 1999).

3. C-type lectin (403-532): La familia de lectinas tipo-C se unen a un amplio número de ligandos, incluyendo diferentes tipos de carbohidratos, y median distintos procesos biológicos *in vivo,* incluyendo señalización celular y exocitosis. Se ha demostrado que el dominio lectina tipo-C de poliquistina 1 se une *in vitro* a distintos tipos de proteínas de la matriz extracelular. (Weston BS, y col. 2001).

4. LDL-A (639-671): Se encuentra en la porción extracelular de numerosas proteínas y se piensa que son regiones de unión a ligando. Hasta la fecha no hay estudios centrados en los potenciales ligandos de este dominio en la poliquistina 1.

5. PKD repeats (domain 1: 273-356, domains 2-16:851-2145): Un hecho único y distintivo de poliquistina 1 es la presencia de 16 copias de los dominios PKD inicialmente llamados “Ig-like”. La primera repetición PKD se encuentra situada entre los dominios lectina tipo-C y LDL-A; las quince restantes se encuentran después del LDL-A y están unidas por 5 a 7 amino ácidos espaciadores. Mediante análisis de resonancia magnética nuclear se demostró que la estructura del primer dominio PKD de la poliquistina 1 humana tiene un plegamiento característico denominado “Ig-like” fold. La similitud de los patrones de los residuos aminoacídicos de los dominios 2 a 16 condujo a pensar en la

existencia de una misma estructura secundaria para los dominios restantes. (Bycroft, M y col. 1999). Análisis iniciales de la secuencia primaria de la poliquistina 1 mostraron que al menos dos de los dominios PKD poseen una significativa similitud evolutiva con el I-set de los dominios Ig encontrados en moléculas de adhesión celular y receptores de adhesión. (Hughes J, y col. 1995). Sin embargo, basándose en los datos estructurales, Bycroft *et col*. propusieron que los dominios PKD son completamente distintos a la familia de las inmunoglobulinas.

6. REJ (2146-3109): muestra una alta identidad de secuencia con el receptor de la gelatina del huevo en el erizo de mar (suREJ) (Moy GW y col, 1996). Esta identidad hace pensar que el módulo REJ está involucrado en la regulación de procesos de transporte que controlan el influjo de calcio. (Ikeda M, y col, 2002).

7. GPS (3012-3060): también conocido como dominio latrophilin /CL like. Estudios recientes demuestran que la poliquistina 1 es cortada endógenamente en este dominio en un proceso que requiere el módulo REJ (Quian F, y col. 2002).

8. PLAT (3118-3232): Se encuentra entre el primer y el segundo dominio transmembrana, en la región intracelular. Posee homología con la familia de las lipoxigenasas y las lipasas. Se piensa que el dominio PLAT de la poliquistina 1 está involucrado en interacciones proteína-proteína y lípido-proteína. (Ponting CP, y col. 1999).

9. Coiled-coil (4193-4248): La presencia de este tipo de motivo en la porción C-terminal de la poliquistina 1 (así como poliquistina-2) predicen la interacción entre ambos tipos de poliquistina en esta región. (Tsiokas L, y col 1997).

10. Dominios Transmembrana: De función todavía no descrita, se cree que pueden tener un papel como canal de iones al presentar una estructura secundaria similar a la de otros canales conocidos.

Biología funcional.

Desde el punto de vista funcional, se conoce que además de la interacción entre la poliquistina 1 y la 2, existe una interacción con proteínas G hetero-triméricas y con el regulador de señal de la proteína-G (RGS), activando la transcripción de la proteína de activación 1 (AP 1), la proteína quinasa C (PKC) y estabilizando la beta-catenina.

También actúa sobre proteínas asociadas con el citoesqueleto (HAX-1) y con proteínas asociadas con CD2.

Tanto la poliquistina 1 como la 2 forman un complejo integrado en un canal de Ca+2 y la alteración homeodinámica en Ca+2 intra y extra celular tiene importantes efectos en diversas funciones celulares, incluyendo la secreción, la síntesis proteica, la expresión génica, la regulación del ciclo celular y la apoptosis, varias de las cuales presentan una manifestación anormal en el epitelio de los quistes renales.

Un modelo funcional in vitro, para el estudio del impacto de las mutaciones por transfección de cDNA del gen PKD1 en células MCKD está disponible a través del grupo de investigación de Germino. En este modelo, células transfectadas con PKD1 presentaban tubulogénesis espontánea, un incremento en el grado de proliferación celular y una mayor resistencia a la apoptosis. Los mecanismos de este comportamiento son todavía desconocidos.

Se ha demostrado la expresión de poliquistina 1 en órganos que no resultan afectos en la Poliquistosis Renal. Así se ha encontrado expresión de poliquistina 1 en: miocardio, cerebro, epitelio intestinal, ciertas células endoteliales, plexo neural y páncreas (GENG y col., 1996; GENG y col., 1997).

La expresión de la poliquistina 1 está localizada en el epitelio del túbulo renal, ducto biliar hepático y ducto pancreático, siendo mayores los niveles de expresión en el riñón fetal que en el adulto.

Se ha constatado un aumento de la expresión de poliquistina 1 en muchos quistes, pero no en todos, de riñones de individuos afectos con ADPKD.

Diagnóstico

Desde que se instauró la ecografía como método fiable de detección de quistes renales, esta técnica no invasiva es el método actual más utilizado para diagnosticar ADPKD (HOGEWIND y col.,1980). Sin embargo, con los criterios ecográficos establecidos por Ravine (RAVINE y col., 1994), ADPKD solo puede ser excluida con certeza después de los 30 años de edad. Esto contrasta con el análisis de ligamiento que puede realizarse incluso antes del nacimiento, siempre que se proporcione un número suficiente de familiares para el estudio. El análisis de mutaciones será el método de evaluación en el futuro, ya que permitirá diagnosticar a individuos de riesgo en los que no sea posible realizar análisis de ligamiento.

El avance en las técnicas de diagnóstico ha permitido descartar la exclusividad de esta enfermedad de la etapa adulta al haber hecho posible la detección de casos infantiles (ARICETA y col., 1999; CORDAL y col., 1999).

Bioinformática

Se puede hablar de un origen de la bioinformática o biología computacional a finales de los años 60, época en la que surgen una serie de desarrollos fundamentales como el primer algoritmo de alineamiento (Gibbs and McIntyre, 1970; Needleman and Wunsch, 1970), las primeras matrices de substitución preferencial de amino-ácidos en secuencias proteicas (Clarke, 1970; Epstein, 1967), los primeros estudios formales sobre estructura primaria de proteínas (Krzywicki and Slonimski, 1967), y otros.

En la década de los 70 y como consecuencia de los descubrimientos hechos en los años anteriores se propusierin toda una serie de problemas computacionales en biología molecular como la predicción de la estructura del RNA (Tinoco et al., 1971), nuevos métodos de alineamiento de secuencia (Beyer et al., 1974; Gibss et al.,1971; Sackin, 1971), también se realizaron los primeros análisis filogenéticos de familias de macromoléculas (Wu et al., 1974), las primeras simulaciones de regulación metabólica (Heinrich and Rapoport, 1977) , etc. Pero el avance más importante se produjo hacia el final de la década cuando se recopilaron para su almacenamiento y distribución en bases de datos públicas las primeras secuencias de proteínas (Dayhoff,

1978) y estructuras proteicas (Bernstein et al., 1977), bases de datos que se ampliarían enormemente en el futuro.

En la década de los 80 se podría decir que es cuando la biología computacional toma consciencia de si misma y se establece como disciplina independiente, con sus propios problemas y logros.

En 1980 había quedado clara la necesidad del análisis computacional de secuencias nucleotídicas para una mejor comprensión de la biología (Gingeras and Roberts, 1980).

Se desarrollaron algoritmos eficientes para el análisis del creciente volumen de información y se pusieron a disposición de la comunidad científica. También se observó la primera actividad comercial alrededor de estos desarrollos (Devereux et al., 1984).

En esta época se produce también una subdivisión del área en cuatro campos:

(i) Análisis de secuencia. Se desarrollan algoritmos clave como el algoritmo Smith-Waterman para el alineamiento de secuencias por programación dinámica (Smith and Waterman, 1981) o la familia de algoritmos para búsqueda en bases de datos FASTA (Lipman and Pearson, 1985; Wilbur and Lipman, 1983).

(ii) Bases de datos moleculares: Comienza la fase inicial en el desarrollo de bases de datos y aparecen dos de los mayores servidores de información nucleotídica GenBank (Bilofsky et al., 1986) y EMBL Data Library (Hamm and Cameron, 1986).

(iii) Predicción de estructuras proteicas: El campo del análisis y predicción de estructuras proteicas experimentó un crecimiento significativo en esta década. Se experimentó con varios métodos para la representación y visualización de estructuras proteicas incluyendo entre otras la derivación de coordenadas a partir de estereo-diagramas (Rossmann and Argos, 1980) y la definición de dominios (Rashin,1981).

(iv) Evolución molecular: La evolución de proteínas también se convirtió en un área de investigación fundamental con numerosos descubrimientos como los cambios coordinados de residuos

(Altschuh et al., 1988) o la relación entre la divergencia de secuencias y estructura (Chothia and Lesk, 1986).

En los años 90 surgen y se popularizan tecnologías como internet, los sistemas de ventanas (Apple, X-windows), bases de datos, como Genbank y MedLine que se distribuyen a través de CD-ROM y aparecen lenguajes de programación interpretados como perl y python. En lo que a desarrollos científicos se refiere, se hacen disponibles herramientas como BLAST (Altschul et al., 1990) y RasMol (Sayle and Milner-White, 1995).

Protein Structure

Proteins are non-linear molecules composed by a succession of amino acids (a.a.). Rather, this string of a.a. folds into a three-dimensional structure unique for each protein.

One of the major goals in bioinformatics is to understand the relationship between a.a. sequence and tertiary structure (three-dimensional structure) of a protein, as protein function depends on three-dimensional structure.

Protein structure can be studied at four different levels:

- Primary structure: Also known as “Covalent structure”, refers to the linear sequence of amino acids. (With the exception of disulfide bonds, the other structural levels involve non-covalent interactions).

  Amino acid sequence is defined by the gene that encodes it. The gene is transcribed into mRNA and mRNA is translated into protein.

- Secondary structure: This structural level is defined by hydrogen bonding within the peptide backbone.

  The two most common secondary structure elements are alpha helix and beta sheets but there are other structures like loops or coiled-coils.

  - Alpha helix: The alpha helix is the most common type of secondary structure, being the 3.6 helix the most usual type. This helix contains 3.6 a.a. per turn with an H-bond formed every fourth residue. The normal length of the helix is 10 a.a. that correspond to 3 complete turns. The normal location of an alpha-helix is at the surface of proteins where they provide an interface with the aqueous environment. The inner facing side of the helix tends to have hydrophobic a.a. while the outer facing side tend to be hydrophilic. (The third a.a. of each set of four uses to be hydrophobic). Alpha helix present in cell membranes presents a higher amount of hydrophobic a.a.

    Computer programs can detect alpha helix, as there are certain a.a. more prone to be present within these structures (A, E, L, M).

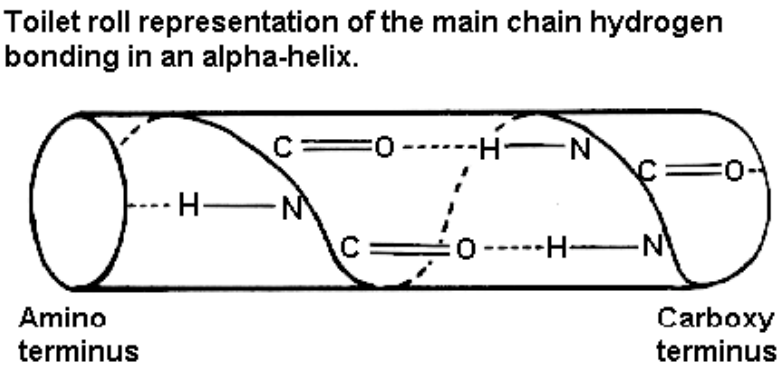


Figure 4: Graphical representation of an alpha helix.

- Beta sheet: Formed by 5 to 10 consecutive a.a. that link with another 5-10 a.a. further down the chain via hydrogen bonds (H-bonds). The regions can be adjacent, separated by a short loop in between, or far apart, with other structures in between. Beta sheets can be parallel, antiparallel or form mixed sheets (succession of parallel and antiparallel chains). Amino acids in the interior strands of the sheet form two H-bonds with neighboring a.a. while each a.a. on the outside strand forms only one H-bond.

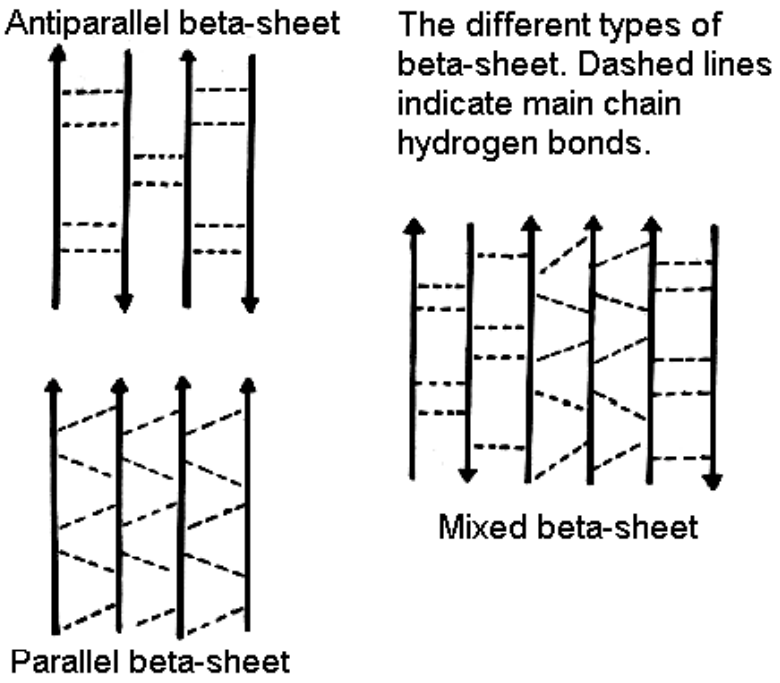


Figure 5: Types of beta-sheets.

  - Loops: Are regions that are between sheets and helix. Loops can present different three-dimensional structures and they can be of various lengths. As loops use to be found on the surface of the protein, a.a. within these structures are not as constrained by space and environment as a.a. in the core region, so more substitutions, insertions or deletions are expected in this area. Loops are also frequently found as components of the active site of the protein (as they use to have charged and polar a.a.).
  - Coil: Is a region of secondary structure that is not a helix, a sheet or a turn.

- Tertiary structure: is the “global” folding of a single polypeptide chain. It describes how the polypeptide chain folds, assembling the different secondary structure subunits in a particular arrangement.

  One of the driving forces that determines how a protein folds is the hydrophobic effect. The polypeptide chain folds in such a way that nonpolar a.a. get hidden within the structure while the side chain of polar residues is exposed to the outer surface.

  H-bonds involving groups from the carbon backbone and the side chain take part into the stabilization of the tertiary structure.

  Disulfide-bonds between cystein residues are also important for stabilizing tertiary structure.

  Insights into three-dimensional structure are of great help when planning experiments aimed at the understanding of protein function and during drug design process

  Depending on the secondary structural elements that are part of tertiary structure, folds can be classified in different topologies:

    - All Alpha:
      - The lone Helix: Small peptides that consist of a single helix
      - The helix-turn-helix: Anti-parallel helix connected by a small loop.

- The four-helix bundle: Corresponds to four helix connected by three loops or two plus two helix that are set together.

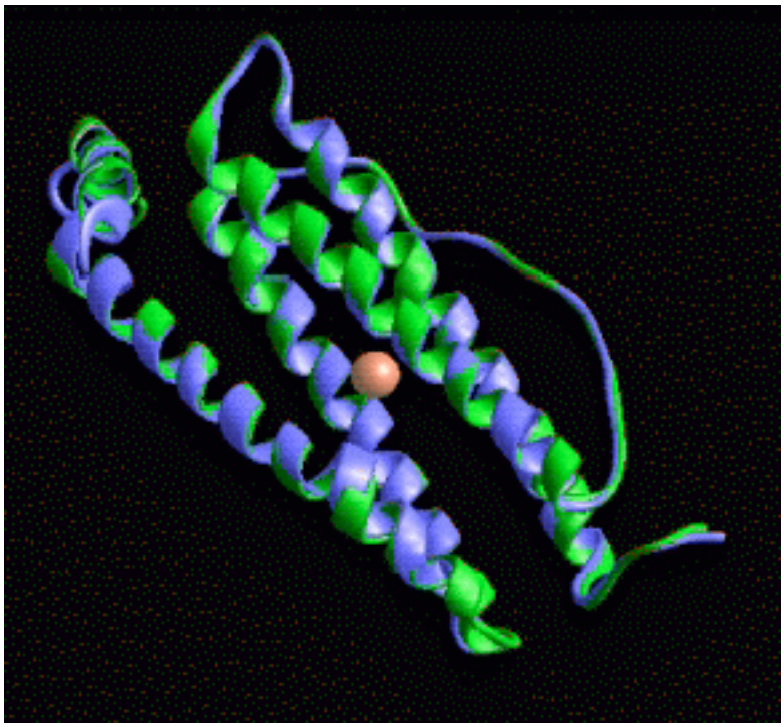


Figure 6. All-Alpha protein.

- Other.

- All beta:
  - Beta-sandwiches: Two beta-sheets packed against each other.
  - Beta-barrels: Antiparallel beta-sheets that form a circular structure

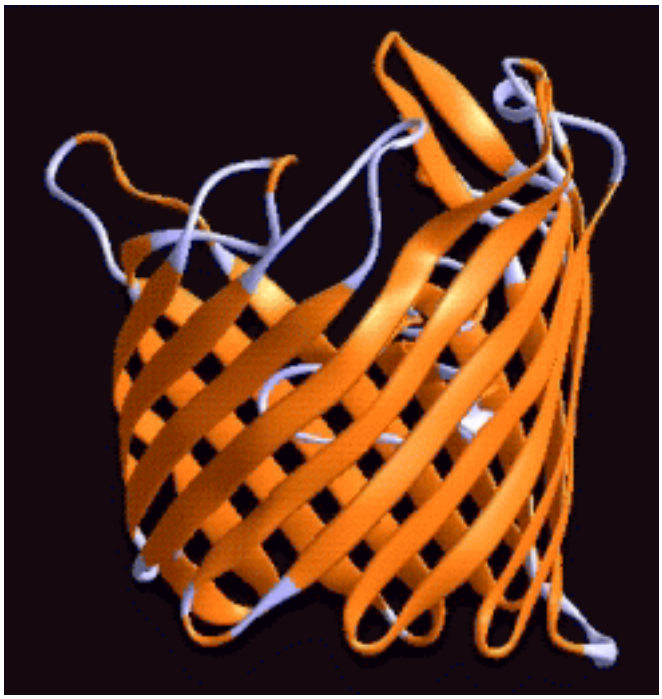


Figure 7. beta-Barrel.

- Up-and-Down antiparallel: beta-sheets connected by loops. It's the simplest structure conformation.
- Beta-propellers: Is a superbarrell composed by 5-8 four stranded beta-sheets.
- Beta-trefoils: Fold with three axis of symmetry
- Beta-helix: The beta-strands wind round the structure describing a helical topology

- Alpha/beta:
  - Alpha-beta horseshoe: Structure with a horseshoe shape where the beta-sheets are on the inside and the helix are on the outer part.

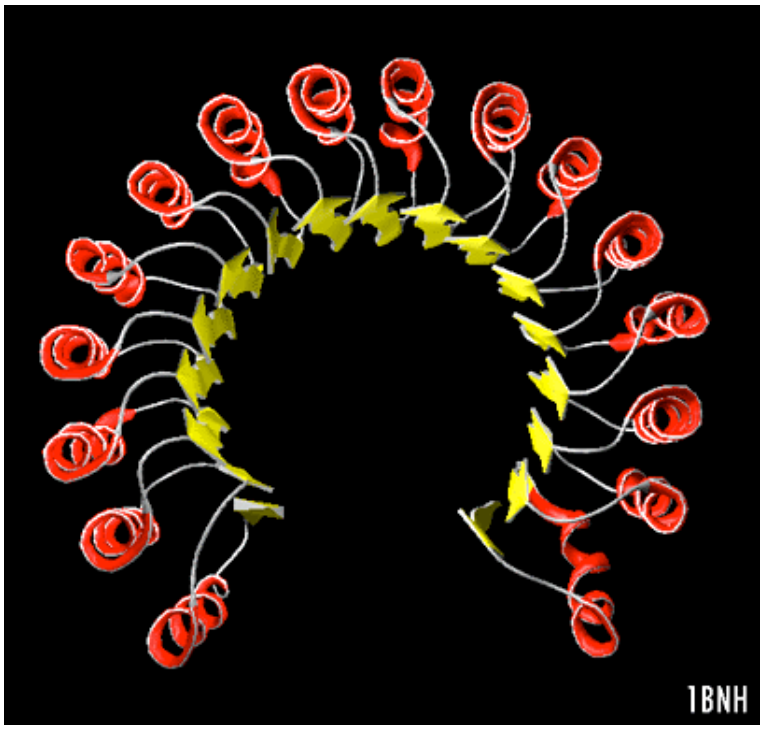


Figure 8: Horseshoe structure.

- Alpha/beta barrels: This structure contains alpha and beta structures that are situated in one plain. Alfa structures can be on one or both sides of the structure depending on the beta-sheet (if it have a reverse point or not).
- Alpha+beta: This is where all those folds which include significant alpha and beta secondary structural elements are collected, but for which those elements are 'mixed', in the sense that they do NOT exhibit the wound alpha-beta topology.

- Small disulphide-rich folds: The members of this family contain large number of disulfide bonds, which stabilize the fold.

There are three different techniques applicable to obtain the tertiary structure of a protein. These are:

- Crystallography

  This technique exploits the fact that crystals diffract X-rays. X-rays have the proper wavelength to be scattered by the electron

cloud of an atom of comparable size. Based on the diffraction pattern of the periodic assembly of molecules or atoms in the crystal, the electron density map can be defined. Phase information must be extracted either from the diffraction data or from supplementing diffraction experiments to complete reconstruction. The model is then built into an experimental electron density map.

Crystallography can provide an answer to some structure related questions (global fold, atomic details of bonding) with no size limitation for the studied molecule.

On the other side, crystallographic studies need a good crystal to be found which is not always possible, and almost no information about dynamic behavior of the molecule can be obtained from one single diffraction experiment.

- NMR

  Nuclear Magnetic Resonance is a phenomenon, which occurs when the nuclei of certain atoms are immersed in a static magnetic field and exposed to a second oscillating magnetic field. The atoms that experience this phenomenon are those that have a property called spin. Spin can be thought of as a small magnetic field that causes the nucleus to produce the NMR signal.

  NMR spectroscopy has been used to study chemical structure using one and second dimensional techniques. Time domain NMR spectroscopic techniques are used to probe molecular dynamics in solutions such as reaction kinetics.

  The limitations of NMR spectroscopy result from the low inherent sensitivity and from the high complexity and information content of NMR spectra.

  Until today NMR spectroscopy can determine the structure of proteins with a mass of up to 30 kDa.

- Computer Protein Modeling:

  This method is based on the attempt to infer the three-dimensional folding of a protein using computer simulation.

  Theoretical protein modeling provides low-resolution models that cannot be used for detailed studies of protein-ligand interaction but hold enough essential information about the spatial arrangement of important residues to guide the design of experiments. These models can also be used for function dilucidation of proteins with unknown function.

  Methods can be classified in three different groups:

  - Homology/Comparative Modeling: There are striking similarities between the three-dimensional structures of some proteins. Comparative modeling exploits these structural similarities between proteins by constructing a three-dimensional structure based upon the known structure of one or more related proteins.
  - Threading: Also known as fold recognition, this method may be used to suggest a general structure for a new protein. When a new protein has a sequence identity with other proteins lower that 20-30%, which makes the use of comparative modeling techniques inappropriate, then threading can be useful to solve such a problem. Threading consists in checking the polypeptide chain against all known structural conformations (the motivation of threading depends on the idea that it does appear to be a finite set of protein folds).
  - Ab Initio: This approach to protein modeling attempts to solve structure from first principles. The problem is to explore the conformational space of the molecule in order to identify the most appropriate structure. As the number of conformations tends to be very high, it is usual to try to find only the lowest energy structures.

- Quaternary structure: Consists in the association of two or more polypeptide chains to form a multiple-subunit structure resulting in an active functional unit.

  This structural level is not present in every protein as not every protein functional unit is formed by a protein complex.

  Quaternary structures are stabilized by all types of non covalent interactions (hydrogen, van der Walls, ionic) but in rarer instances, disulfide bonds between cystein residues of different chains are involved in complex stabilization.

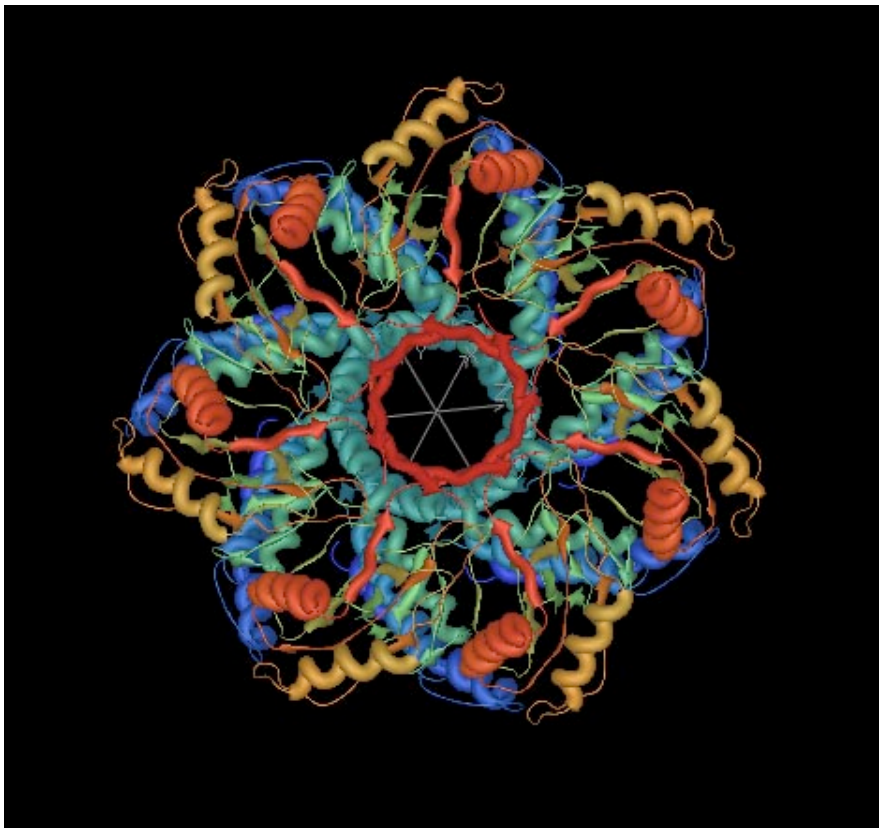


Figure 9: Quaternary structure of 1mxm formed by a complex of 7 copies of the same a.a. chain.

Structure Validation Methods

The Ramachandran Plot

One way to check about the feasibility of a protein structure is the use of the Ramachandran plot (Ramachandran et al., 1963).

Ramachandran plot is based on the polypeptide chain property that says that the main chain N-Calpha and the Calpha-C bonds are relatively free to rotate. The rotations represent the torsion angles phi and psi.

By the use of computer models of small polypeptides with different phi and psi value combinations it was able to determine stable conformations. For each conformation, the structure was examined for close contacts between atoms (based on their van der Waals radii). Those phi and psi conformations, which cause atoms to collide, correspond to sterically disallowed conformations of the polypeptide backbone.

A graph was generated with the allowed and disallowed regions and this graph can be used for a fast and easy check of the quality of a protein structure.

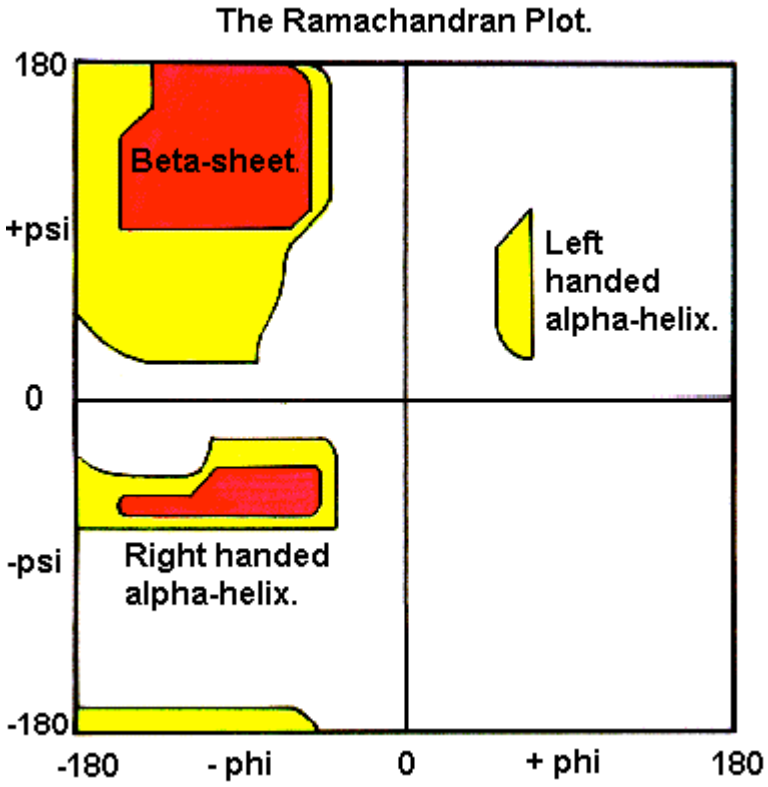


Figure 10: Ramachandran plot. The red and yellow areas show the allowed conformations, being the red areas those where the Van der Waals radii used was more restrictive.

PROCHECK

PROCHECK is a program to check the stereochemical quality of protein structures (Laskowski et al., 1993). It generates various plots and a comprehensive residue-by-residue listing, providing an assessment of the

overall quality of the structure as compared with well refined structures of the same resolution and highlighting regions that may need further investigation.

WHAT-IF

WHAT-IF is a computer program written to aid macromolecular modeling and drug design (Vriend, 1990). WHAT-IF protein validation portion, WHAT-CHECK, checks for clashes between symmetry-related molecules, and includes the determination of a quality factor assessing the distribution of different atom types in the environment around side chain fragments. The expected distributions are compiled from a data set of high resolution structures (Vriend et al., 1993).

Goals

- To develop a software tool to automatize the alignment and analysis of the data obtained by direct sequenciation, and parsing of the information about changes.
- Implementation of a relational database to integrate genotypical and phenotypical information of each individual.
- Generation of a Web Interface to ease access to information contained in the database.
- “Ab initio” modeling of the tertiary structure to define functional regions of the protein.

MATERIALS Y METHODS

CRITERIOS DE INCLUSIÓN

La mayor parte de los pacientes seleccionados para el estudio provienen del área de influencia del Complexo Hospitalario Universitario de Santiago de Compostela, aunque también se han incluido familias de otras zonas de Galicia, principalmente Vigo. Varias familias son de origen belga y han sido proporcionadas por el grupo del Dr. Devuyst de la Universidad Católica de Lovaina, Bruselas.

Antes de incluirlos en el estudio los individuos fueron debidamente informados y se obtuvo el consentimiento explícito de los pacientes.

Todos los individuos fueron sometidos a un análisis ecográfico con el fin de determinar su estado clínico. Se tuvieron en cuenta características especiales de la enfermedad tales como la penetrancia dependiente de la edad. Un individuo se consideró afecto si presentaba un quiste bilateral o dos quistes unilaterales en un rango de edad de 0-30, si presenta al menos dos quistes en cada riñón entre los 30-60 años, y un mínimo de cuatro quistes por riñón en personas mayores de 60 años. También se consideró la presencia de quistes en órganos como el hígado, el páncreas y los ovarios, así como alteraciones aórticas.

OBTENCIÓN DEL ADN

El DNA genómico fue extraído a partir de muestras de sangre de cada paciente. Para la extracción del DNA genómico se empleo el kit comercial PureGene (Gentra) siguiendo instrucciones del fabricante. Una vez extraído el DNA se diluye a 100 ng/µl con agua destilada y se mantiene a –20ºC para su posterior utilización.

ANÁLISIS DE MUTACIONES EN PKD1

Secuenciación del gen PKD1.

Para evitar la secuenciación de alguna de las secuencias homólogas presentes en el genoma se siguió la estrategia descrita por el Germino (Phakdeekitchaoren B, et al. 2001).

PCR "long-range"

Se diseñaron distintos primers para la obtención de diferentes secciones del gen.

Para cada reacción se emplearon como molde 500 ng de DNA genómico. La amplificación se llevó a cabo con la ayuda de un termociclador Hybaid PCR Sprint, con una temperatura de desnaturalización inicial de 95ºC durante 3 minutos, y un periodo de "hot start" de 80ºC durante 1 minuto, a lo que seguieron una serie de ciclos de desnaturalización a 95ºC durante 20 segundos, anillamiento y extensión con temperaturas especificas para cada primer y un periodo final de extensión a 72ºC durante 10 minutos.

El volumen final de la PCR fue de 50µl, con 4 U (unidades) de rTh DNA polimerasa, XL reaction mix (Applied Biosystems) y una concentración final de Mg de 0.8mM.

Durante la realización de este trabajo se secuenciaron las regiones correspondientes a los exones 11 y 12 por lo que los primers utilizados para la PCR-LR fueron:

BPF12: 5'-CCGCCCCCAGGAGCCTAGACG -3'

BPR5B: 5'-GTAGGACAAGTAGGCGAGGTGCCAAT -3'

PCR "nested"

A partir del producto de amplificación de la "Long Range PCR" se realizaron una serie de diluciones seriadas hasta una concentración final de 1:10 con el fin de minimizar el efecto de contaminación genómica.

Estas diluciones se usaron como molde para realizar PCR Nested, con el propósito de obtener secuencias de menor longitud más fáciles de emplear en el proceso de secuenciado.

Las PCR nested se realizaron siguiendo el protocolo: 4µl de producto diluído de LR-PCR. Para que la reacción tenga lugar se empleó el kit comercial 2XbioMix (BioLine) siguiendo las recomendaciones del fabricante. Las reacciones tuvieron lugar en un termociclador Hybaid PCR Sprint usando el programa: 95ºC durante 3 minutos, 35 ciclos de desnaturalización a 95ºC durante 20 segundos, anillamiento y extensión específicos para cada pareja de primers. Finalmente se realizó un paso de extensión final a 72ºC durante 10 minutos.

Los primers usados fueron:

11F2: 5'-GGGGTCCACGGGCCATG-3'

BPR5B: 5'-GTAGGACAAGTAGGCGAGGTGCCAAT -3'

El producto resultante se purificó finalmente usando un Centricon (Milipore) o mediante el uso de enzimas EXO y SAP (Amersham-Pharmacia-Biotech).

Protocolo EXO-SAP:

Se añade 1µl EXO y 1µl SAP y utilizando un termociclador se calienta a 37ºC durante 30 minutos y a continuación a 80ºC 15 minutos. El producto final se recoge y se conserva a –20ºC.

Secuenciación y búsqueda de variantes

La secuenciación se lleva a cabo usando los primers específicos diseñados para cada exón (o para parte de los exones en aquellos superiores a una longitud de 500 bp.).

11F2 (Primera mitad del exón 11): 5'-GGGGTCCACGGGCCATG-3'

11F1 (Segunda mitad del exón 11): 5'-TGCCCCTGGGAGACCAACGATAC-3'

12F (exón 12): 5'-GAGGCGACAGGCTAAGGG-3'

Las secuencias marcadas se obtuvieron mediante la realización de reacciones de PCR usando BigDye Mix (Applied BioSystems). El programa empleado fue: desnaturalización a 95ºC durante 10 minutos, 35 ciclos de desnaturalización a 96ºC durante 15 segundos, anillamiento a 50ºC durante 10 segundos y una elongación a 60ºC durante 3 minutos. Se llevó a cabo un paso final de elongación a 60ºC durante 10 minutos.

Las secuencias obtenidas se purificaron con etanol siguiendo el protocolo: 31.25µl etanol absoluto, 3µl de NaOH y 7.25µl de H2O y 5µl de muestra. Se

deja reposar la muestra durante 15 minutos y se centrifuga a 14000rpm durante 20 minutos. A continuación se retira el sobrenadante y se añade etanol 70% (50-100µl) y se centrifuga a 14000 rpm durante 6 minutos. Finalmente se retira el sobrenadante y se seca a 37ºC durante 30 minutos.

Para el proceso de secuenciación se empleó un secuenciador ABI 3100-Avant (Applied BioSystems), el cual se cargo con las muestras purificadas en el paso anterior, las cuales se resuspendieron, en el momento de la carga, en 10µl de formamida.

Análisis de las secuencias realizadas

Corrección de las secuencias

Un primer análisis de las secuencias obtenidas se llevó a cabo mediante el uso del software ABI Prism Sequencing Analyser v3.7 (Applied BioSystems). Las secuencias se procesaron con dicho programa y se guardaron en formato FASTA.

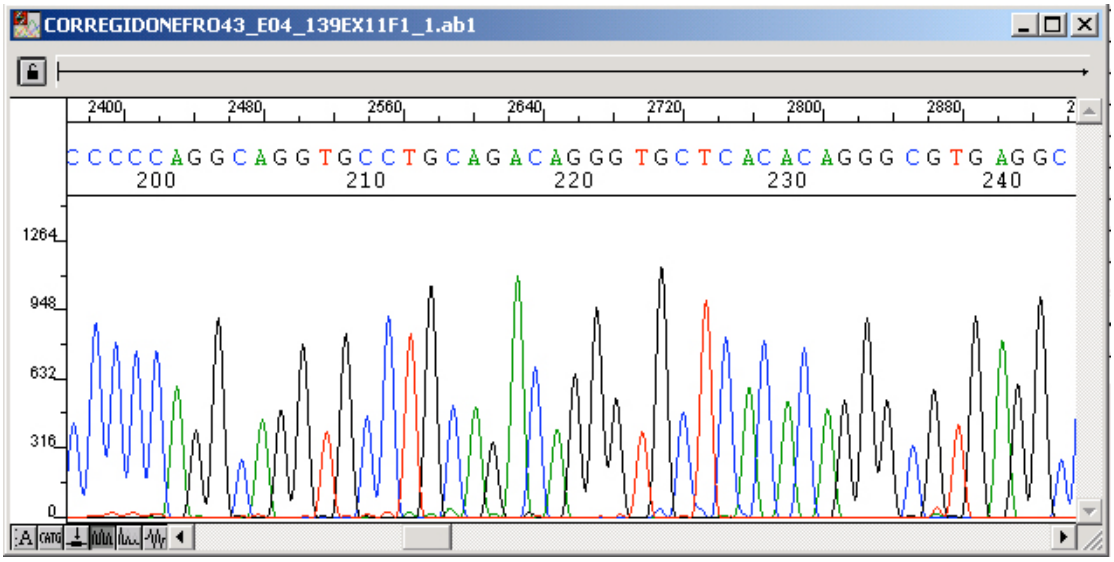


Figure 11: Electroferograma generado por el software Sequencing Analyser.

Las secuencias generadas por el software anteriormente citado presentaron la problemática de introducir una N en aquellas posiciones en las que se detectó una ambigüedad, esto representaba una desventaja en nuestro estudio ya que aunque nos permitía discernir en que posiciones existía un heterocigoto no podíamos saber que ácido nucleico había sido el que se había introducido en la secuencia.

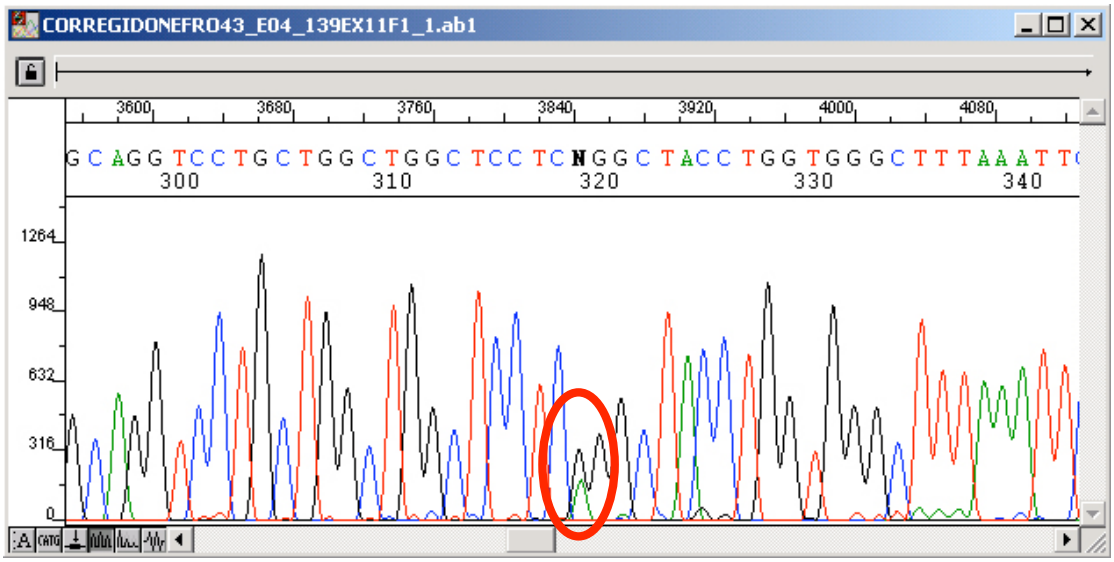


Figure 12: Posición en la que se ha añadido una N en lugar de R (A o G).

Con el fin de solucionar este problema se creó una aplicación en java (Apéndice CorrectSeq.java.) la cual facilita la corrección de aquellas posiciones donde aparece una N por el correspondiente símbolo según código IUPAC (Apéndice Código IUPAC).

El funcionamiento de este programa consiste en la apertura del archivo FASTA que contiene la secuencia a corregir. La secuencia se carga en una "lista indexada" con lo que se crea un puntero a cada posición de la secuencia, por lo que para hacer una corrección solo hay que introducir la posición a cambiar y el nuevo símbolo a añadir. El programa también se ha utilizado para corregir secuencias en las que el programa de análisis (ABI sequence analiser) ha cometido errores de lectura de los electroferogramas.

Las secuencias corregidas con este programa vuelven a ser guardadas en formato FASTA una vez acabadas las correcciones.

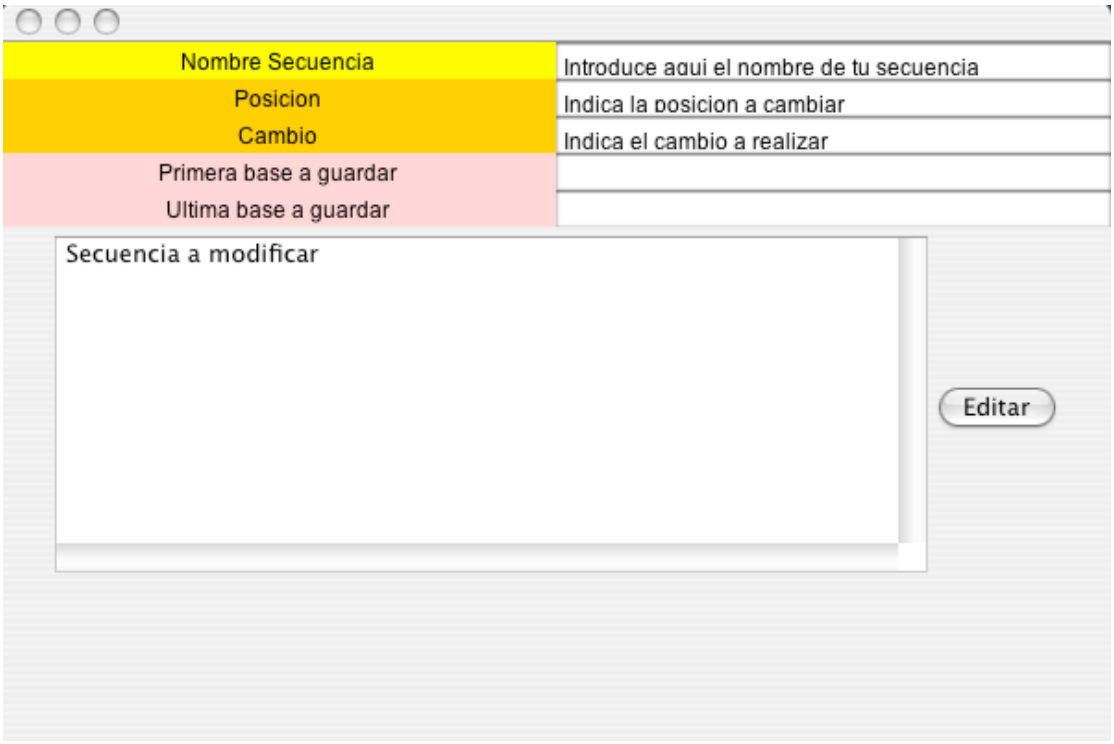


Figure 13: Interfaz gráfico de usuario del programa CorrectSeq.java corriendo en un ordenador con sistema operativo Mac OS X.

Automatización de la búsqueda y anotación de cambios en la secuencia

Comparación de secuencia:

Para la comparación de las secuencias obtenidas con la secuencia de DNA Wild-type descrita en GenBank (HSU24498) y con la secuencia mRNA Wild-type (HSU24497), se utilizó el algoritmo BLAST, para lo cual se instaló la suite BLAST (ftp://ftp.ncbi.nih.gov/BLAST/) en modo local. En la realización de los alineamientos se utilizó el programa bl2seq que es la aplicación del paquete BLAST para alinear 2 secuencias. (Altschul,S.F.).(FiguraX).

```
Query= 1045EX18F1.seq
         (352 letters)

>PKD1
          Length = 53522

 Score =  690 bits (348), Expect = 0.0
 Identities = 351/352 (99%)
 Strand = Plus / Plus


Query: 1     aaaatgcttagtgaggaggctgtgggggtccagtcaagtgggctctccagctgcagggct 60
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 36800 aaaatgcttagtgaggaggctgtgggggtccagtcaagtgggctctccagctgcagggct 36859


Query: 61    gggggtgggagccaggtgaggacccgtgtagagaggagggcgtgtgcaaggagtggggcc 120
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 36860 gggggtgggagccaggtgaggacccgtgtagagaggagggcgtgtgcaaggagtggggcc 36919


Query: 121   aggagcggggctggacactgctggctccacacaggggcccagcagggagctcgtatgccg 180
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 36920 aggagcggggctggacactgctggctccacacaggggcccagcagggagctcgtatgccg 36979


Query: 181   ctcgtgcctgaagcagacgctgcacaagctggaggccatgatgctcatcctgcaggcaga 240
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 36980 ctcgtgcctgaagcagacgctgcacaagctggaggccatgatgctcatcctgcaggcaga 37039


Query: 241   gaccaccgcgggcaccgtgatgcccaccgccatcggagacagcatcctcaacatcacagg 300
              |||||||||||||||||||| ||||||||||||||||||||||||||||||||||||||
Sbjct: 37040 gaccaccgcgggcaccgtgacgcccaccgccatcggagacagcatcctcaacatcacagg 37099


Query: 301   tgccgcggcccgtgccccatgccacccgcccgccccgtgcggccctttcctc 352
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 37100 tgccgcggcccgtgccccatgccacccgcccgccccgtgcggccctttcctc 37151
```

Figura 14: Ejemplo del resultado de un alineamiento utilizando bl2seq.

Para automatizar la realización de los alineamientos con BLAST y su posterior análisis se escribió un programa (Apéndice 1. AnotateDifs.pl), el cual se desarrollo empleando el lenguaje de programación PERL (www.perl.org) y hace uso de los módulos BioPERL (www.bioperl.org). La elección de PERL como lenguaje de programación, en detrimento de otros lenguajes a priori más potentes, como JAVA o C++, se debe en parte a la facilidad que presenta PERL para trabajar con textos y por su habilidad para manejar “expresiones regulares”. Además, la existencia de los numerosos módulos ya escritos, accesibles a través de la distribución gratuita BioPERL, facilitan la generación de un código más depurado y rápido de ejecución.

Para el análisis de los alineamientos se establecieron en el código del programa una serie de filtros que nos permiten analizar sólo aquellos alineamientos significativos y eliminar del análisis alineamientos no específicos que puedan ralentizar la ejecución del programa e inducir a la introducción de información errónea en la base de datos. Estos filtros

consisten en seleccionar solo aquellos alineamientos con una longitud mínima de 100 pb y que presenten una significancia superior al 90%. Aunque el valor de significancia puede parecer un poco bajo, sobre todo tratándose de secuencias para el mismo gen obtenidas a partir de individuos de la misma especie, se decidió utilizar estos valores al encontrarse individuos que presentaban delecciones de hasta 14 pb, las cuales no eran detectadas utilizando unos valores de significancia más restrictivos (se comprobó, además, que este valor no influía negativamente, incluyendo secuencias de baja calidad).

Si un alineamiento es seleccionado para análisis el programa busca los cambios presentes en la secuencia y los anota en una tabla. Junto con el cambio encontrado el programa añade información extra de utilidad para posteriores estudios, como son, el individuo en el que se ha detectado, la base que se ha sustituido y el tipo de cambio ocurrido (transición, transversión, inserción o deleción), también se anota el exón en el que se produjo el cambio, el dominio proteico al que corresponde, la posición en el DNA genómico y el mRNA y calcula la posición que se ve afectada en la secuencia de amino ácidos. Toda esta información se recoge en un archivo de texto que es utilizado posteriormente para introducir la información en la base de datos (como se describirá más adelante).

Para el análisis de la secuencia de amino ácidos se creó otro programa en PERL (protTrans.pl). Este programa traduce las secuencias nucleotídicas, obtenidas por secuenciación directa a secuencia de amino ácidos. Al no conocer en que posición del codón comenzaban nuestras secuencias era necesario realizar la traducción en los seis posibles “frames”. Para realizar la traducción protTrans.pl utiliza un programa externo perteneciente al paquete de software gratuito EMBOSS (transeq). Las secuencias de amino ácidos obtenidas se comparan con la descrita en GenBank para la proteína poliquistina 1 (P98161) mediante el programa supermatcher del paquete de aplicaciones para el análisis biológico EMBOSS (Figura 15).

```
########################################
# Program:  supermatcher
# Rundate:  Mon Feb 16 22:04:08 2004
# Align_format: simple
# Report_file: l236EXl8Fl.seq.supermatch
########################################
#=======================================
#
# Aligned_sequences: 2
# 1: PKDl
# 2: l236EXl8Fl.seq_3
# Matrix: EBLOSUM62
# Gap_penalty: l0.0
# Extend_penalty: 0.5
#
# Length: 49
# Identity:      48/49 (98.0%)
# Similarity:    48/49 (98.0%)
# Gaps:           0/49 ( 0.0%)
# Score: 238.0
#
#
#=======================================

PKDl            2673 gpsrelvcrsclkqtlhkleammlilqaettagtvtptaigdsilnitg   2721
                     |.||||||||||||||||||||||||||||||||||||||||||||||||
l236EXl8Fl.se     85 GSSRELVCRSCLKQTLHKLEAMMLILQAETTAGTVTPTAIGDSILNITG    133

//
```

Figura 15: Resultado de un alineamiento de dos secuencias de amino-ácidos realizada con el programa supermatcher

Al igual que con las secuencias de ácidos nucleicos, se seleccionan, de manera automática, aquellos alineamientos que cumplen unos criterios de calidad determinados y se analizan con el fin de anotar los cambios encontrados. Debido a que un solo cambio en la secuencia de nucleótidos puede alterar toda la estructura de la proteína se establecieron unos filtros muy poco restrictivos con el fin de poder seleccionar y analizar los alineamientos. Se anotan no solo los cambios sino que se indican además, el tipo de cambio (conservativo, non-sense, etc) y las propiedades de los amino-ácidos afectados, además se anota la estructura secundaria que se ve afectada (hélice, lámina, etc.), la cual fue calculada previamente mediante la combinación de diferentes algoritmos específicos para la determinación de estructuras secundarias.

Toda la información obtenida se guarda en una lista que se utilizará una vez generada la base de datos.

Desarrollo de una base de datos de mutaciones

Creación de la base de datos de mutaciones

Con la información generada a partir de los programas anteriormente citados se creó una base de datos relacional, para lo cual se seleccionó MySQL (www.mysql.com) debido a su versatilidad, potencia y bajo coste de mantenimiento. Para que la generación de las tablas que forman la base de datos y la carga de la información que contiene se realice de manera automática, se escribieron dos scripts usando lenguaje de programación Unix Shell Script y SQL. Uno de los scripts genera y carga las tablas con la información génica y fenotípica obtenida a partir de estudios clínicos (Create.PKDGnosis.DB) mientras que el otro se emplea para crear los índices necesarios para acelerar la ejecución de la base de datos (CreateIndex). La ejecución de ambos scripts permitió la integración de la información genotípica con la fenotípica.

Finalmente, se escribieron diferentes algoritmos en SQL para poder extraer información de interés a partir de la base de datos, como fue el estudio de los lugares donde ocurre un mayor número de cambios (lo que nos ayudará a un mejor entendimiento de la proteína y su función).

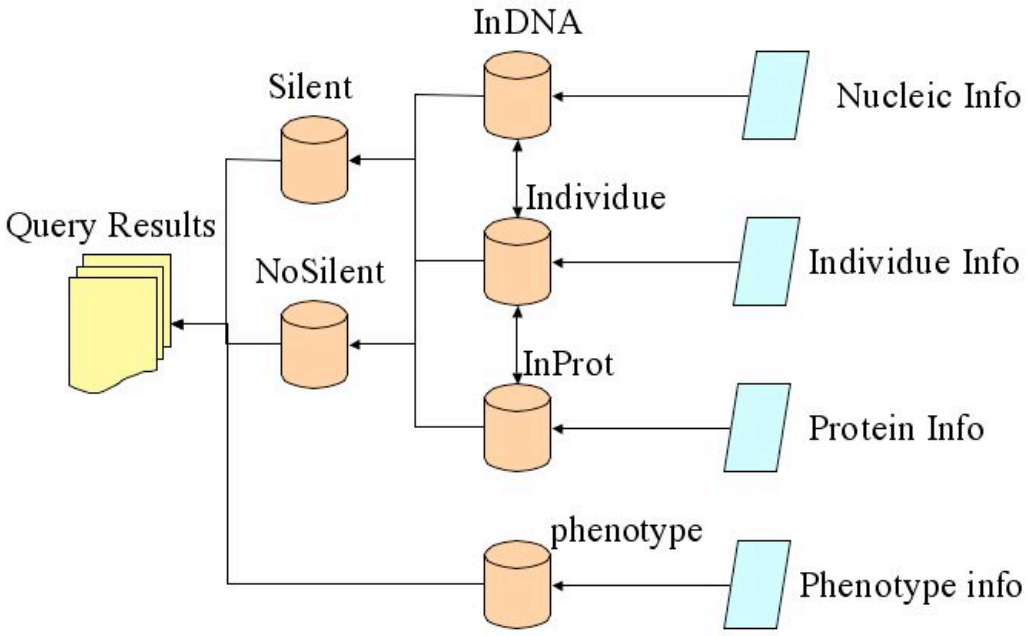


Figure 16: Diagrama del flujo de información en PKDGnosys database.

Desarrollo de un interfaz gráfico de usuario para la presentación de la información contenida en la base de datos.

Con el fin de facilitar a los investigadores el acceso a la información contenida en la base de datos, se ha desarrollado una interfaz gráfica de usuario. Esta interfaz ha sido escrita en formato HTML (www.w3.org) y para otorgarle un comportamiento dinámico a las páginas web generadas se ha utilizado el lenguaje de programación php (www.php.net) junto con javascript (http://devedge.netscape.com/central/javascript/). La razón de utilizar HTML se debe a la posibilidad de crear un formulario de acceso que sea familiar para la mayoría de usuarios, así como muy fácil de emplear. El uso de php se justifica porque permite la creación de sitios web dinámicos, lo que implica que las páginas son creadas de manera instantánea según los parámetros recibidos del formulario, y porque es imprescindible para conectar la base de datos con el formulario de búsqueda, además nos permite la generación de gráficos de manera automática con la información requerida por el investigador.

Al estar desarrollada pensando en internet y encontrándose montada sobre un servidor web Apache 2.0 corriendo sobre plataforma Linux (Mandrake 9.1) nos permitirá en un futuro hacer la base de datos accesible de manera pública a través de internet (http://www.freelancebio.com/PKDGnosys/).

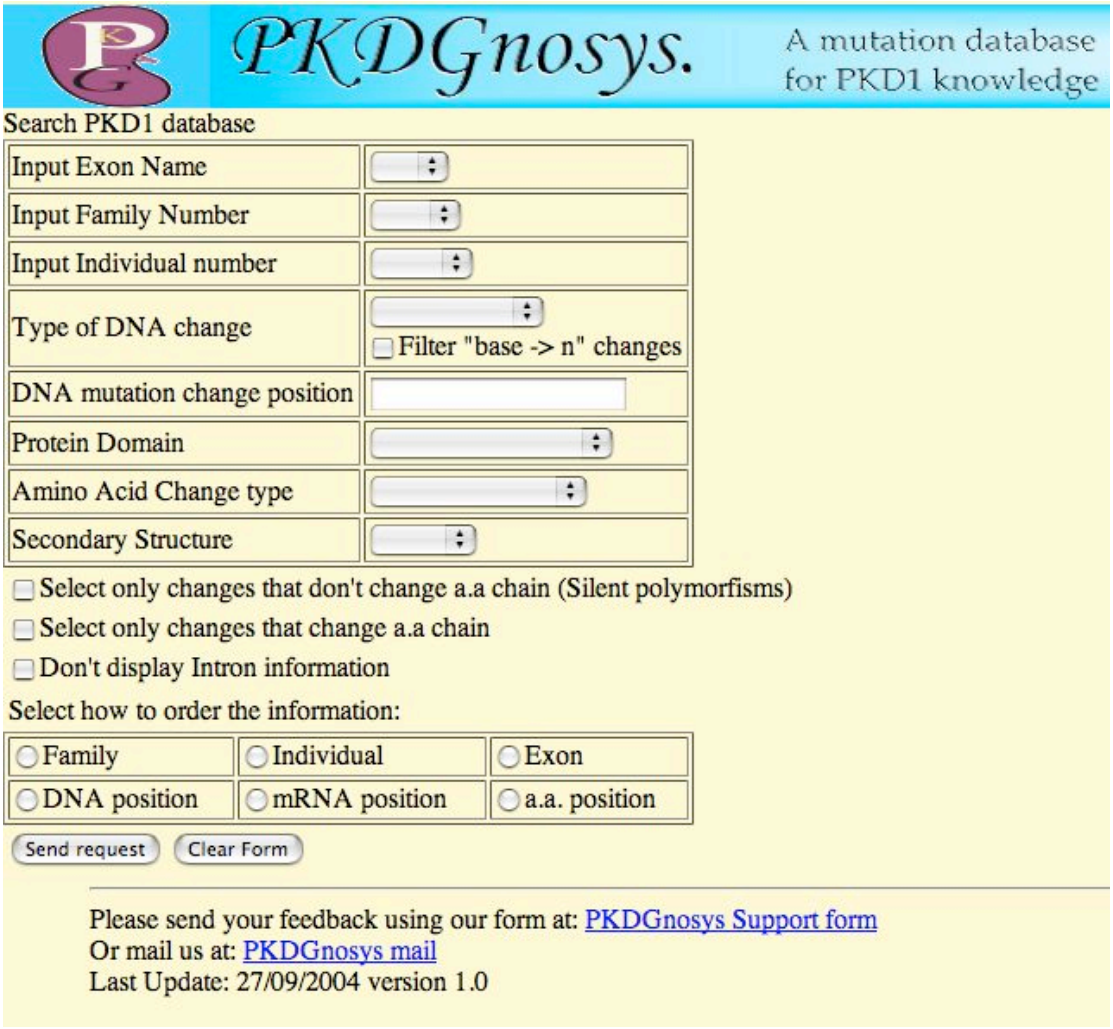
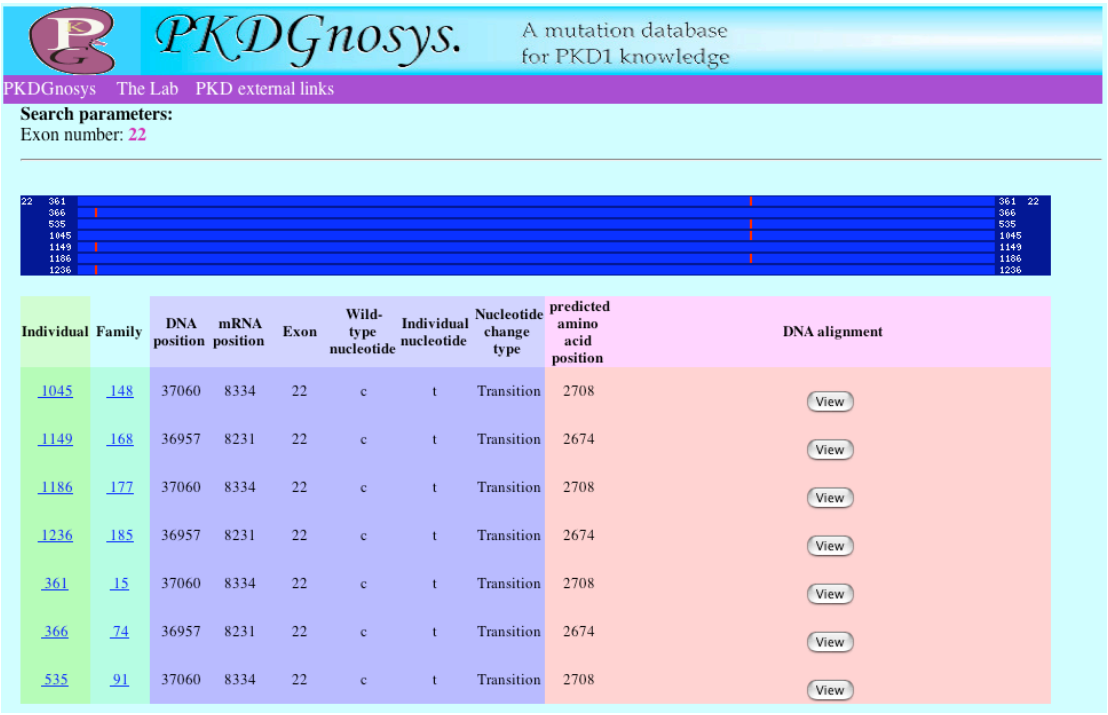


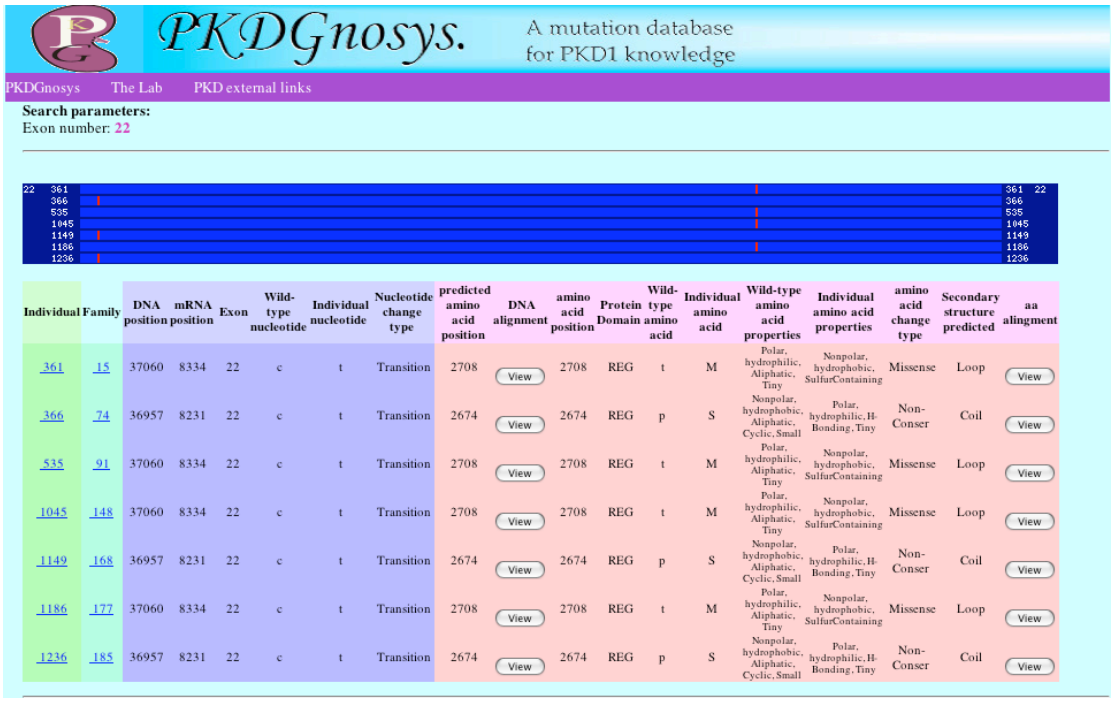
Figure 17: Formulario de consulta a la base de datos.

Figure 18: Gráfica generada de manera dinámica con PHP que muestra en rojo las posiciones correspondientes a cambios detectados en la secuencia.



Individual	Family	DNA position	mRNA position	Exon	Wild-type nucleotide	Individual nucleotide	Nucleotide change type	predicted amino acid position	DNA alignment
1045	148	37060	8334	22	c	t	Transition	2708	View
1149	168	36957	8231	22	c	t	Transition	2674	View
1186	177	37060	8334	22	c	t	Transition	2708	View
1236	185	36957	8231	22	c	t	Transition	2674	View
361	15	37060	8334	22	c	t	Transition	2708	View
366	74	36957	8231	22	c	t	Transition	2674	View
535	91	37060	8334	22	c	t	Transition	2708	View

Figure 19: Tabla que muestra la información correspondiente a los cambios detectados. Se muestra solo la información correspondiente a los ácidos nucleicos.



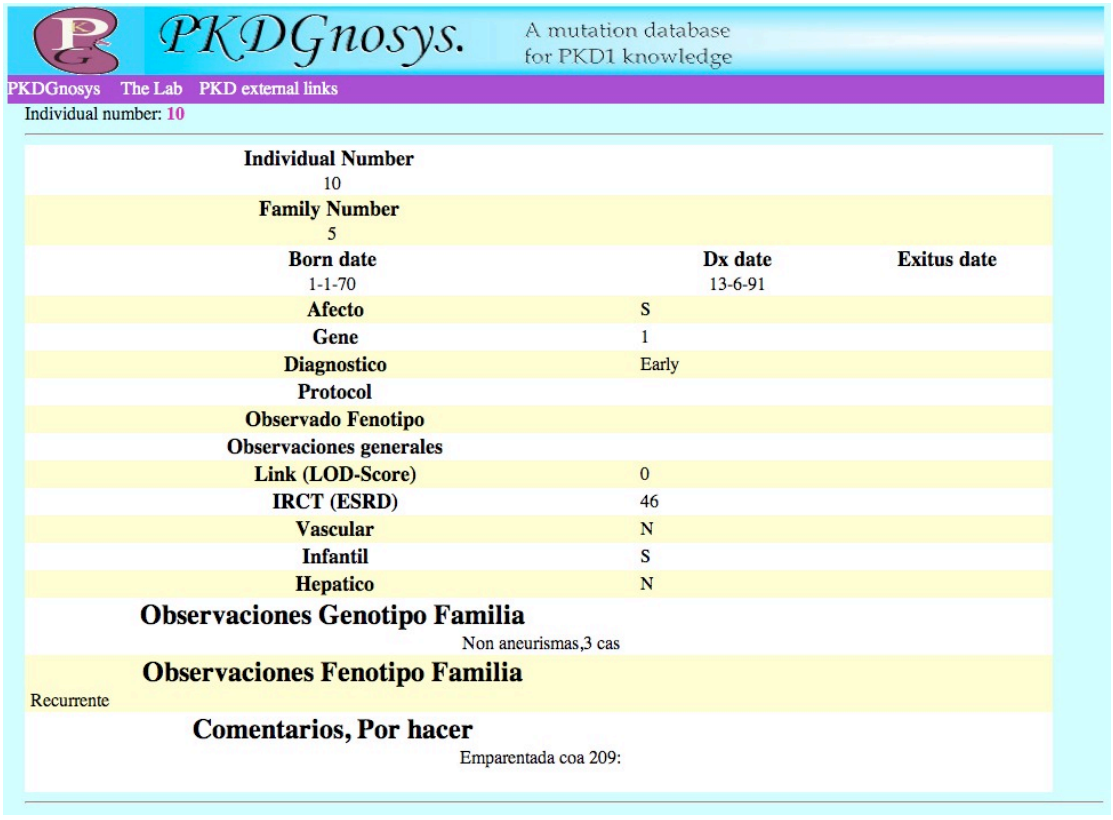
PKDGnosys. A mutation database for PKD1 knowledge

PKDGnosys The Lab PKD external links

Search parameters:
Exon number: 22

Individual	Family	DNA position	mRNA position	Exon	Wild-type nucleotide	Individual nucleotide	Nucleotide change type	predicted amino acid position	DNA alignment	amino acid position	Protein Domain	Wild-type amino acid	Individual amino acid	Wild-type amino acid properties	Individual amino acid properties	amino acid change type	Secondary structure predicted	aa alingment
361	15	37060	8334	22	c	t	Transition	2708	View	2708	REG	t	M	Polar, hydrophilic, Aliphatic, Tiny	Nonpolar, hydrophobic, SulfurContaining	Missense	Loop	View
366	74	36957	8231	22	c	t	Transition	2674	View	2674	REG	p	S	Nonpolar, hydrophobic, Aliphatic, Cyclic, Small	Polar, hydrophilic, H-Bonding, Tiny	Non-Conser	Coil	View
535	91	37060	8334	22	c	t	Transition	2708	View	2708	REG	t	M	Polar, hydrophilic, Aliphatic, Tiny	Nonpolar, hydrophobic, SulfurContaining	Missense	Loop	View
1045	148	37060	8334	22	c	t	Transition	2708	View	2708	REG	t	M	Polar, hydrophilic, Aliphatic, Tiny	Nonpolar, hydrophobic, SulfurContaining	Missense	Loop	View
1149	168	36957	8231	22	c	t	Transition	2674	View	2674	REG	p	S	Nonpolar, hydrophobic, Aliphatic, Cyclic, Small	Polar, hydrophilic, H-Bonding, Tiny	Non-Conser	Coil	View
1186	177	37060	8334	22	c	t	Transition	2708	View	2708	REG	t	M	Polar, hydrophilic, Aliphatic, Tiny	Nonpolar, hydrophobic, SulfurContaining	Missense	Loop	View
1236	185	36957	8231	22	c	t	Transition	2674	View	2674	REG	p	S	Nonpolar, hydrophobic, Aliphatic, Cyclic, Small	Polar, hydrophilic, H-Bonding, Tiny	Non-Conser	Coil	View

Figure 20: Tabla en la que se ha seleccionado que se muestre la información correspondiente a los amino ácidos.



PKDGnosys. A mutation database for PKD1 knowledge

PKDGnosys The Lab PKD external links

Individual number: 10

Individual Number		
10		
Family Number		
5		
Born date	**Dx date**	**Exitus date**
1-1-70	13-6-91	
Afecto	S	
Gene	1	
Diagnostico	Early	
Protocol		
Observado Fenotipo		
Observaciones generales		
Link (LOD-Score)	0	
IRCT (ESRD)	46	
Vascular	N	
Infantil	S	
Hepatico	N	

Observaciones Genotipo Familia

Non aneurismas,3 cas

Observaciones Fenotipo Familia

Recurrente

Comentarios, Por hacer

Emparentada coa 209:

Figure 21: Tabla que muestra la información fenotípica de un individuo contenida en la base de datos.

PROTEIN SEQUENCE ANALYSIS

Secondary Structure

Secondary structure determination.

For the determination of the secondary structure of the a.a. sequence of Polycystin-1 a number of web servers were used (NNPredict, PROF, SSPRO, PHDsec). A multiple alignment was performed with the secondary structure sequences obtained in order to get a consensus sequence. To get this consensus sequence the “Consensus” web application was used. From the different consensus sequences obtained the consensus-50% was selected, as this is the one that introduces a smaller number of GAPs in the sequence.

Tertiary Structure

Due to the length and complexity of polycystin-1 most automatic methods for protein modeling were not able to work with the polypeptide chain in a native state.

To solve this issue the amino acid chain was splitted based on two different criteria.

First, the sequence was splitted by its protein domains and domains were modeled separately. This study was made using Rosetta server.

As we were modeling each domain independently of the surrounding domains we found out that this was not the best method to get an idea on how the protein folds.

The second method used consists in splitting the polypeptide chain in 6 sets of around 800 a.a.. With this method we were able to model not only the structure of each domain but also how the different domains interact. As we will show later on, the interaction among different domains showed to play an important role in the final conformation of each single domain (we found that the structure of some domains was completely different if we model them alone or if we model them within a group of domains).

For the study of the tertiary structure of polycystin-1 following the second methodology we used Robetta modeling server.

The reason to use two different servers for both methods was because the limitation in a.a. chain size in Rosetta (<150 a.a. for 3-D simulations) didn't allow us to use the second method with this server.

We also have to note that Robetta is based on Rosetta but as it models longer sequences the running time is much longer so we limited its use to the second method only.

Tertiary structure determination.

For the study of the 36 domains of the protein we used Rosetta server. (REJ domain was not modeled using this method due to the length of its sequence). To use this server we created 36 files with the sequence of each single domain, storing the files in FASTA format.

In order to obtain the three-dimensional coordinates of each domain we selected, in the output options section, to generate the 3D coordinates in PDB format. This option is not selected by default as it slows down the performance of the server.

PDB files of each domain were obtained and stored for further studies.

As in the previous method, the six fragments were stored as independent files in FASTA format. These files were uploaded to Robetta server and the different 3D coordinates obtained for each fragment were stored for validation.

Fragment 1: from a.a. 1 to 680. The domains included are: signal peptide, amino flanking region, LRR, carboxy flanking region, WSC domain, PKD domain 1, C-type lectin domain, LDL-A related motif.

Fragment 2: from a.a. 850 to 1550. Domains PKD domain 2 – 9.

Fragment 3: from a.a. 1550 to 2146. Domains PKD domain 10 – 16.

Fragment 4: from a.a. 2146 to 3110. REJ domain.

Fragment 5: from a.a. 3012 to 3580. Domains GPS, PLAT/LH-2 and Transmembrane (TM) 1 – 4.

Fragment 6: from a.a. 3580 to 4301. Domains TM 5 – 11 and Coiled-coil.

For each fragment 5 models were obtained

Tertiary structure validation

For the models obtained three different validation methods were performed.

We use the validation suite Biotech that automatically performs the three methods (Procheck, Prove, What If).

Validation was only performed for one of the five models obtained for each protein section. The selection of each model was based on how good the structures were in a Ramachandran plot.

Tertiary structure comparison

With the tertiary structure coordinates obtained from Robetta we performed a structural alignment against all known protein structures stored at the PDB database.

To perform these alignments we used the program Dali than can be reached at the E.B.I. web server.

From the results obtained from running DALI we got a list of proteins that share similar folds. The most representative of this list of proteins (those ones with better Z-score and RMSD values or with a function more in concordance with the probable function of polycystin-1) were selected and a pairwise alignment of the structures of these proteins was performed. The difference with the new alignments and those ones performed with DALI was that in the case of the new alignments the algorithm searches for flexible regions within the protein improving alignment scores and getting a more accurate idea on how similar both of the aligned structures are. The pairwise alignments were performed with the web server FlexProt (Shatsky M.).

Multiple sequence alignment

For the alignment of multiple amino acid sequences and the alignment of the secondary structure of different amino acid sequences the software tool Clustal (Higgins, D.G. et al 1988; Thompson, J.D. 1997) and the web server Multalin (Corpet, F.) were used.

RESULTS AND DISCUSSION

Exon 11-12 sequenciation:

A set of changes were annotated in the database from the information obtained by the sequenciation of PKD1 gene exons 11 and 12 of each of the affected individuals. Mutations within these exons affect domains PKD domain 2 and 3.

Some of the most relevant changes found, because of their implications, were

- 11 bp insertion in individual 139, exon 11. 21223-21234INS[11].
- 1 bp insertion in individual 1159. 21626-21627INS[t].

A sequence variation was found within exon 11, this variant was the sequence 21239-gccac-21243 being the wild type one 21239-caacg-21243. (Change in the amino acid sequence from 774-AT-775 to 774-QR-775, being both changes non conservative). We describe this sequence variation as not linked to ADPKD as we found this change in healthy and ADPKD affected individuals.

Table 1: Other SNPs observed within exons 11-12 were:

Gene position	Type of DNA change	Wildtype Nucleotide	Observed Nucleotide	Type of Amino acid Change
21135	Transition	G	A	Silent
21207	Transversion	G	C	Silent
21515	Transversion	C	G	Silent
21570	Transversion	C	G	Silent

Database Information (by 30-09-2004)

Number of individuals studied = 210

Number of exons annotated in the database = 28

Exons= 5, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 33, 34, 35, 36, 38, 41, 43, 46.

Nucleotide change statistics:

1. Number of annotated nucleotide changes (changes at different DNA positions)= 384
2. Number of changes that occur within an exon= 228
3. Silent changes= 81
4. Non silent changes = 150
5. Insertions= 11
6. Deletions= 25
7. Transitions= 69
8. Transversion= 52
9. Non Silent InFrame= 119
10. Non Silent FrameShift = 35

Table 2: This table shows the total number of changes described within an exon and how these changes are distributed depending on the nucleotide change type. The MySQL query used was: Select Exon, ChType, Count(Distinct DNApos) from genotype group by Exon, ChType;

Exon	Exon Length	Transition	Transversion	Insertion	Deletion
5	672	2	2	0	1
11	756	23	24	6	3
12	132	1	1	0	0
13	176	5	5	3	3
14	134	3	0	0	0
15	3620	29	18	6	8
17	144	1	0	0	0
18	280	1	1	0	0
19	214	4	0	0	0
20	160	3	0	0	0
21	153	1	0	0	1
22	145	2	0	0	0
23	630	9	7	1	3
24	157	6	1	0	3
25	253	11	4	0	0
26	196	4	0	0	0
28	144	2	1	1	0
29	211	1	0	0	3
30	127	1	0	1	0
33	185	0	0	0	0
34	94	0	0	0	0
35	119	2	1	0	0
36	203	0	2	1	0
40	140	5	10	0	0

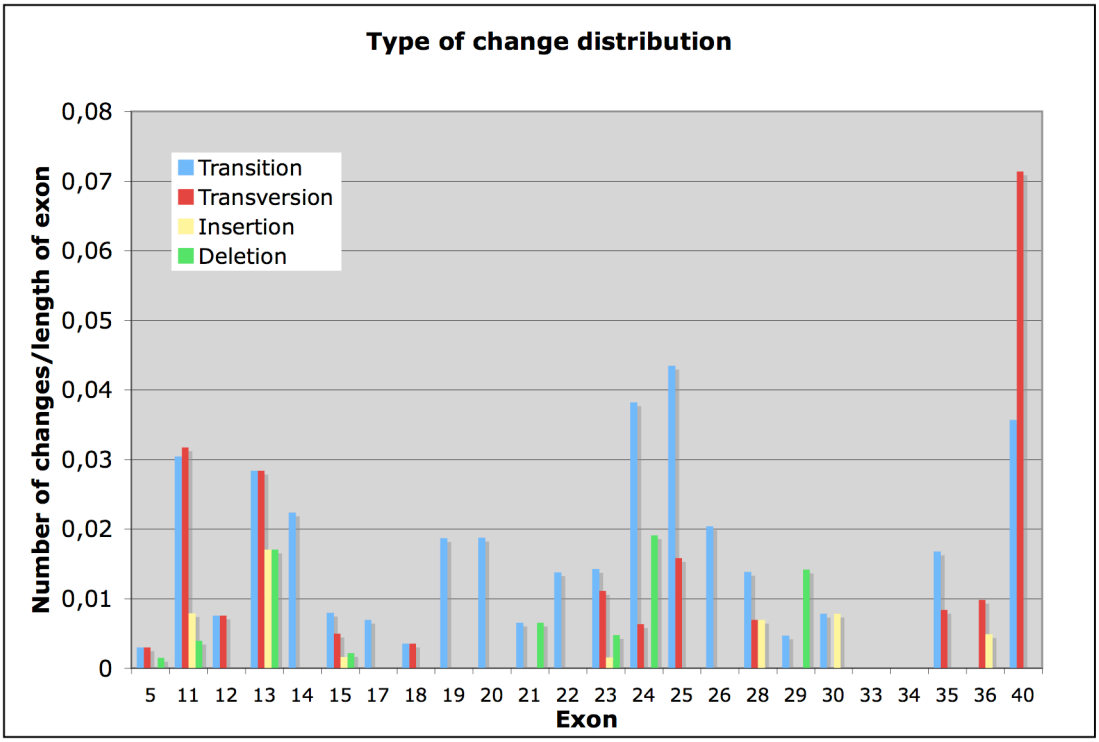


Figure 22: Graph showing the distribution of changes within an exon. As expected in frame changes are more frequent than frameshift changes.

Table 3: Distribution of changes within protein domains. From these results it can be shown which parts of the protein are less change prone and more conserved. It can be seen that the number of changes in the last transmembrane domains is null meaning that this structure might be quite conserved. The same can be said about the PKD domains as we observed that the region corresponding to the PKD domain core presents a low number of changes for every PKD domain. REJ and GPS seem to be the most flexible domains of the protein as these two domains are the ones that present a higher number of non-silent changes.

Exon	Domain	Total number of changes	Silent	No Silent
5	PKD domain 1	1	1	0
5	WSC	3	2	1
11	PKD domain 2	12	5	7
11	PKD domain 2 core	2	1	1
12	PKD domain 3	2	0	2
13	PKD domain 3	12	4	8
14	PKD domain 4	3	2	1
15	PKD domain 4	13	7	6
15	PKD domain 5	6	2	4
15	PKD domain 6	7	3	4
15	PKD domain 6 core	1	1	0
15	PKD domain 7	2	0	2
15	PKD domain 7 core	1	1	0
15	PKD domain 8	2	0	2
15	PKD domain 9	3	1	2
15	PKD domain 9 core	1	0	1
15	PKD domain 10	2	2	0
15	PKD domain 11	6	0	6
15	PKD domain 11 core	2	1	1
15	PKD domain 14	2	2	0
15	PKD domain 15	8	0	8
15	PKD domain 15 core	1	0	1
15	PKD domain 16	2	1	1
17	REJ	1	1	0
18	REJ	2	1	1
19	REJ	4	2	2
20	REJ	3	2	1
21	REJ	2	0	2
22	REJ	2	0	2
23	REJ	9	1	8
25	GPS	6	2	4
26	TM 1	1	1	0
28	PLAT/LH2	4	2	2
30	TM 2	2	2	0
36	TM 4	2	0	2

From the data at the database in was not possible to establish a relationship between the number of changes per individual and the age when the individual enters End Stage Renal Disease (ESRD). This data is in concordance with that, by Devuyst *et al,* where the high variability in the age of ESRD was described.

Table 4: Distribution of ESRD depending on number of amino acid changes per individual.

MySQL Query used: select genotype.Nind, count(genotype.Nind)as NumberOfChanges, phenotype.Dx from genotype, phenotype where genotype.GenaaSym!=genotype.InaaSym and genotype.Exon!=0 and genotype.Nind=phenotype.Nind group by Nind order by NumberOfChanges

Number of changes	Average age when enters ESRD	Range of ages
x>12	51	71-33
8>x≤12	55	84-35
4>x≤8	50	66-48
X≤4	53	76-33

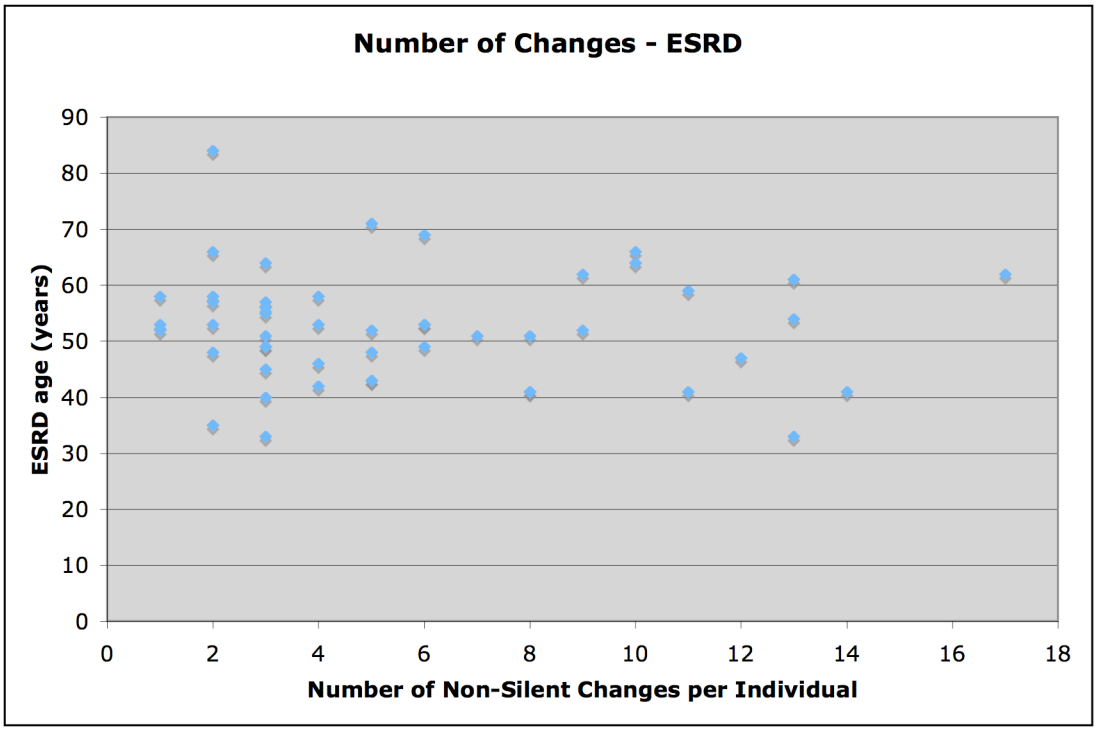


Figure 23: Distribution of the number of changes and its relation with the age of entrance in ESRD.

Type of a.a. changes:

Table 5: It is expected that Frameshift mutations will also lead to Non sense changes but only changes that code for a stop codon are annotated in this table.

Type of a.a. change	Number of changes
Conservative	59
Non-conservative	60
Non sense	9 (stop codon)

Table 6: As expected the highest number of changes occur in coil/loop regions were it's believed that the protein is more flexible. This fact also indicates that the protein has no enzymatic activity as for enzymes loop structures use to be very well conserved.

Type of secondary structure affected	Number of changes
Coil/loop	85
Helix	21
Sheet	44

Table 7: Changes observed in a frequency higher than ¼ of the studied population. These changes were observed in healthy and ADPKD affected individuals so they don't seem to be linked to the development of the disease. Changes from 21238 to 21245 are the result of an error in the alignment performed by BLAST and the sequence variant at this area was presented above (exon 11-12 sequenciation section).

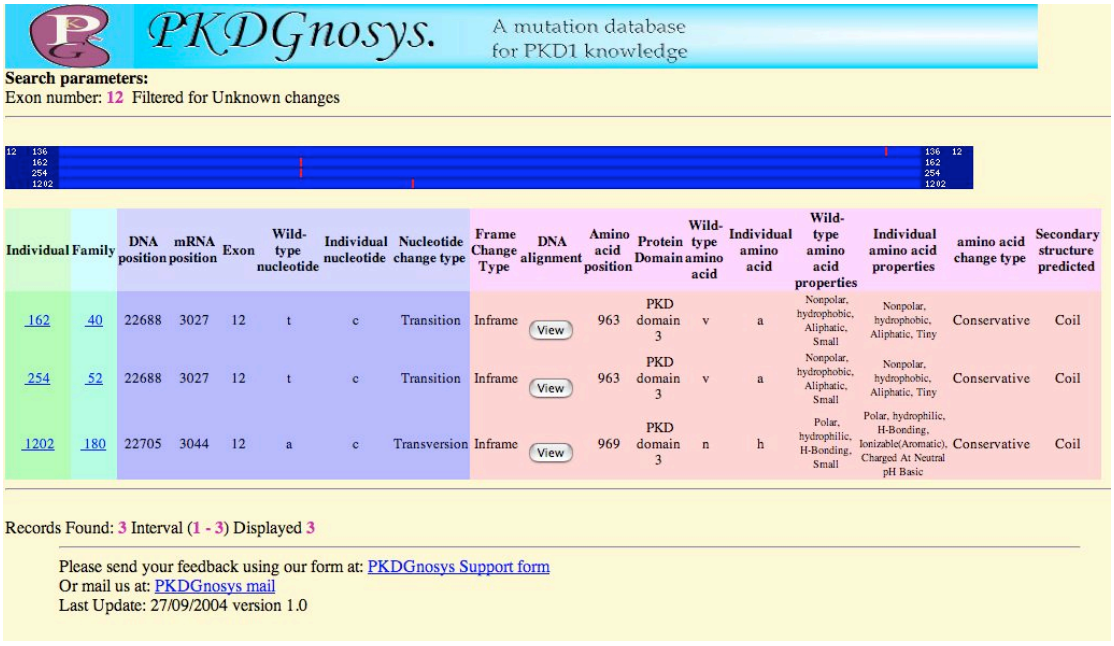
Query: 242 tgcagccacgg-aacagctcaccgtgctgctgggcttgaggcccaaccctggactgcggc 300

Sbjct: 21235 tgca-caacgggaacagctcaccgtgctgctgggcttgaggcccaaccctggactgcggc 21293

DNA position	Number of occurrences	Type of amino acid change
21135	134	Silent
21207	147	Silent
21238	147	Silent
21240	147	Non Silent Non Conservative
21238 21245	133	Non Silent Non Conservative
21515	130	Silent
21570	93	Silent
21885	121	Silent
23536	57	Silent
23633	57	Silent
24607	116	Silent
29210	60	Silent
33871	115	Silent
33872	118	Non Silent Conservative
34130	82	Silent
34205	139	Silent

Web Interface

The web interface developed was designed for an easy access to the data stored in the database.



PKDGnosys. A mutation database for PKD1 knowledge

Search parameters:
Exon number: 12 Filtered for Unknown changes

12 136 162 254 1202

Individual	Family	DNA position	mRNA position	Exon	Wild-type nucleotide	Individual nucleotide	Nucleotide change type	Frame Change Type	DNA alignment	Amino acid position	Protein Domain	Wild-type amino acid	Individual amino acid	Wild-type amino acid properties	Individual amino acid properties	amino acid change type	Secondary structure predicted
162	40	22688	3027	12	t	c	Transition	Inframe	View	963	PKD domain 3	v	a	Nonpolar, hydrophobic, Aliphatic, Small	Nonpolar, hydrophobic, Aliphatic, Tiny	Conservative	Coil
254	52	22688	3027	12	t	c	Transition	Inframe	View	963	PKD domain 3	v	a	Nonpolar, hydrophobic, Aliphatic, Small	Nonpolar, hydrophobic, Aliphatic, Tiny	Conservative	Coil
1202	180	22705	3044	12	a	c	Transversion	Inframe	View	969	PKD domain 3	n	h	Polar, hydrophilic, H-Bonding, Small	Polar, hydrophilic, H-Bonding, Ionizable(Aromatic), Charged At Neutral pH Basic	Conservative	Coil

Records Found: 3 Interval (1 - 3) Displayed 3

Please send your feedback using our form at: PKDGnosys Support form
Or mail us at: PKDGnosys mail
Last Update: 27/09/2004 version 1.0

Figure 24: Representation of the resulting window for exon 12 and non-silent changes

The query results window is divided in two different parts. The first one shows the graphical representation of the changes encountered in the exon queried. The blue line representing the total length of the exon and the red line the position were the change was found. This method helps us getting an idea on at what position of the exon a change has occurred and to get an idea if there is a region within the exon that is more error prone. In the case shown above the number of changes within exon 12 was quite small so no conclusion were formulated.

The second part of the results window shows all the information about the changes found in exon 12. First it displays the individual and family to whom the individual belongs. Next are the nucleotide change information and finally the amino acid change information.

The graphical display allowed for the visualization of SNPs profiles and as shown in the figure bellow it was a very useful tool for the fast interpretation of these SNP profiles.

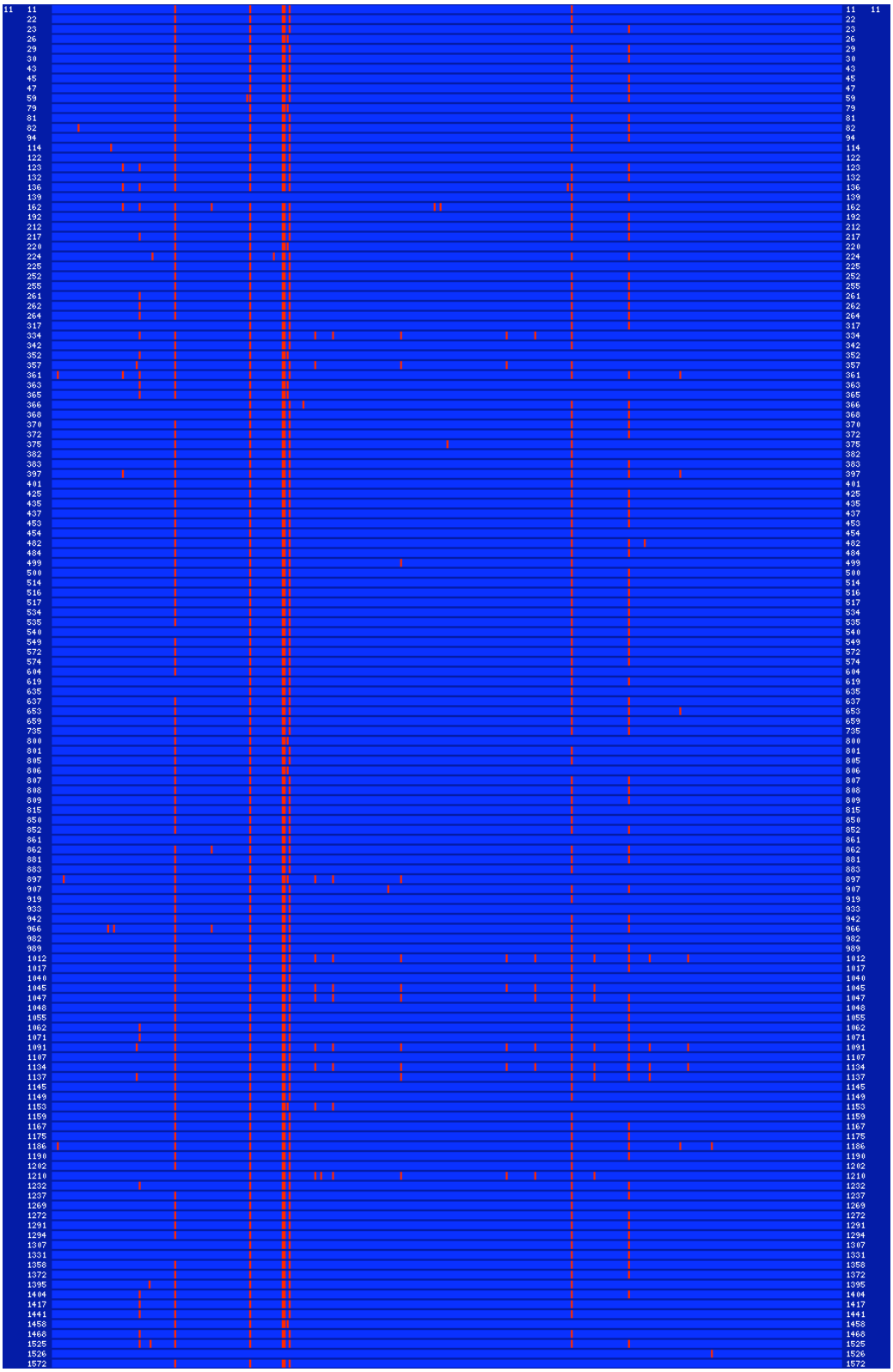
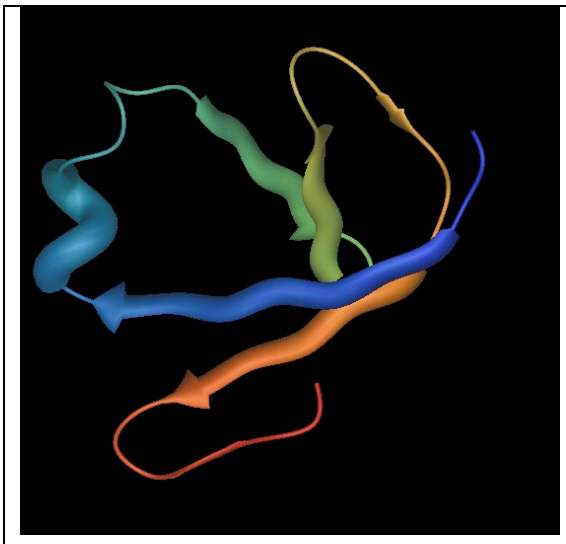


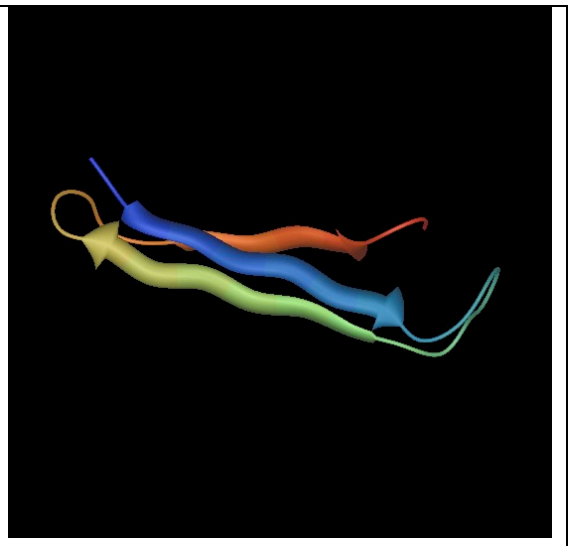
Figure 25: Graphical output from querying the database for changes in exon 11. As it can be seen, there is a group of changes that occur almost in every individual studied. There is also a group of changes that is repeated in a few individuals and always come together. These changes could be intra-familiar or a profile characteristic of the individual from the same region. More studies will be performed on these profiles in the future.

Tertiary structure determination.

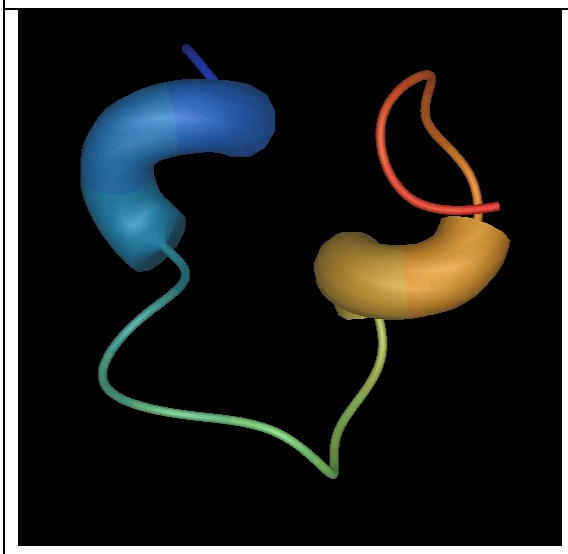
Rosetta. We modeled 36 domains.



LRR-CFlank

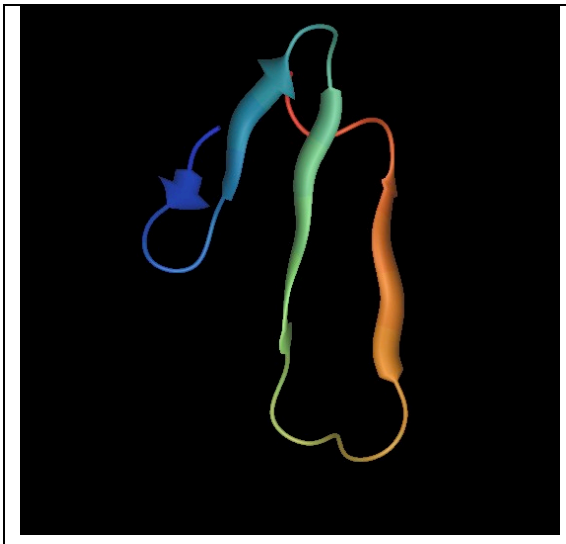


LRR-NFlank

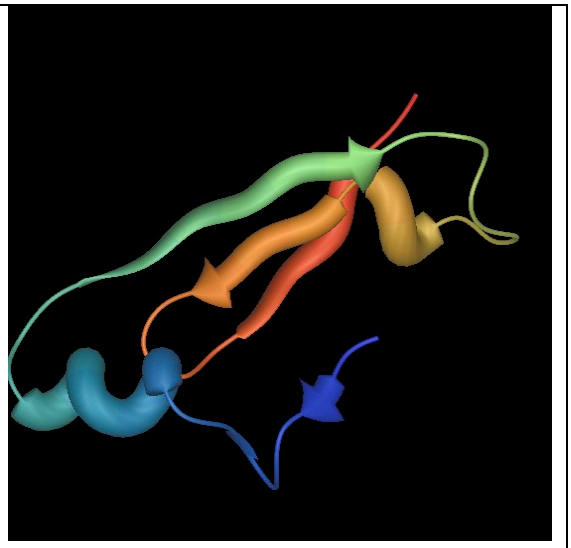


LRR-1

LRR-2



LDL-Recep

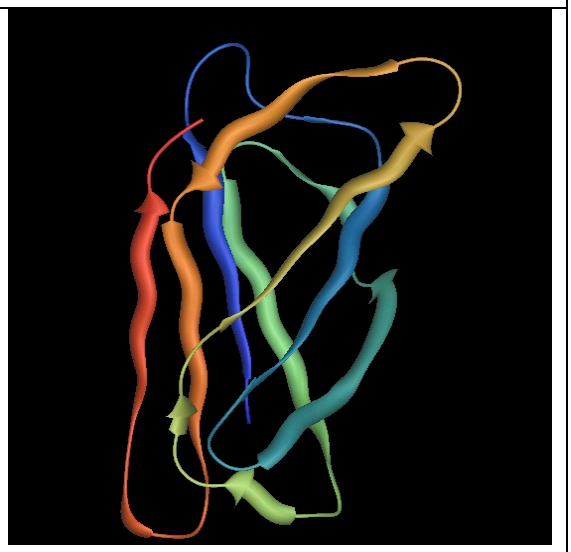


GPS

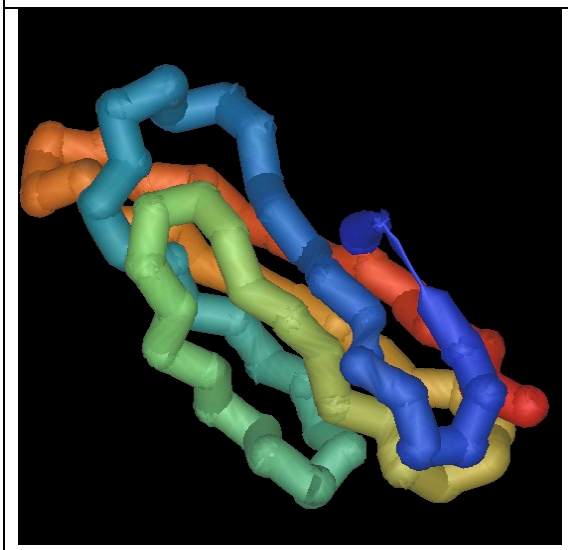
REJ

Not available due to the length of the sequence made it impossible to get a model by Rosetta.

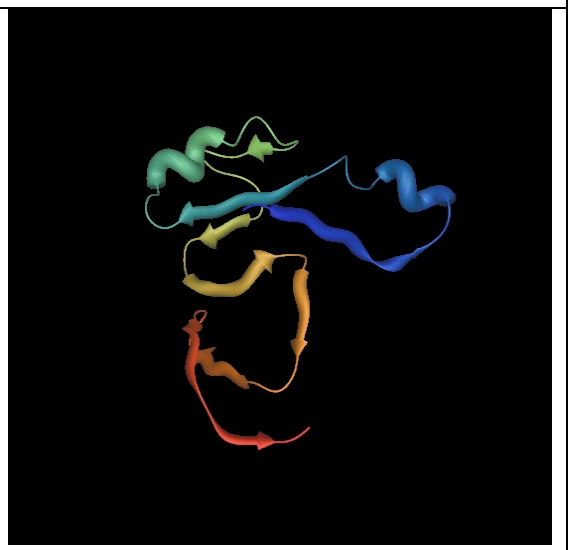
REJ domain was modeled using Robetta.



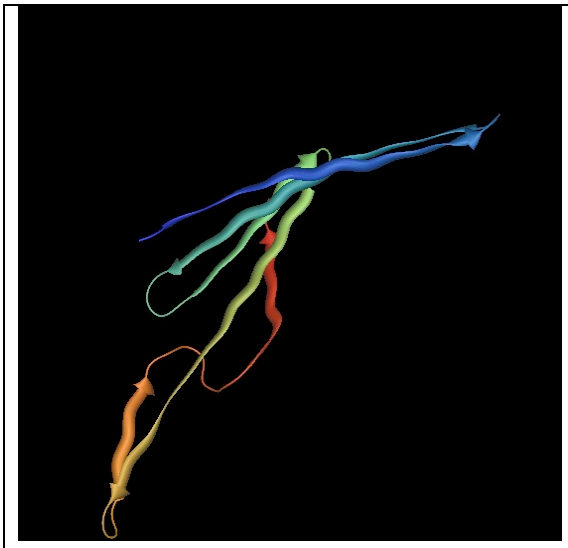
PLAT



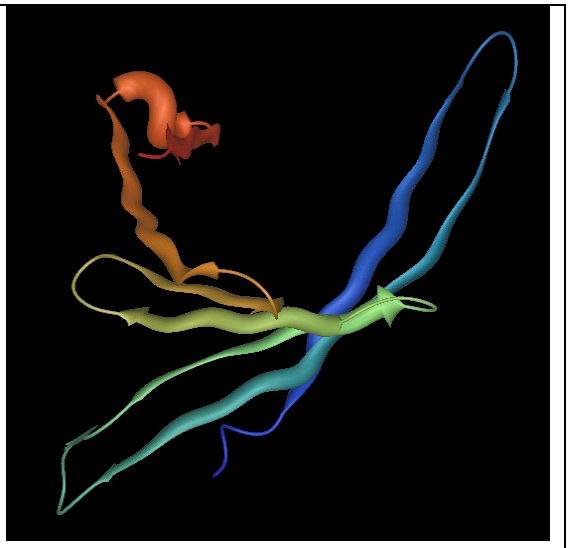
PKD-domain1



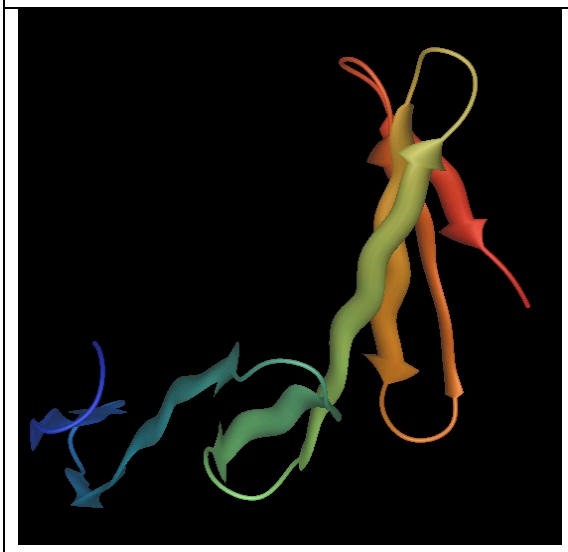
PKD-domain2



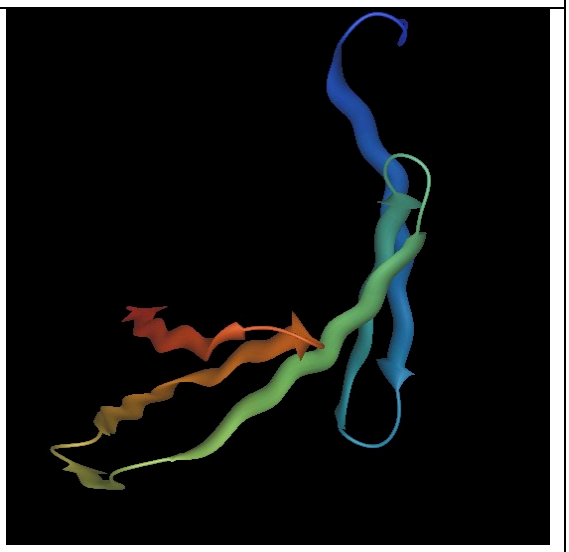
PKD-domain3



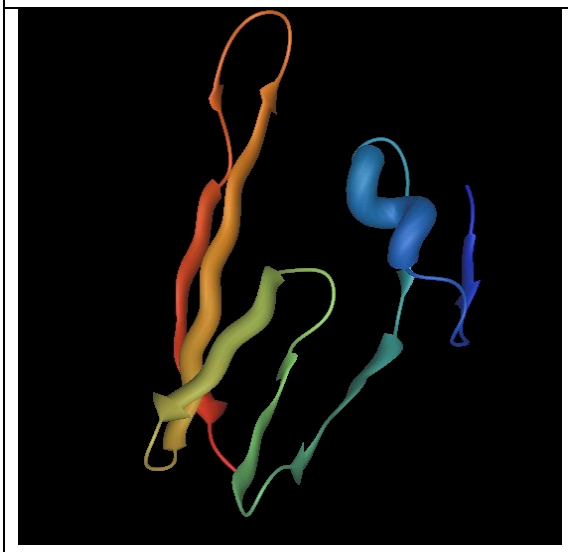
PKD-domain4



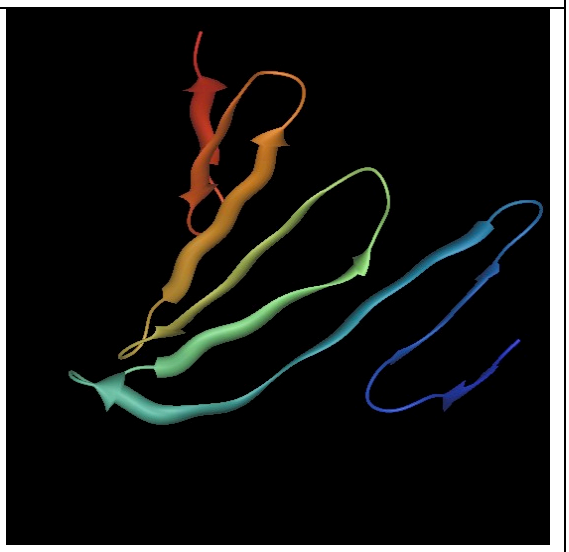
PKD-domain5



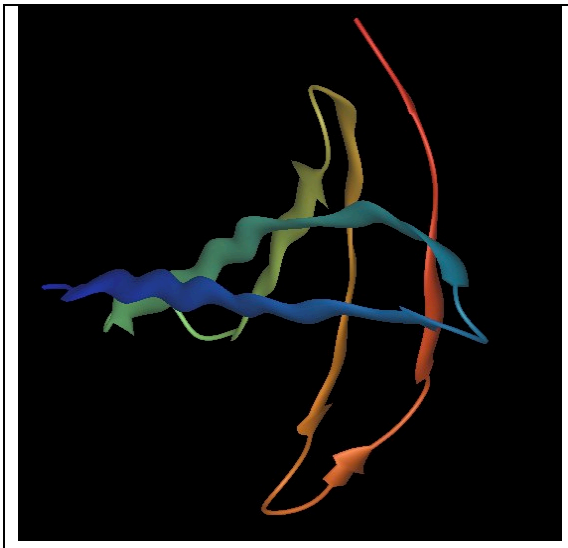
PKD-domain6



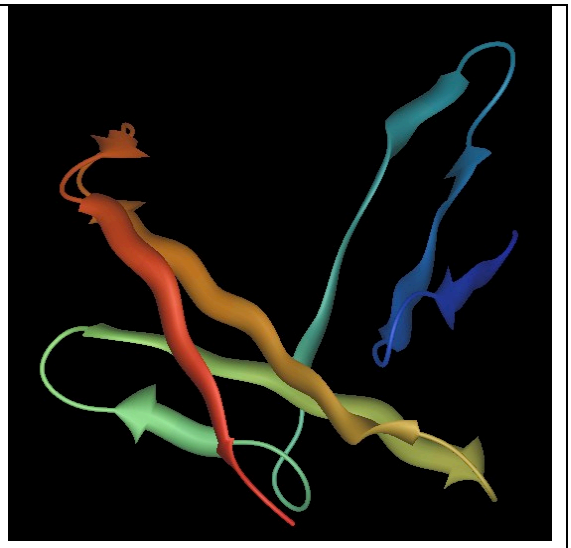
PKD-domain7



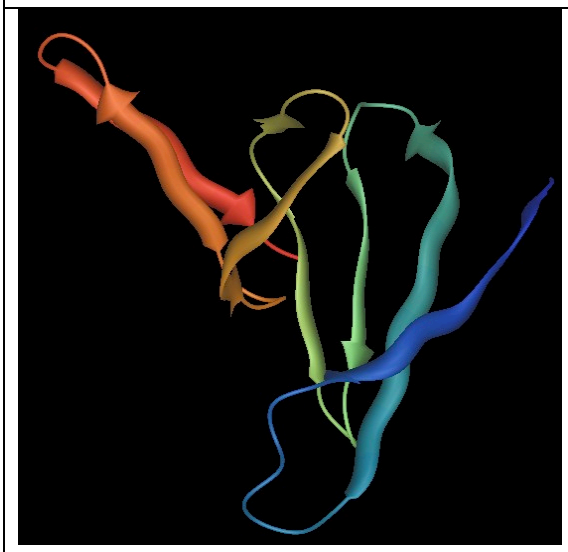
PKD-domain8



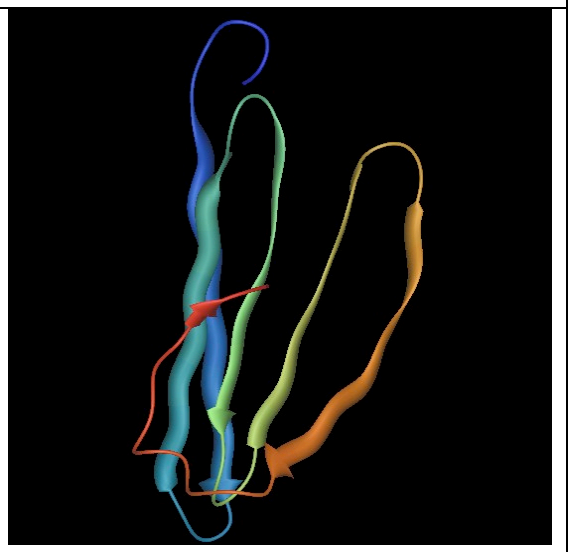
PKD-domain9



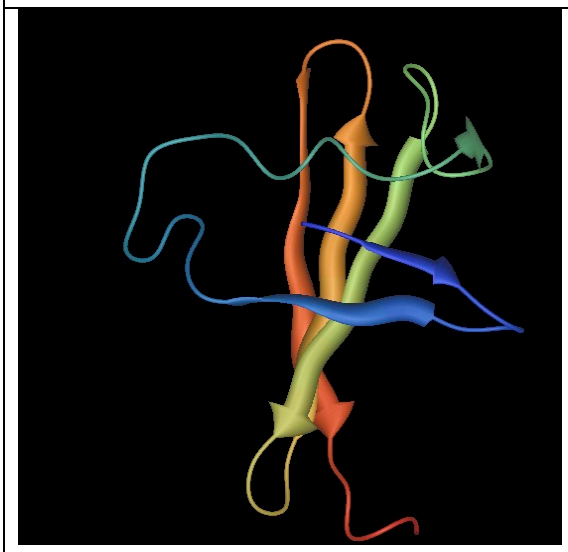
PKD-domain10



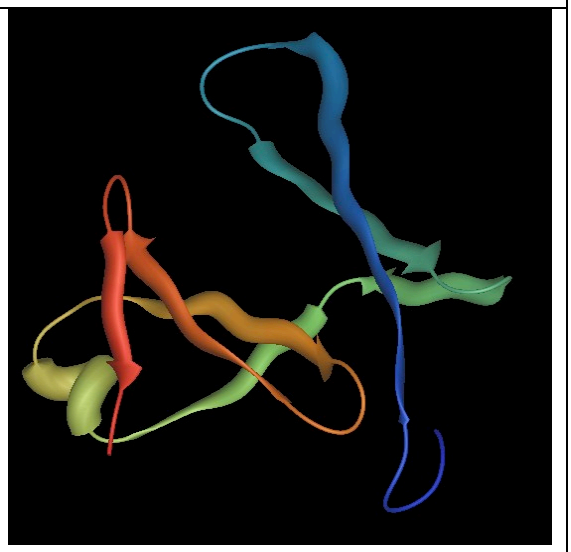
PKD-domain11



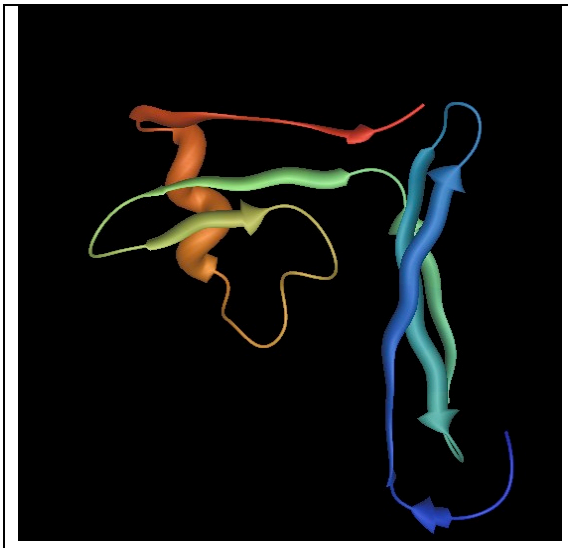
PKD-domain12



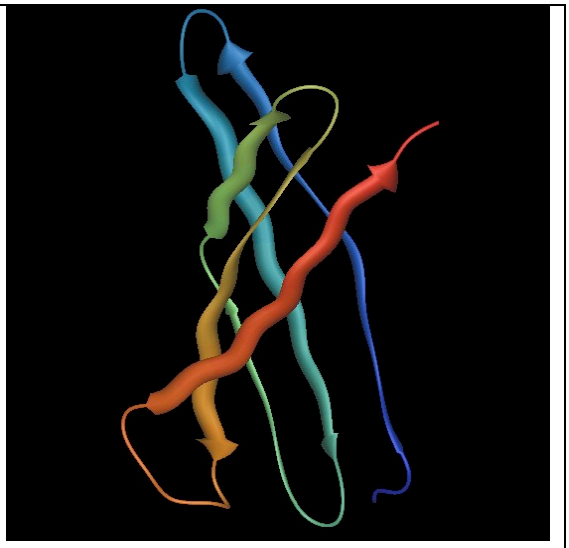
PKD-domain13



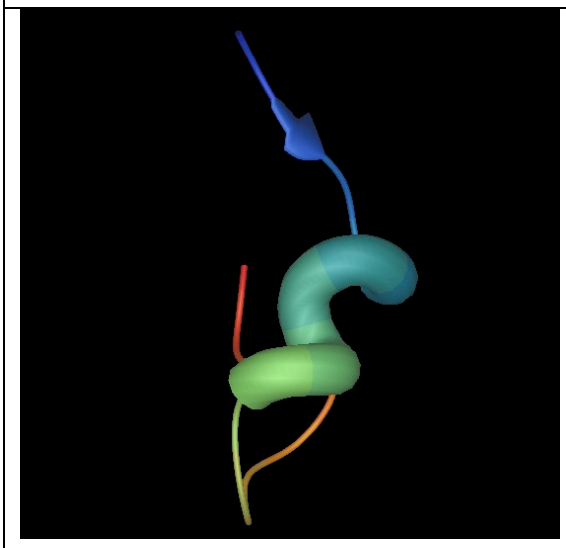
PKD-domain14



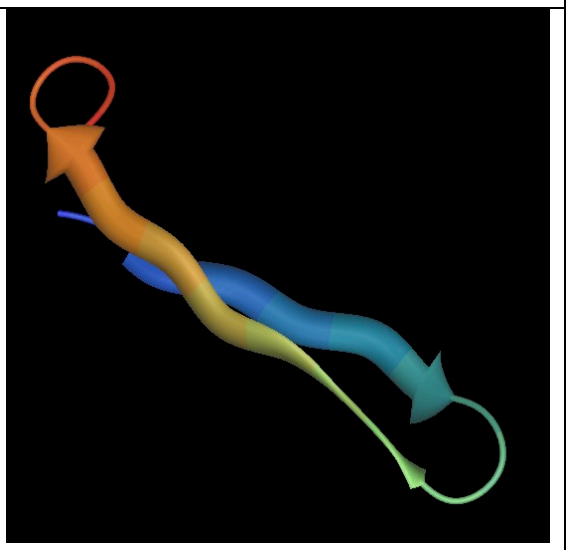
PKD-domain15



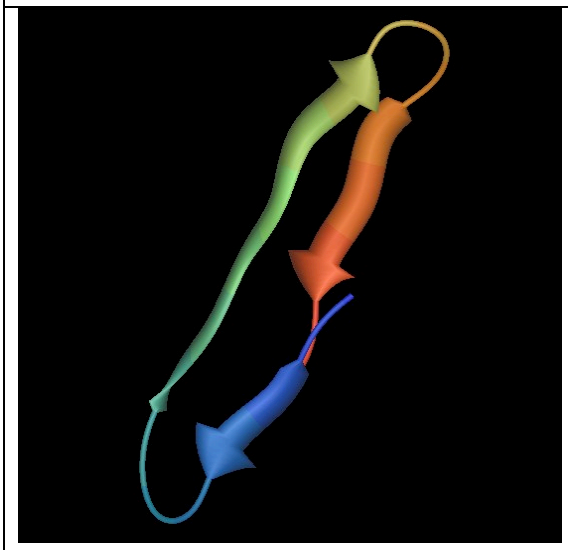
PKD-domain16



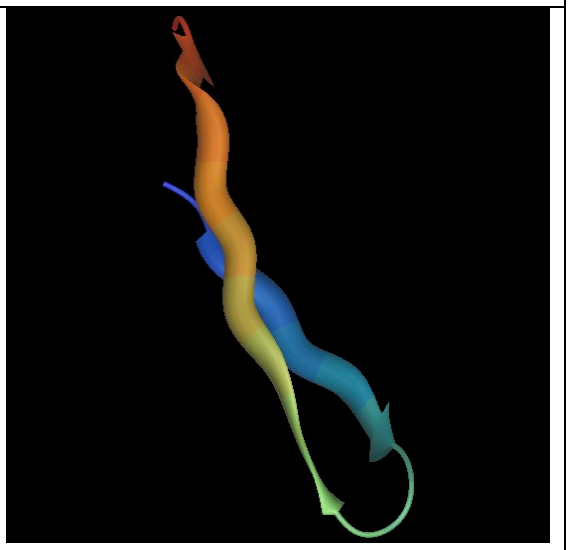
Transmembrane 1



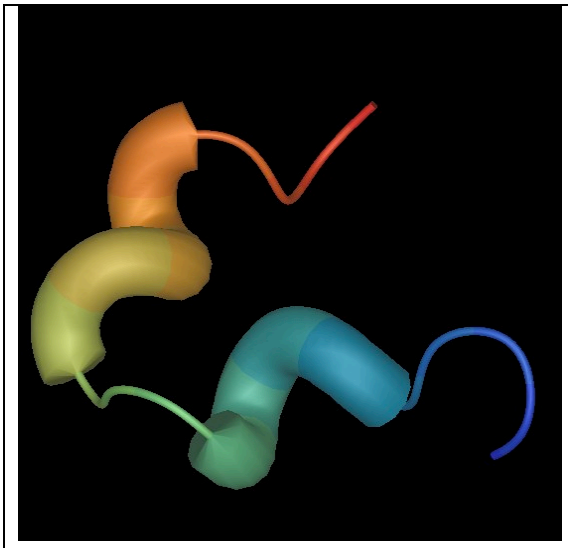
Transmembrane 2



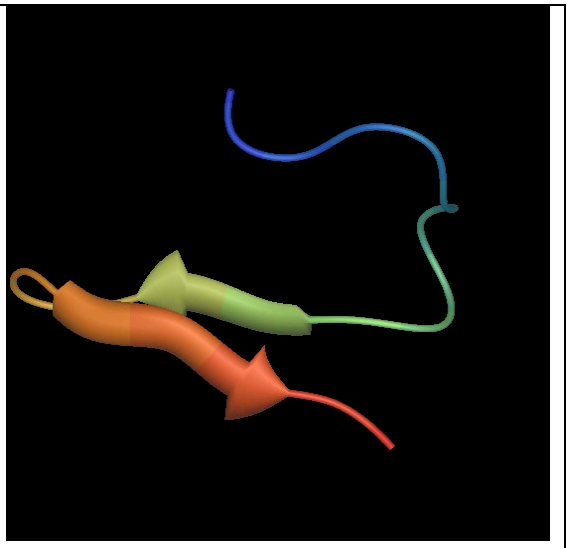
Transmembrane 3



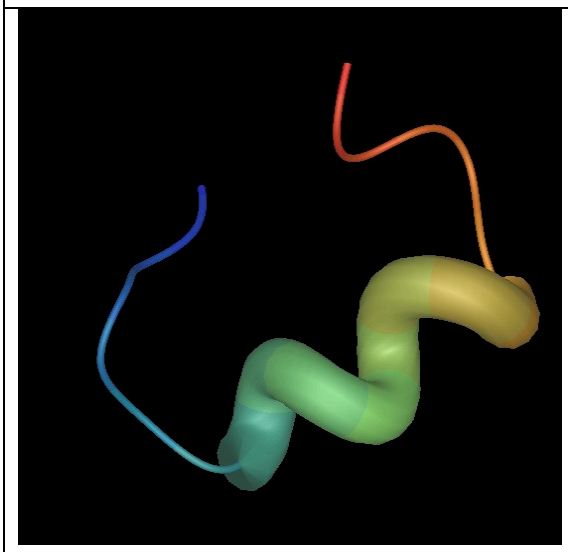
Transmembrane 4



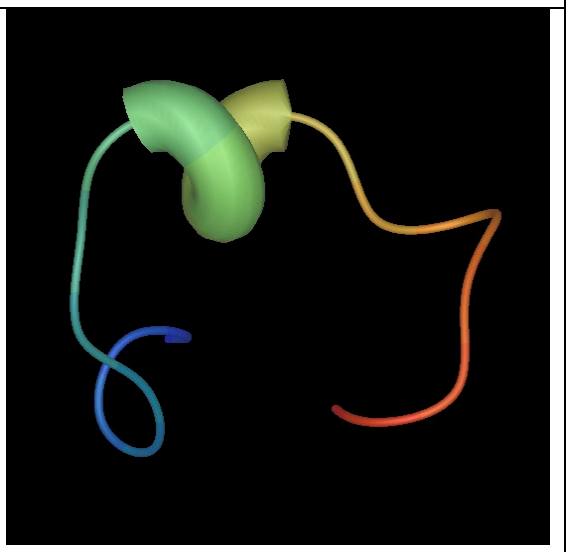
Transmembrane 5



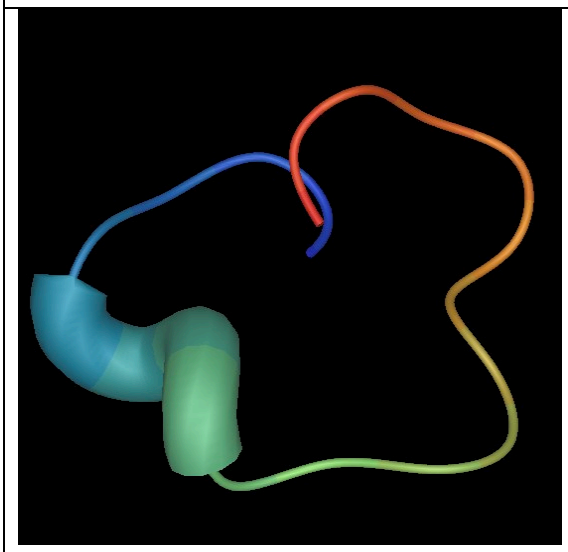
Transmembrane 6



Transmembrane 7

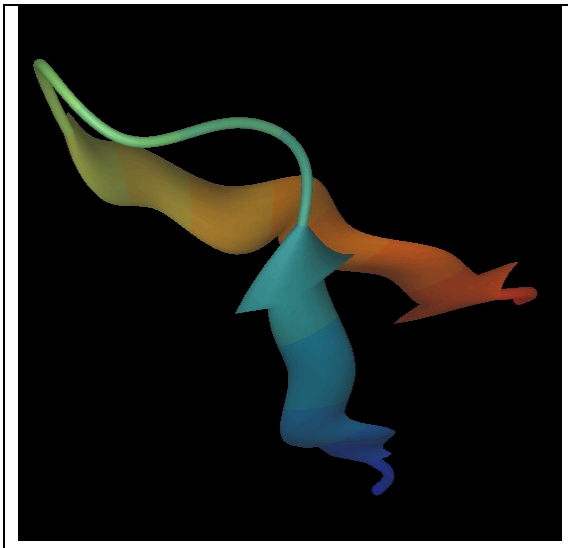


Transmembrane 8

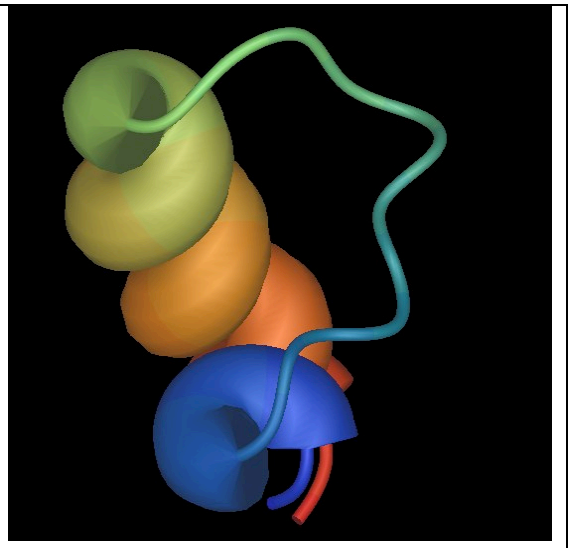


Transmembrane 9

Transmembrane 10



Transmembrane 11



Coiled-Coil

Figure 26: Rosetta 36 domain structural model.

From these results it should be pointed that even tough most of the domains present the expected structure, the tertiary structure in concordance with that predicted from the secondary structure, there are a number of domains where the tertiary structure is far from what it was expected (predicted by secondary structure analysis) as in the case with transmembrane domains 2, 3, 4, 6, 10 and 11, domains that were modeled as beta-sheets instead of as alpha-helix (alpha-helix is more typical structure for transmembrane domains and for channels) and LRR1--LRR2 where both domains were modeled as different structures, and due to their high level of sequence homology both domains should share tertiary structure.

PKD domains show beta-sheet folds in every case but the arrangement of the beta-sheets differ from one domain to another.

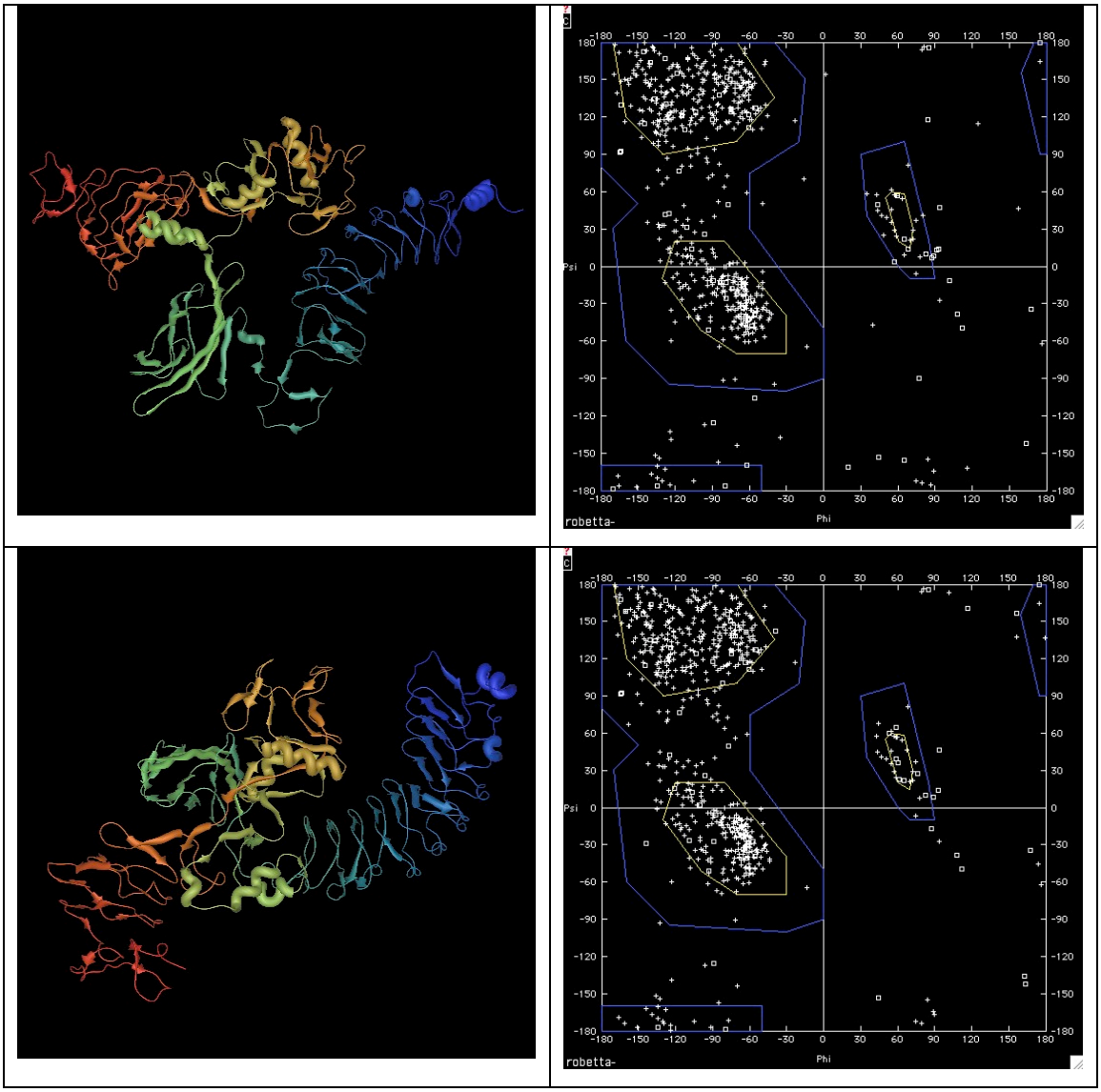
These results indicate that modeling each domain independently can be an error prone process and that the influence of surrounding amino acids and domains can be crucial for final domain structure determination.

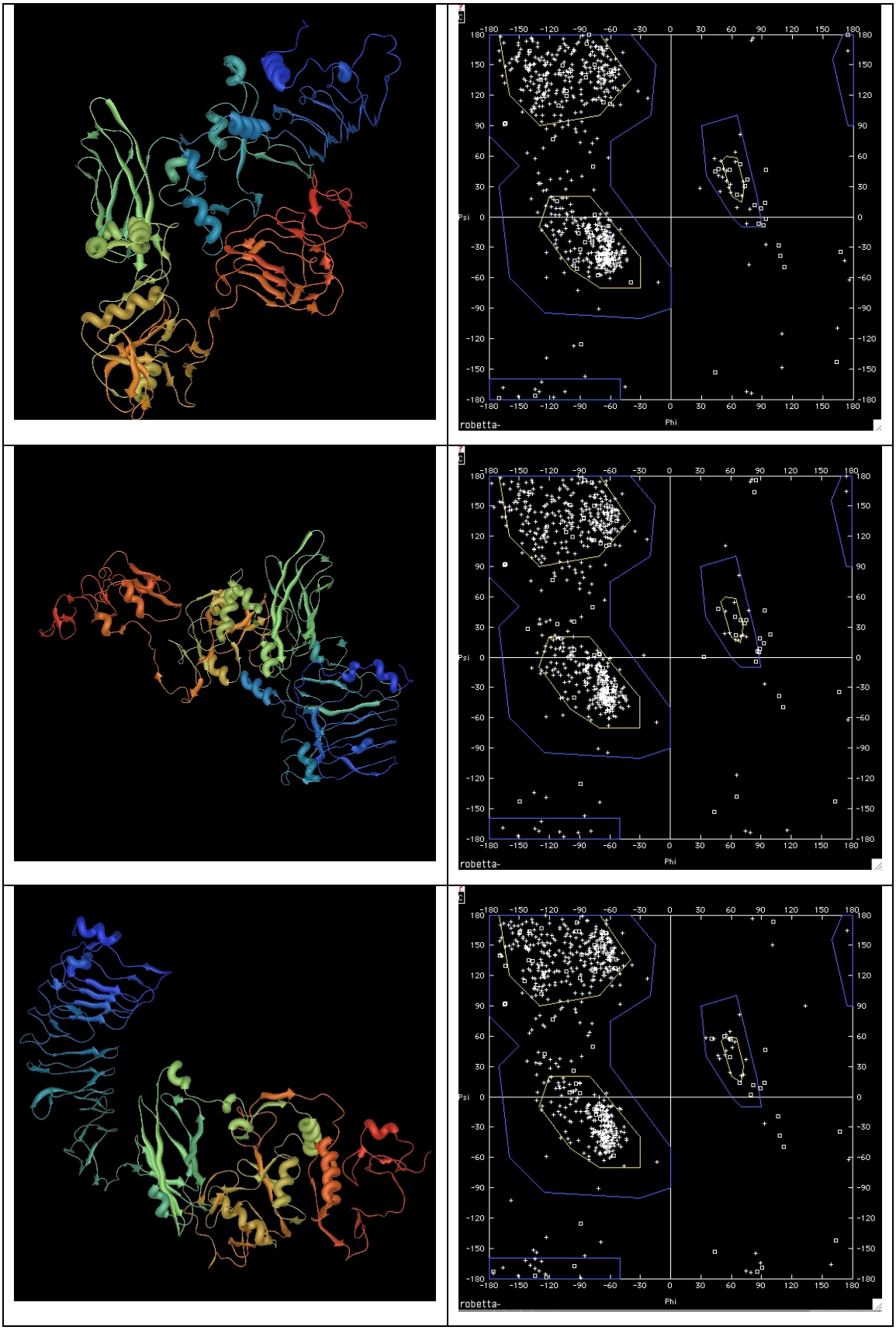
Robetta. Six fragments were used:

For each fragment 5 models were obtained.

Fragment 1: from a.a. 1 to 680. Domains that includes are signal peptide, amino flanking region, LRR, carboxy flanking region, WSC domain, PKD domain 1, C-type lectin domain, and LDL-A related motif.

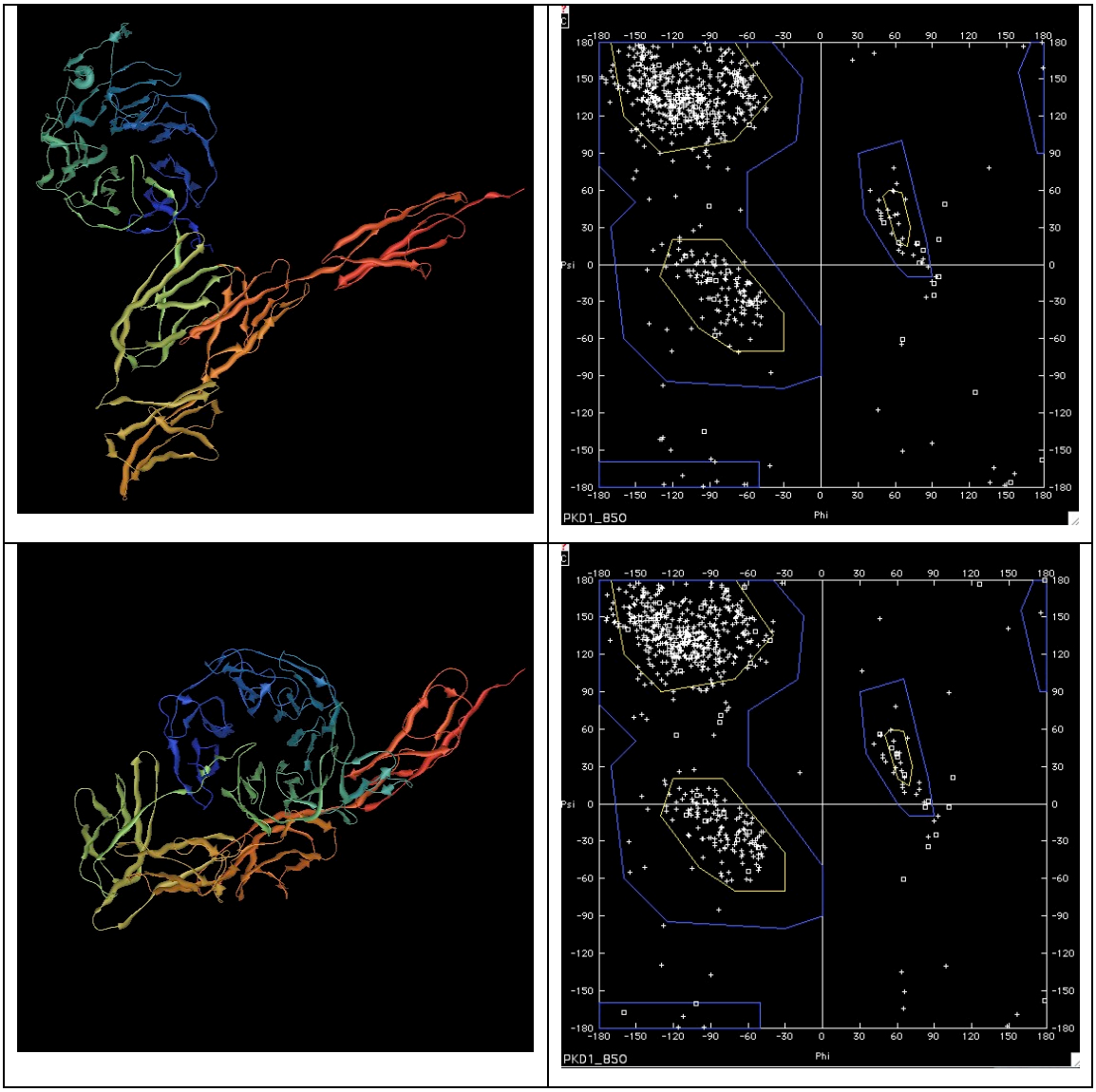
In this model we can distinguish a long antiparallel beta-sheet corresponding to LRR 1-2 and to the LRR flanking regions. The first amino acids of the protein adopt an alpha-helix structure that could correspond to the signal peptide. PKD1 domain present the expected structure corresponding to that described at the Brookhaven protein databank. We should point the presence of various alpha-helix mostly in the interdomain regions.

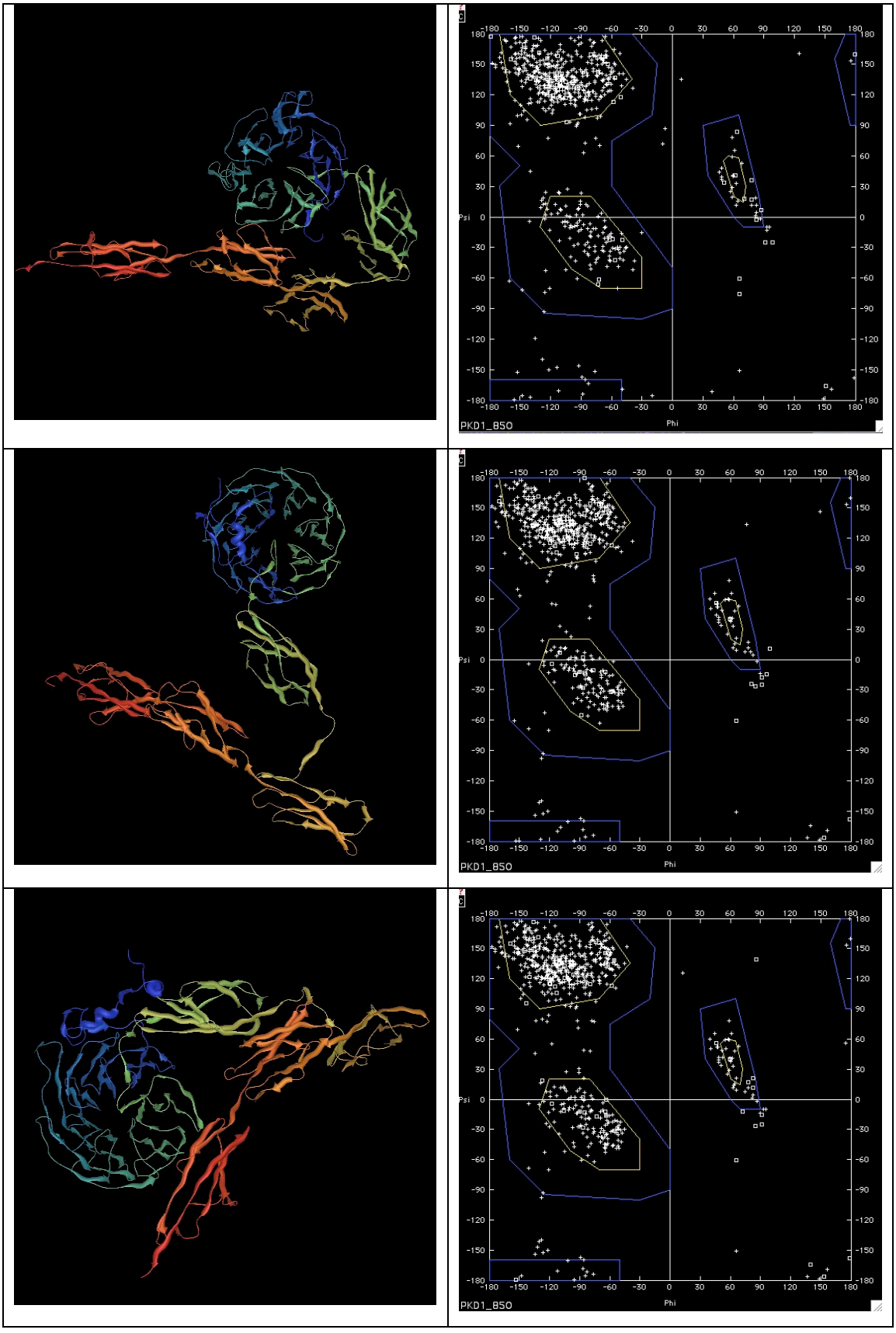



Psi
Phi
robetta-

Fragment 2: from a.a. 850 to 1550. Domains PKD domain 2 – 9.

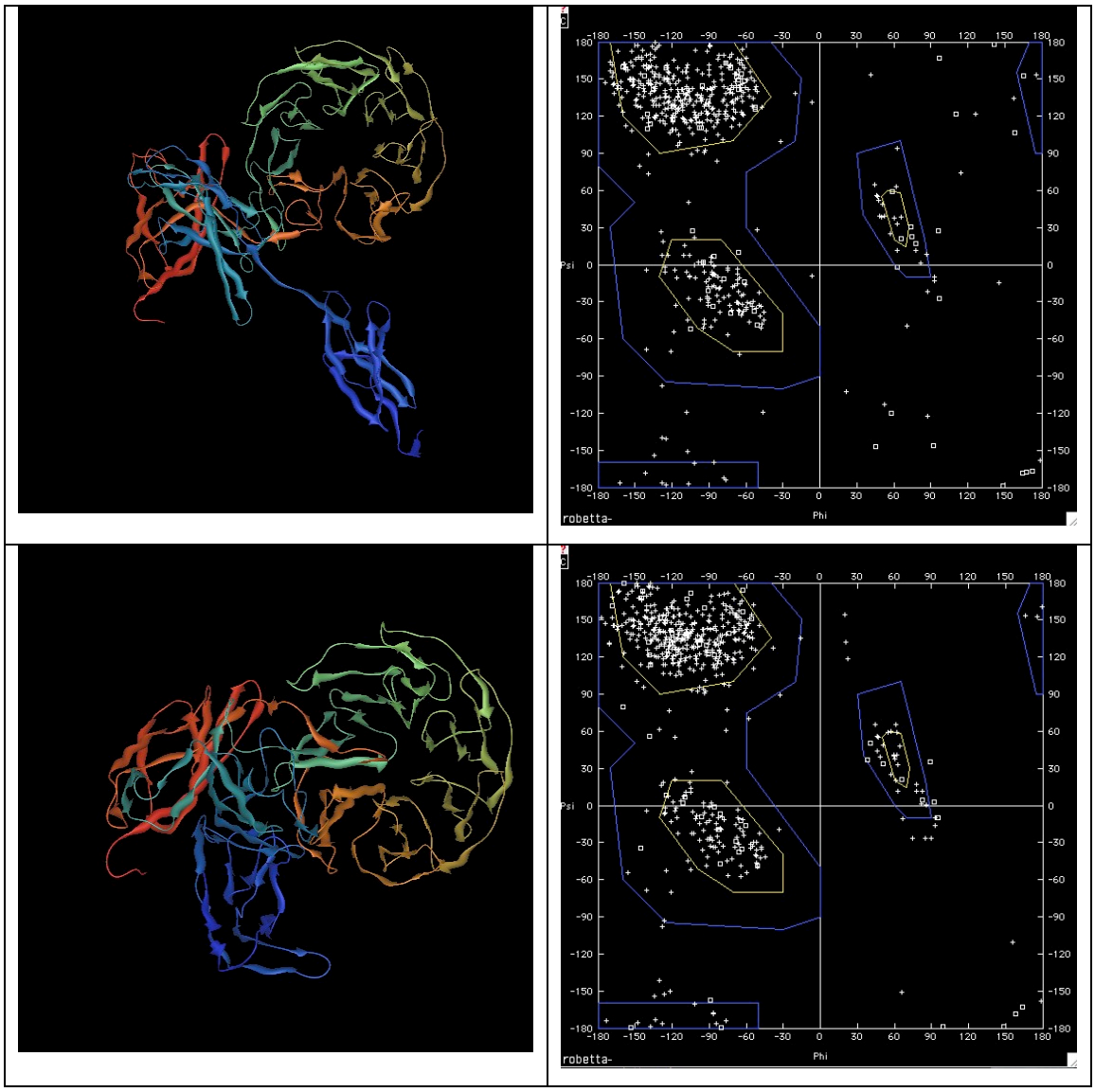
As predicted with the Robetta method, PKD domains adopt an all beta structure but, far from expected, these beta-sheet arrange in a special way to form a 7-bladed beta-propeller (PKD domain 2-5 and part of domain 6).

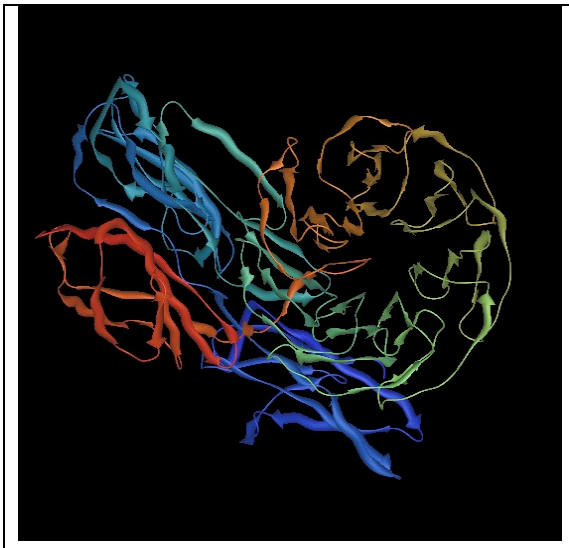


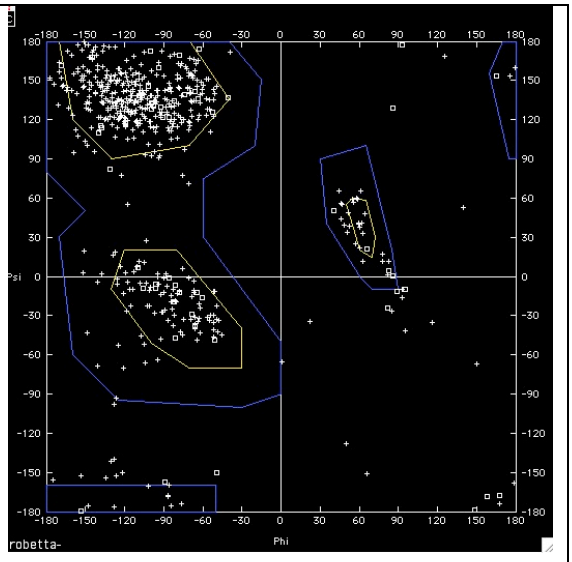

Psi
Phi
PKD1_850

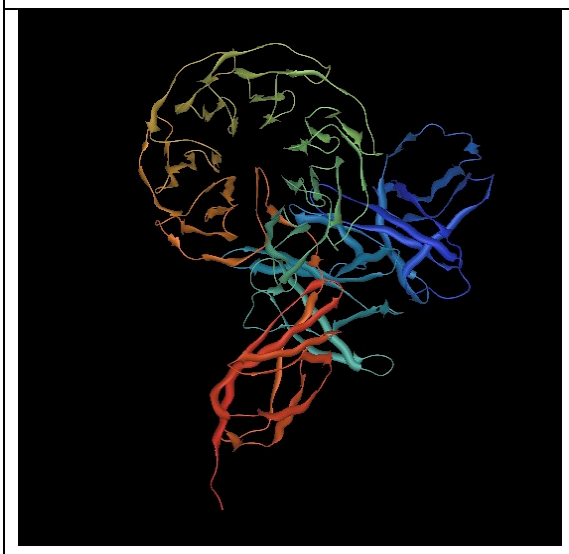
Fragment 3: from a.a. 1550 to 2146. Domains PKD domain 10 – 16.

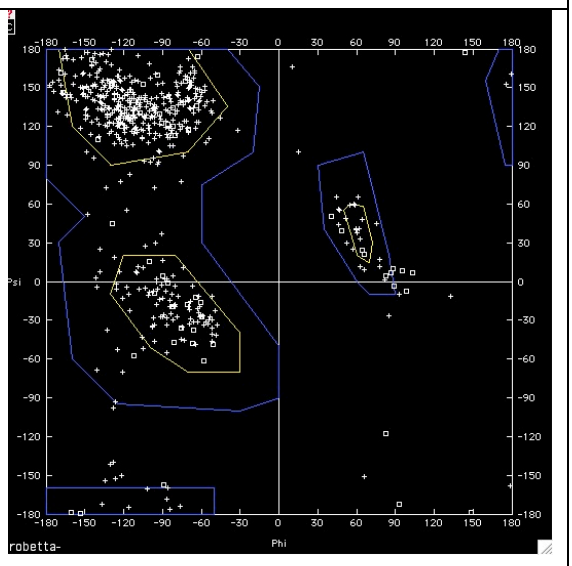
As in the case with the previous fragment, PKD domains arrange to form a 7-bladed beta-propeller, in this case the 7 blades correspond to the domains 12-15 and part of PKD domain 16. PKD domains form an all beta structure.

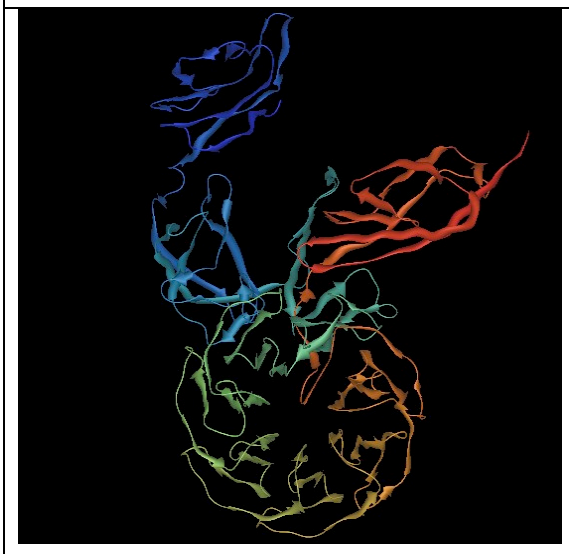


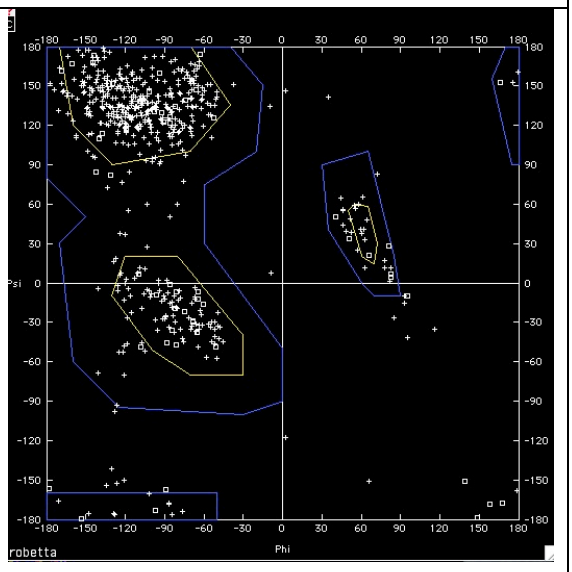



Psi
Phi
robetta-



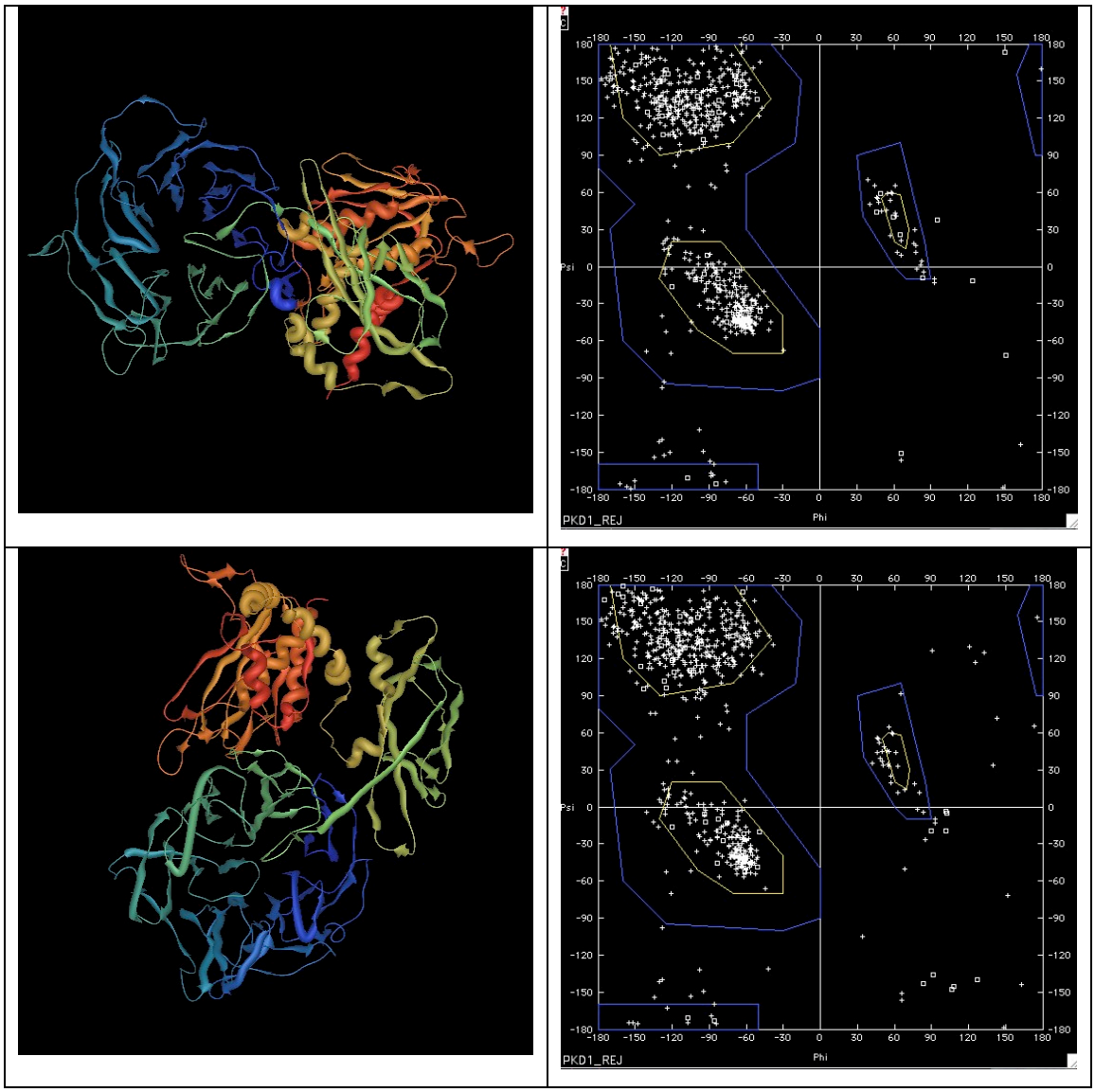

Psi
Phi
robetta-

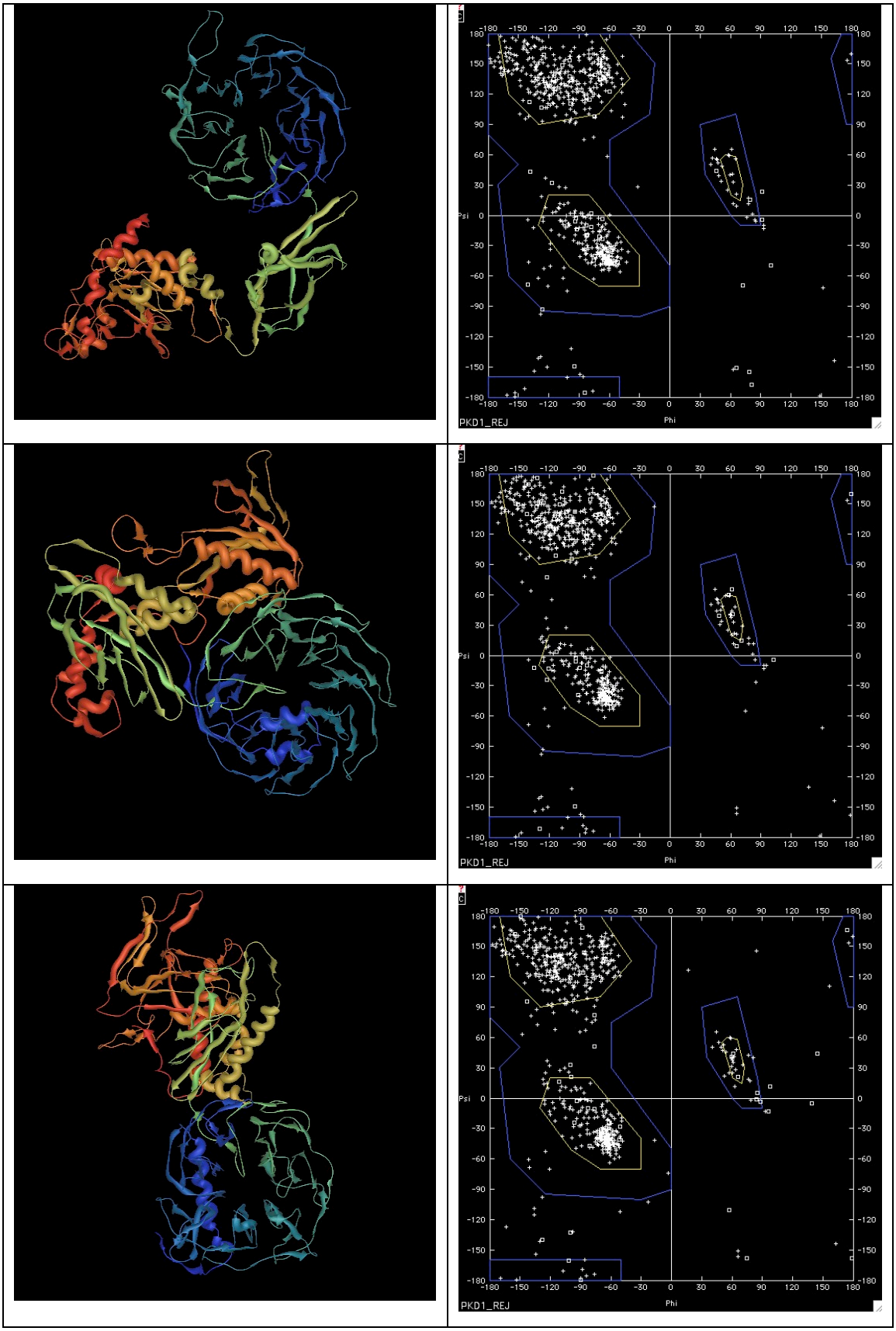



Psi
Phi
robetta

Fragment 4: from a.a. 2146 to 3110. REJ domain.

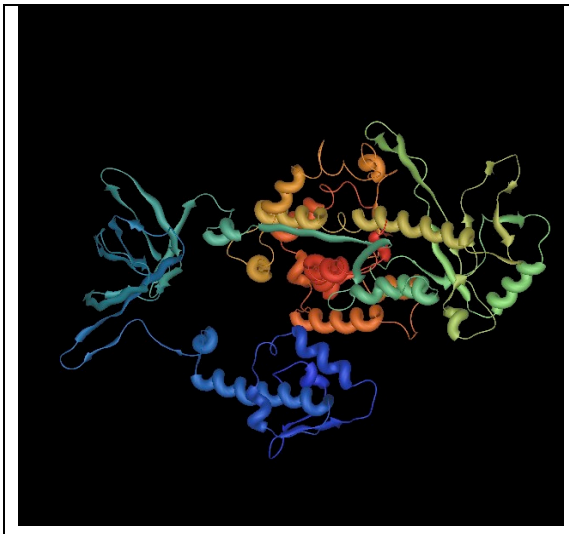
First half of REJ domain arrange to form a 7-bladed beta-propeller and a structure close to that described for PKD domain 1, the second half of the protein forms an alpha-beta complex.

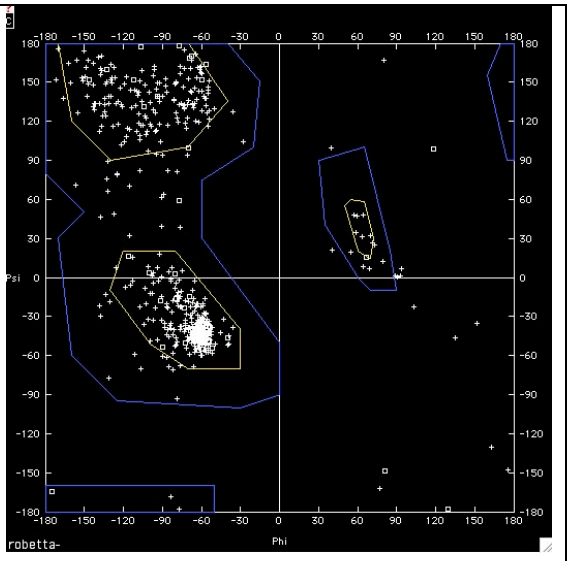


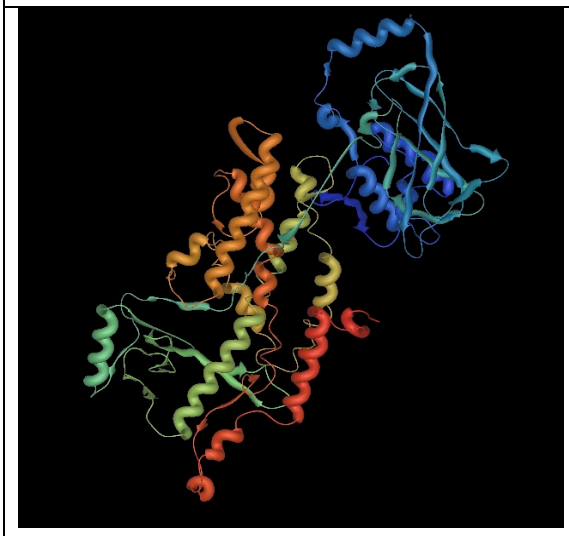

Psi
Phi
PKD1_REJ

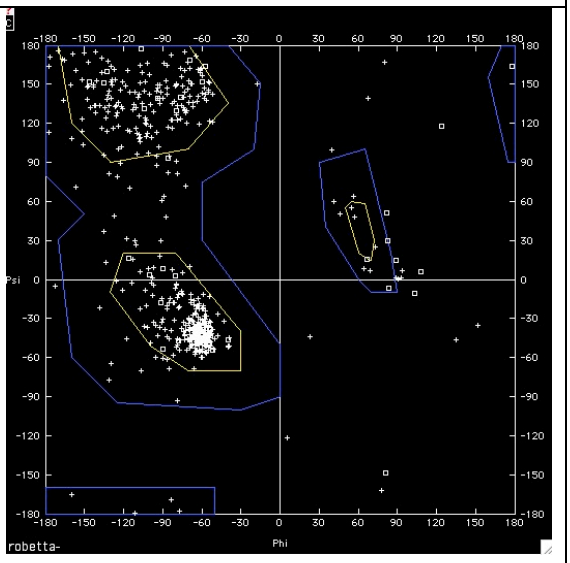
Fragment 5: from a.a. 3012 to 3580. Domains GPS, PLAT/LH-2 and Transmembrane (TM) 1 – 4.

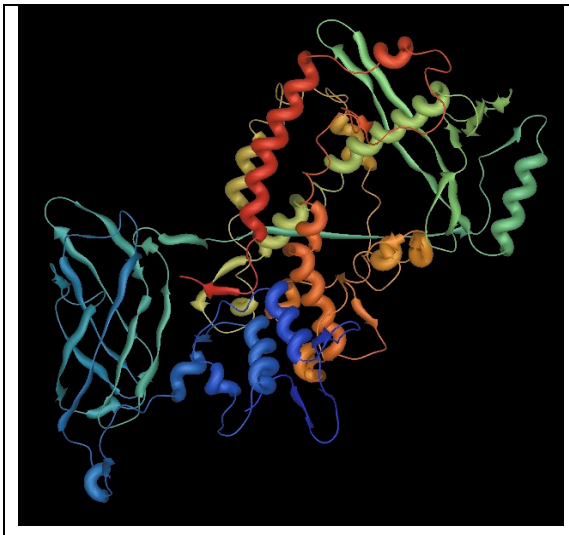
As expected for this fragment we obtained a high number of alpha-helix corresponding to the transmembrane domains. The big beta-sheet structure in this fragment corresponds to the PLAT domain, which is located in the citoplasmatic part of the protein.

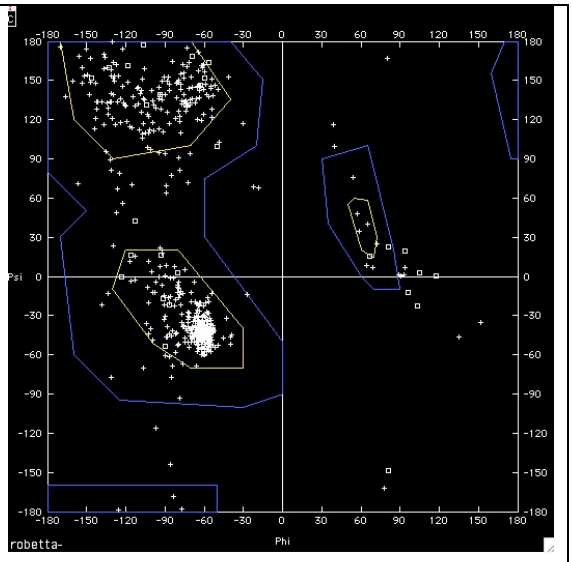


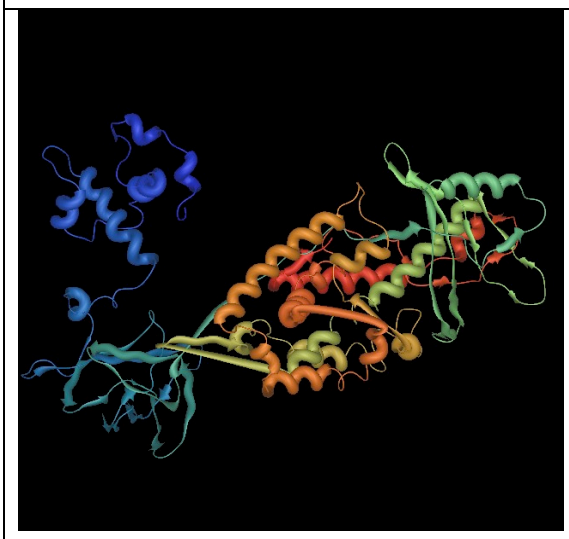


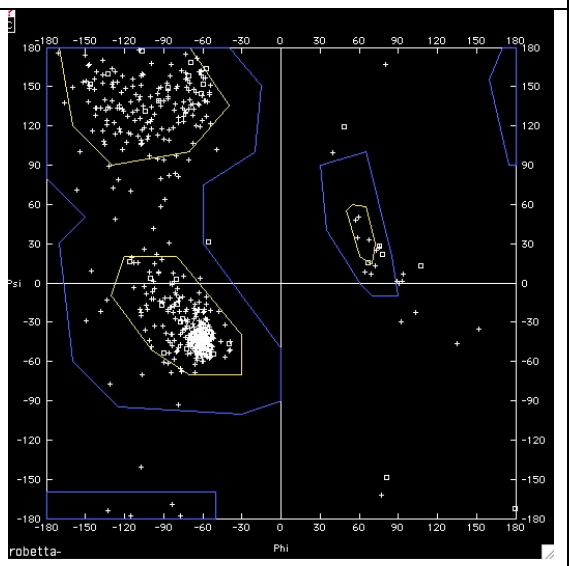


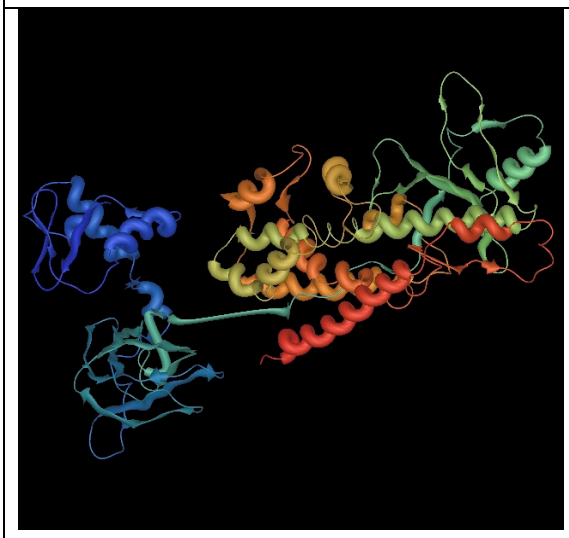


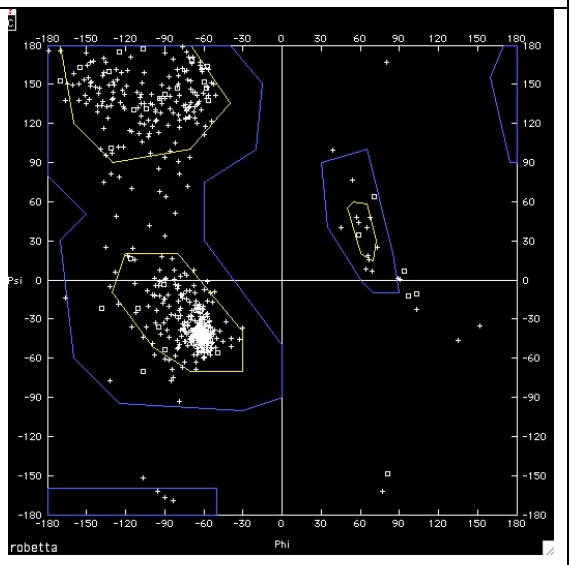



Psi
Phi
robetta-



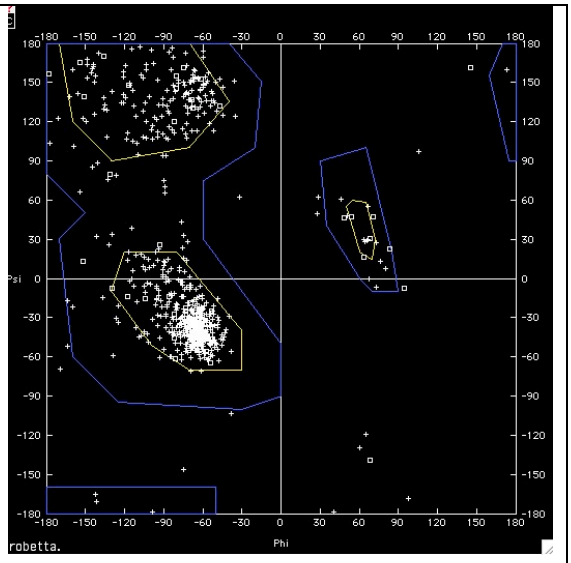

Psi
Phi
robetta-

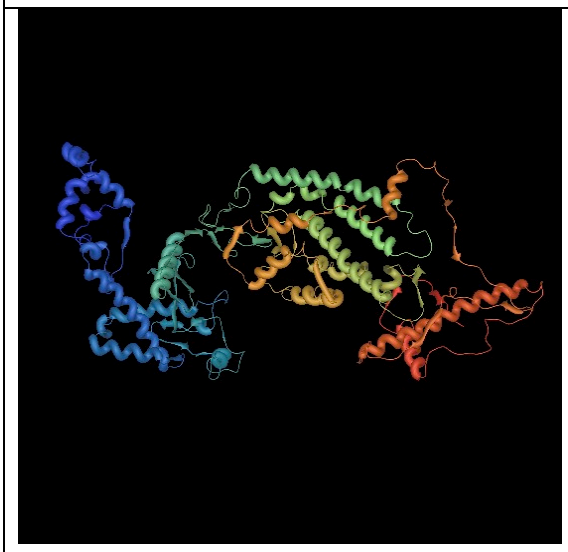


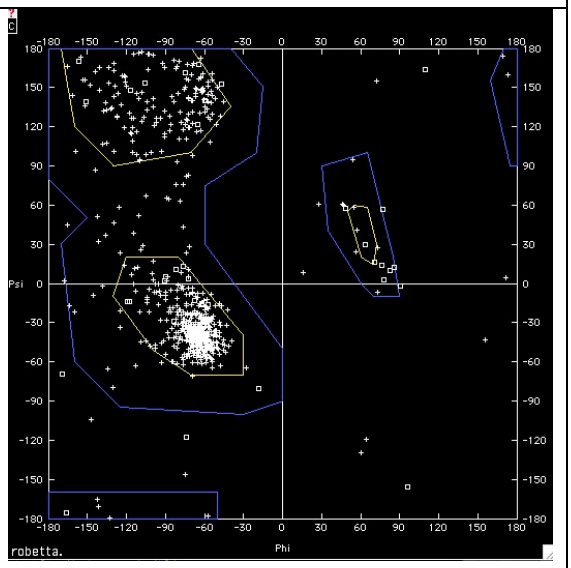

Psi
Phi
robetta

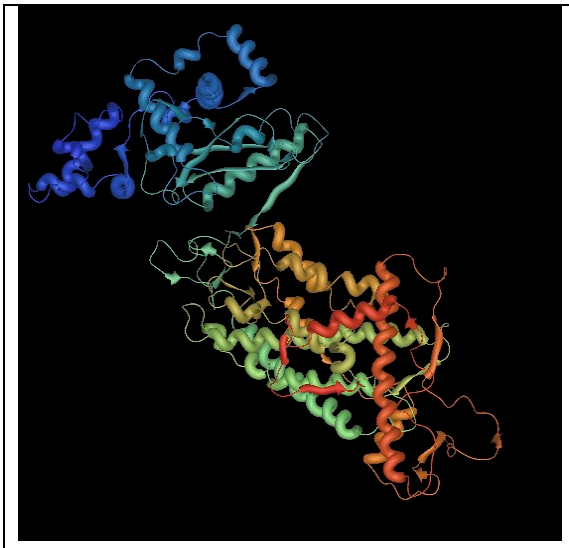
Fragment 6: from a.a. 3580 to 4301. Domains TM 5 – 11 and Coiled-coil.

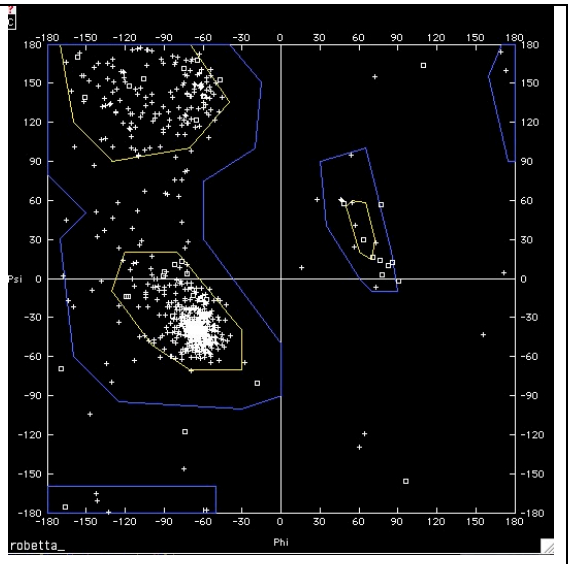
All alpha structure corresponding to the transmembrane domains and to the coiled-coil domain. It should be noted the presence of some small beta-sheet mostly at the loop regions in the intra and extracellular part of the protein.

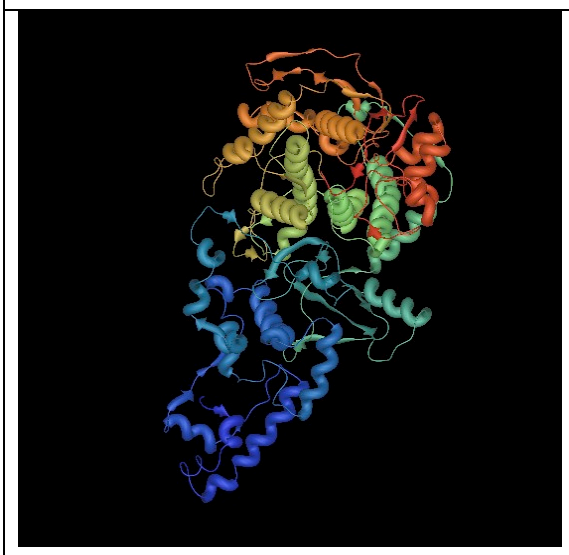


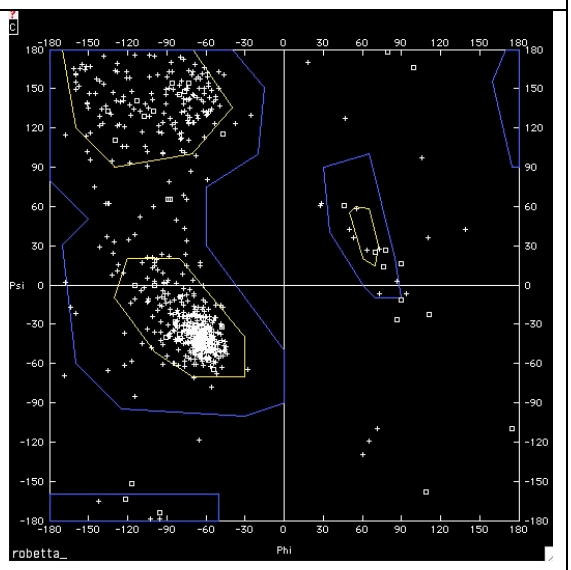


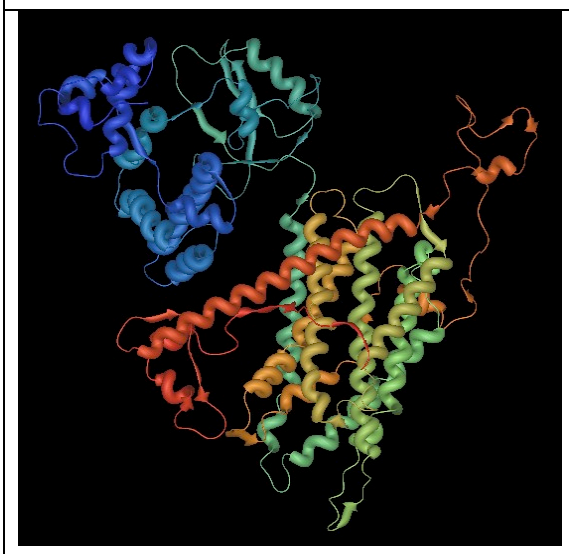


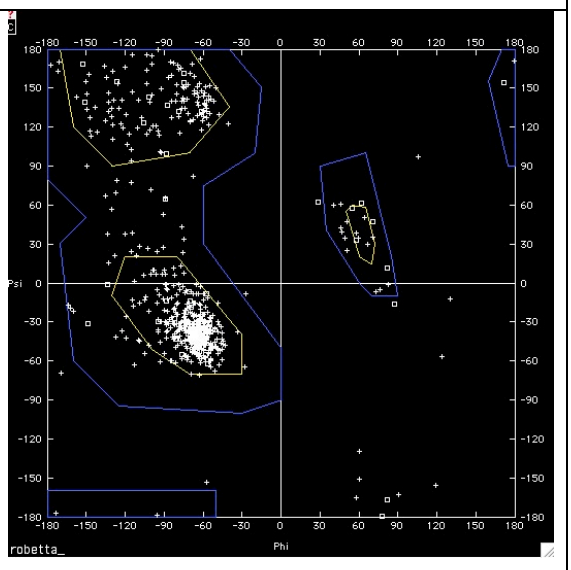



Psi
Phi
robetta_




Psi
Phi
robetta_




Psi
Phi
robetta_

First thing to note from the structural models is that it can be distinguished a predominance of beta-sheets in the intra and extra cellular parts of the protein and a higher number of alpha-helix in the transmembrane region.

In the extracellular part of the protein (models 1-3), there exist at least three 7-bladed beta-propellers (2 for the PKD domains and 1 for REJ). For the PKD domains, each beta-propeller is formed by 4 complete domains plus part of a fifth domain, as in the modeling process the 15 PKD domains that lie together were splited,, we believe that a third beta-propeller within this region is located between the 2 beta-propellers already described so the distribution of the beta-propeller would be: First beta-propeller from PKD domain 2 to PKD domain 5 PKD domain 6 acts as link with the next one, second beta-propeller from PKD domain 7 to 10, PKD domain 11 acts as link and third beta-propeller from PKD domain 12 to PKD domain 15. PKD domain 16 would link with the beta-propeller at the beginning of the REJ domain. 7-bladed beta propellers are folds that consist in seven four stranded beta sheet motifs. Among the group of proteins belonging to this fold there is the integrin alpha superfamily. This proteins acts in the cell binding process, function that was also described for polycystin-1.

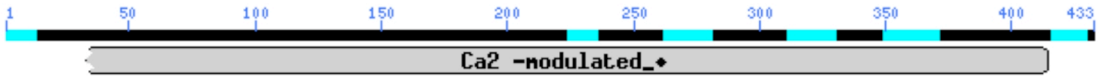
In the transmembrane region the abundance of alpha-helix structures reminds those structures of membrane channels and transporters. As it will be shown this was confirmed in the following experiments.

It should also be noted the presence of a beta-fold in the intracellular part of the protein that reminds that fold described for Lipoxigenase.

Transmembrane domains 6-11

Alignment against the sequence of PKD2 gene (non-specific cation channel Ca2+ mediated) shows high level of similarity with the region corresponding to the transmembrane domains 6-11 of polycystin-1 (result from alignment shown below),

The amino acid sequence of the transmembrane for protein polycystin 1 was used to query the Conserved Domain Database using KOG v1.00 as dataset (Marchler-Bauer A, *et al.*). From the results of the search it was possible to identify the protein domains Transmembrane 6 – 11 as part of the KOG3599 group, being KOG3599 a Ca2+ modulated nonselective cation channel polycystin conserved domain.



gnl|CDD|21380, KOG3599, Ca2+-modulated nonselective cation channel polycystin [Inorganic ion transport and metabolism, KOG3599, Ca2+-modulated nonselective cation channel polycystin [Inorganic ion transport and metabolism, Signal transduction mechanisms]

CD-Length = 798 residues, only 45.5% aligned
Score = 212 bits (542), Expect = 5e-56

```
Query:   32  IKQELHSRAFLAITRSEELWPWMAHVLLPYVHGNQSSPELGPPRLRQVRLQEALYPDPPG   91
Sbjct:  291  ARMIAYENRLLGVPRLRQLRVRNSQ-SCLVLDGFQ-------DSLYIVECYLVYSSDPED  342

Query:   92  PRVHTCSAAGGFSTSDYDVGWESPHNGSGTWAYSAPDLLGAWSWGSCAVYDSGGYVQELG  151
Sbjct:  343  --------DKPWSPYGPWAGDE------FTYSTSKELLLGLEHWGLLASYGGGGYVVLLS  388

Query:  152  LSLEESRDRLRFLQLHNWLDNRSRAVFLELTRYSPAVGLHAAVTLRLEFPAAGRALAALS  211
Sbjct:  389  LSRTESLKAISYLRENNWLDRGTRAVFIDFTLYNADINLFCVVTLRVEFPPTGGVLPSLQ  448

Query:  212  VRPFALRR-LSAGLSLPLLTSVCLLLFAVHFAVAEARTWH-REGRWRVLRLGAWARWLLV  269
Sbjct:  449  LESFKLLRYVSAGSSLIMLCEVVFLLFVLYFAVAEGLKIWIHRLGRYVRSKWNWLDLAIV  508

Query:  270  ALTAATALVRLAQLGAADRQWTRFVRGRPRRFTSFDQVAHVSSAARGLAASLLFLLLVKA  329
Sbjct:  509  LLSVVLLVLMITRTGLADGVLTGFERASPRTFIDFTEVAQWNIAARNLLAFLVFLTTIKL  568

Query:  330  AQHVRFVRQWSVFGKTLCRALPELLGVTLGLVVLGVAYAQLAILLVSSCVDSLWSVAQAL  389
Sbjct:  569  WKVLRFNKTMSQFSSTLSRAWKEIVGFALMFLILFFAYAQLGYLLFGNQVSDFRTFVASI  628

Query:  390  LVLCPGTGLSTLCPAESWHLSPLLCV   415
Sbjct:  629  VTLLRYI-LGDFCPAEIFHANRILGP   653
```

Some of the proteins belonging to this group are:

Polycystin 1 [Homo sapiens]
Human Polycystic Kidney Disease PKD2 related PKD-2 (pkd-2) [Caenorhabditis elegans]
Hypothetical protein ZK945.9 in chromosome II
Similar to KIAA1879 protein [Homo sapiens]
Polycystic kidney disease 1-like 1 protein (Polycystin 1L1)
Polycystin 2 [Homo sapiens]
Polycystic kidney disease and receptor for egg jelly related protein precursor (PKD and REJ homolog)
Polycystic kidney disease 2-like 2 [Homo sapiens]
Polycystic kidney disease 2-like 1; polycystic kidney disease (polycystin)-like; Polycystin-L [Homo sapiens]

Modeling of the transmembrane regions 6-11 alone was performed using Robetta server and a structural alignment was performed with the obtained model using Dali server.

With the sequence corresponding to the transmembrane domains a prediction of function was also made using ProtFun 2.0 server. The result from ProtFun predicts this sequence to belong to the functional category of Transport and binding proteins and to the Gene ontology category of Transporters (results not shown).

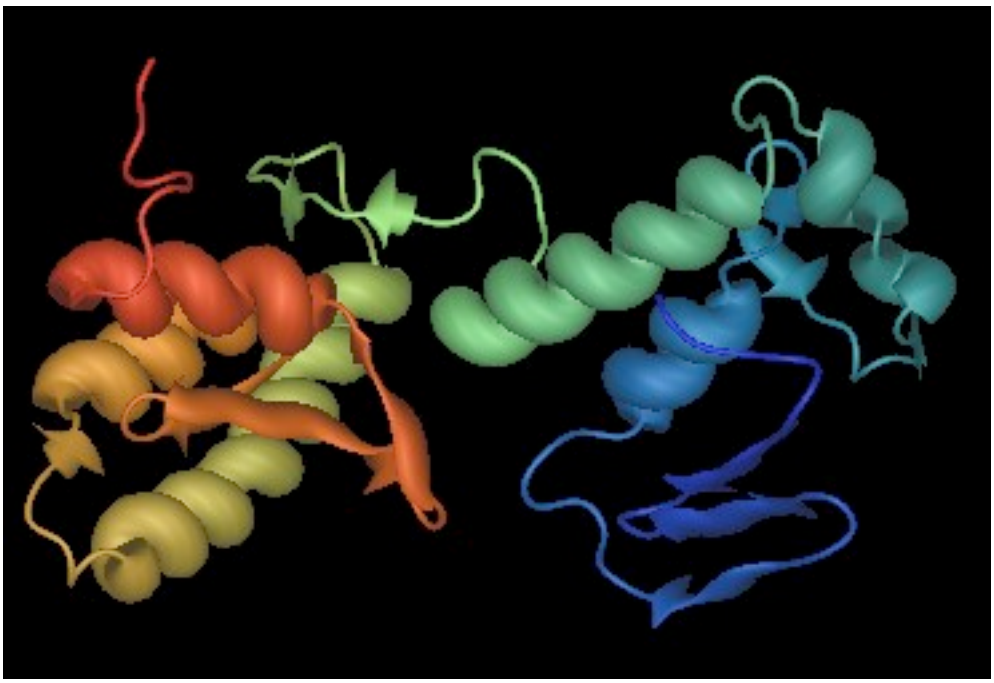


Figure 33: Model of the TM6-11 of polycystin-1.

Tertiary structure validation

Only one model for each set of 5 was selected from the models obtained from the previous step. Selection was made based on the Ramachandran plot (images shown above).

Table 8: List of structural models selected for validation.

Polycystin-1 section	Model	Amino-acids out of core region	Amino-acids making clashes with backbone
1-680	Model 4	14	3
850-1550	Model 2	17	2
1550-2146	Model 4	12	1
2146-310	Model 3	7	2
3012-3580	Model 1	11	1
3580-4301	Models 2-3	9 (for both models)	0 (for both models)

For the models selected three different validation methods were performed (Procheck, Prove, What If).

These methods were automatically performed by the use of Biotech Validation Suite at E.B.I. web server.

Table 9: Result for Vol-Score and Proc-ave for each of the 6 models studied.

Polycystin model	VOL-SCORE	PROC-AVE
1-680	1.89	0.0200
850-1550	2.90	-0.0900
1550-2146	2.05	0.1000
2146-3110	2.33	0.0900
3012-3580	2.15	0.2600
3580-4301	2.20	0.2400

VOL-SCORE by PROVE: The structure value is the Z-score rms for the atoms in the structure. Note that the volumes are computed only for buried atoms. This value is expected to be around 1.0 for an average structure. Higher numbers are indicative of more unusual atomic volumes.

Values are considered poor if they are larger than 1.15.

Values are considered bad if they are larger than 1.30.

The values for our models and the reference ones were calculated for proteins with a resolution of 1.5 Å, so the scores obtained were higher than expected. One reason is due the resolution of the protein models might be lower than the one chosen for the analysis (the higher the value, the lower the resolution). As models obtained from Rosetta have no information about resolution we used in our analysis a resolution value of 1.5 Å as is the value used for the reference quality scores.

PROC-AVE: Overall average G factor for the protein by PROCHECK. The overall G value for the structure represent a carefully weighted average of all the analyses performed by PROCHECK

WHAT-CHECK gave an error for all structural models as ab initio structures lack information required by what-check to perform the structure analysis.

As expected validation studies gave as a result that protein models were not accurate enough to be considered as final structural folds and to be used for ligand binding studies, but good enough for the study of function from structure.

Tertiary structure comparison

Results from Dali alignments are shown in the following table. Only a selected set of aligned proteins is displayed.

Table 10: Most relevant results from Dali structural alignments. Proteins were selected based on the alignment RMSD (Root Mean Square Desviation) score and length of alignment. All of the protein alignments had a Z-score higher than 2. (Data not shown).

Model	Aligned proteins
1-680	Different fragments of tropomodulin protein. Internalin Invasin Fibronectin Gtpase-activating protein Tenascin Human vascular cell adhesion molecule. Interleukin Intercellular adhesion molecule-2 T-cell antigen, receptor and surface glycoprotein cd4 Mucosal addressin cell adhesion molecule-1
850-1550	Surface layer protein Nitrous-oxide reductase Transducin (guanine nucleotide-binding protein) Actin interacting protein Integrin, alpha V Galactose oxidase Clathrin Neuraminidase Sialidase Neural cell adhesion Ciliary neurotropic factor receptor alpha fragment.
1550-2146	Surface layer protein fragment Nitrous-oxide reductase Actin interacting protein 1 Antiviral protein ski8 Integrin alpha fragment Galactose oxidase Clathrin Sialidase Neural cell adhesion molecule Ciliary neurotrophic factor receptor Adsorption protein

2146-3110	Surface layer protein Nitrous-oxide reductase Transducin Actin interactin protein 1 Integrin alpha V Sialidase Neuraminidase Neural cell adhesion Ciliary neurotrophic factor receptor Adsorption protein
3012-3580	Lipoxigenase Ferritin Gtp-binding Flagellar protein flis Rho-gef factor Laminarinase Upper collar protein Rhodopsin receptor Lactose permease Arfaptin 2 fragment Sensory rhodopsin II fragment
3580-4301	Rhodopsin Bacteriorhodopsin Cytochrome c oxidase Arfaptin 2 fragment Alpha 1 catenin Vinculin factor Retinoblastoma protein fragment Amphiphysin fragment Photosystem I p700 chlorophyll a apoprotein Photosynthetic reaction center
TM 6-11	Mechanosensitive channel protein 1mxm

Among these group of proteins with similar folds it is remarkable that there is a high abundance of hydrolase proteins (sialidase, neuraminidase, laminarinase), binding proteins (Ciliary neurotrophic factor receptor, tropomodullin, transducin), cell adhesion (Neural cell adhesion, Intercellular adhesion fragment), transduction and transport proteins (Arrestin, Rhodopsin,

Importin, lactose permease, ferritin, GTP-binding) and signalling proteins (Arfaptin, rho-gef factor, Rhodopsin).

It should also be paid attention to the fact that lipoxygenase (an oxido reductase protein) shows a high level of structural similarity with polycystin-1 segment 5 (length of alignment 412 a.a. and an RMSD of 1.2), this segment contains PLAT domain that was also identified as a Lipoxygenase by other sequence comparison methods.

Taking a deeper look at where those proteins described are aligned, conclusions about protein function can be inferred:

The extracelular part of the protein seems to play a role in cell adhesion and protein binding. The location of polycystin-1 at the cell basolateral membrane could support the role in cell adhesion, being the interaction among polycystins in different cells responsible of the maintenance of cell-cell adhesion. REJ domain is believed to be involved in ion channel regulation.

The intracellular part of the protein plays a role in oxido reductase process and in binding and signaling process.

The transmembrane region of the protein presents a high similarity with different transport and transduction proteins being these results compatible with those from Babich *et al*, where transmembrane domains 6-11 were described as a possible cation channel.

In view of these results we propose that, when polycystin-1 is located in basolateral membrane, it acts as a cell adhesion molecule. When placed in the cilia it acts as a signal receptor and a mechano-sensor. The beta-propellers would act as a mechano-sensor that controls the state of the cation channel (open/close); these structures would also act as filters for large molecules. Beta-propellers oscillation can avoid big molecules to block the cation channel pore and to reach the interior of the cell (Fülöp, V., *et al*).

FlexProt TM6-11 vs. 1mxm

The score of the alignment between both sequences give a RMSD value of 4.69754 Å for the 202 a.a. aligned. It can be easily seen that the regions aligned are those corresponding to the alpha-helix regions that form the channel pore, being the beta-sheet of 1mxm part of the mechanosensor. As seen in Kumánovics *et al.* the pore forming domains are the most conserved ones while evolution of the sensor module of different protein leads to many different structural conformations, and this would explain why PKD1 TM 6-11 lacks the beta sheet section.

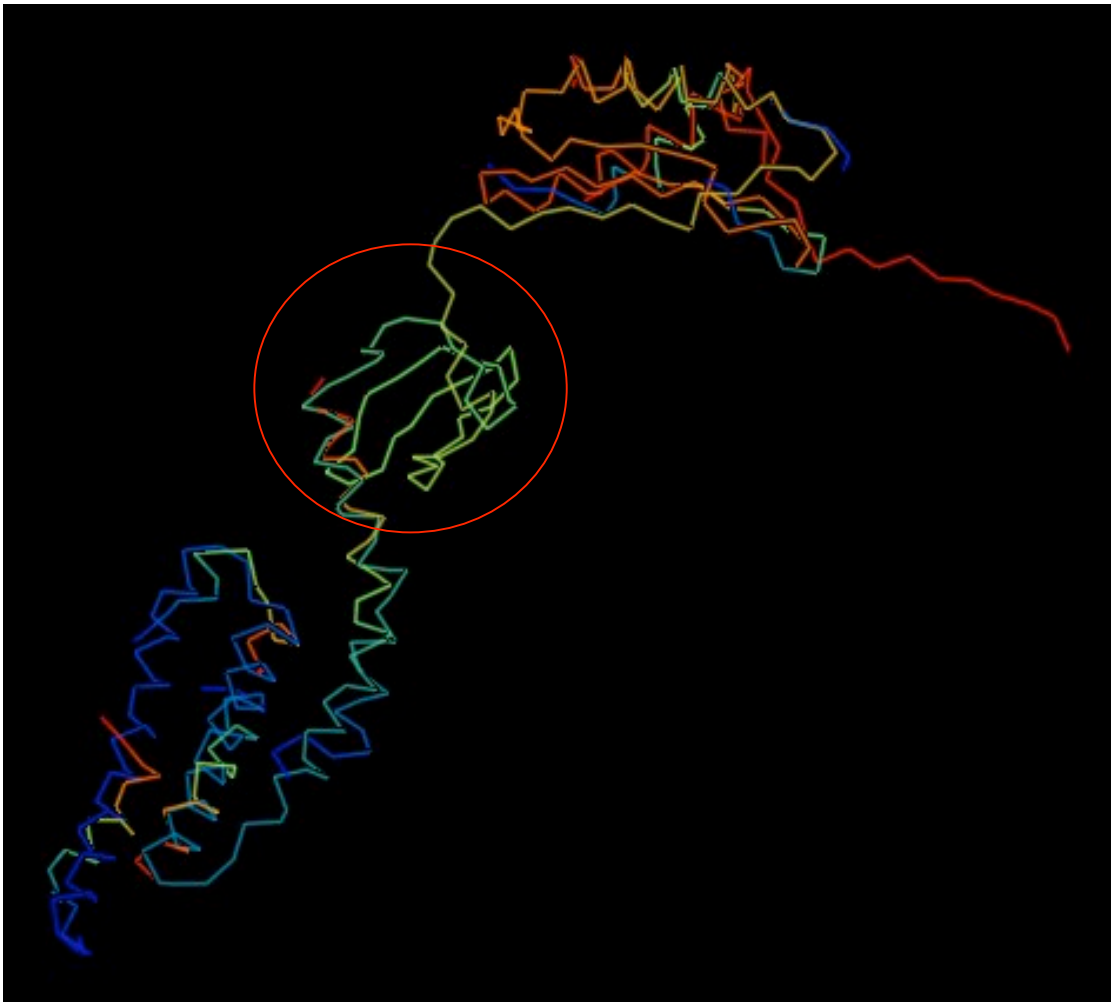


Figure 34: Alignment between polycystin-1 and 1mxm. Beta-sheet region exclusive of 1mxm is contained within the red circle.

As 1mxm forms a mechanosensitive channel, this alignment points to the possibility that polycystin-1 acts as a mechanosensor (as explained above), detecting urine currents that determine the state of the cation channel and activating different pathways.

Beta-propellers and PKD domains

During the study of the structural models obtained for the 15 PKD domains and for REJ domain it was observed that the structural fold of a 7-bladed ß-propeller was quite repetitive and it always seems to be a PKD fold domain right after one ß-propeller.

Studying the length of the fold we found out that it took 4 PKD domains to form such a structure so we generate sequence clusters containing the amino acid sequence of 5 PKD domains. We obtained the sequences:

PKD 2-6_844-1292, PKD 7-11_1305-1715 and PKD 12-16_1729-2142, and perform a multiple sequence alignment using ClustalX.

As we also observed a ß-propeller in the beginning of the sequence corresponding to REJ domain we add its sequence to the multiple alignment.

When aligning the secondary structure of the selected domains we add the sequence of Surface layer protein 1L0Q from Methanosarcina mazei because the three dimensional structure of this protein was used as a template to build the models containing the ß-propellers.

From the results obtained from the alignments we can distinguish that, even though the amino acid sequence homology is not as high as expected, the alignment of the secondary structures shows a high level of conservation in those regions corresponding to ß-sheets and a lower level of conservation corresponding to those amino acids at the loops. It should also the noted that the length of the alignment of the PKD clusters and 1L0Q covers the 100% of the sequence length and for REJ it coincides with an all ß-sheet region, being the secondary structure for the part of REJ that is not aligned completely different from that of the aligned region, what it could indicate a subdivision of REJ into two different structural domains.

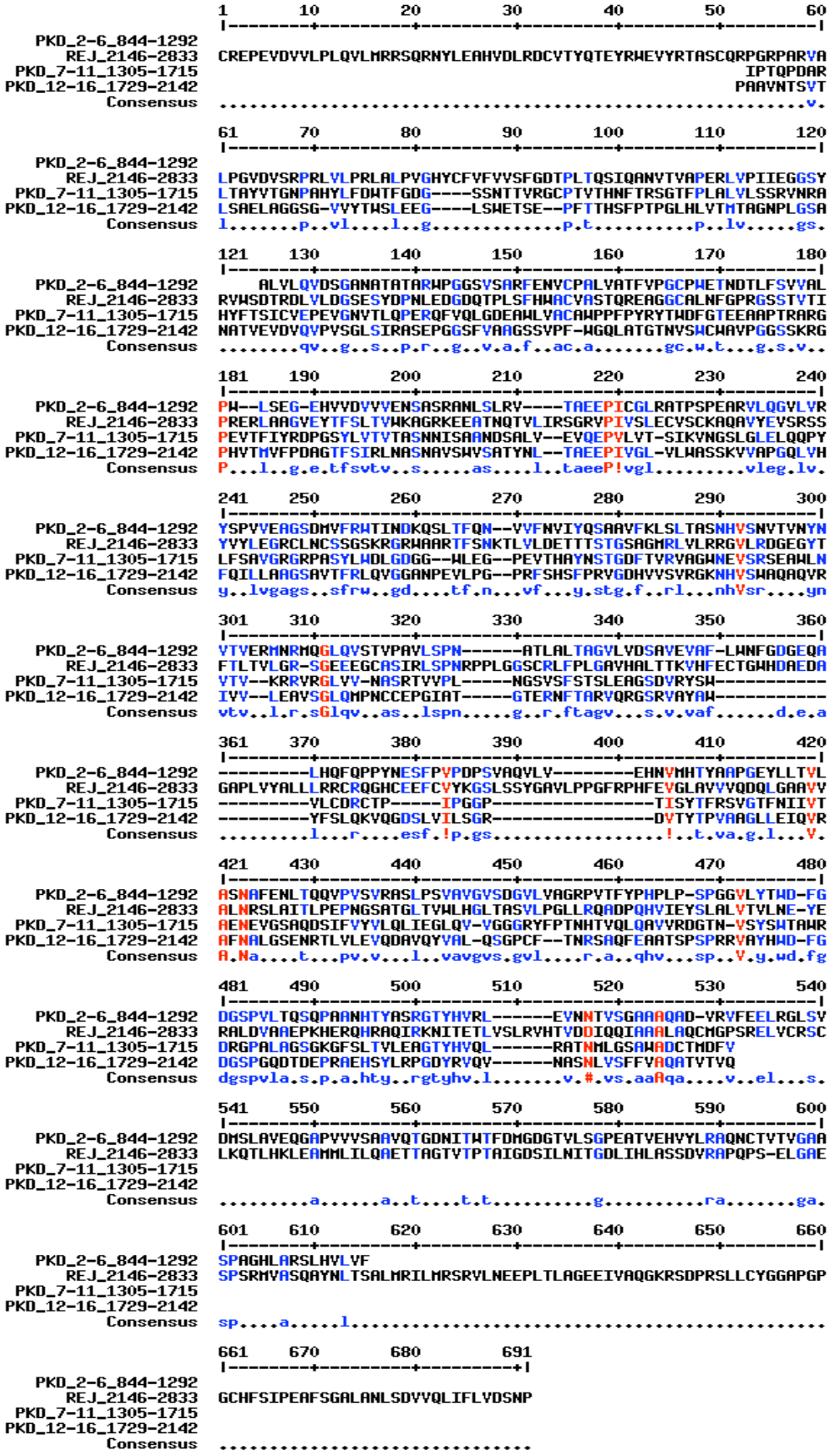


Figure 35: Alignment of the amino acids the three PKD clusters and REJ domain.

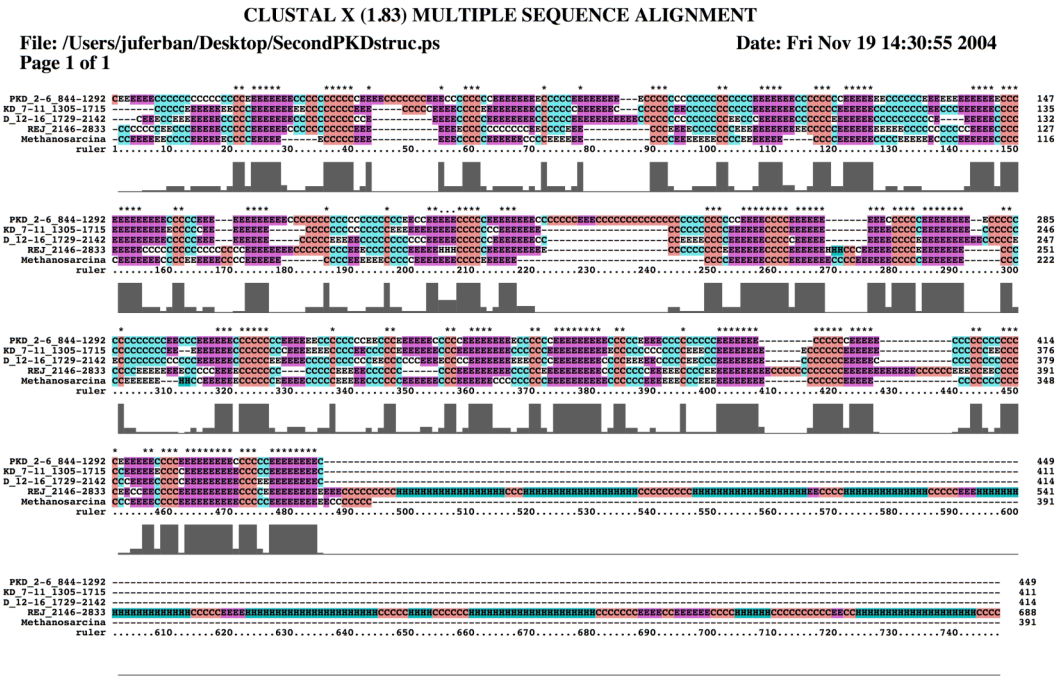


Figure 36: Secondary structure multiple alignment result. The grey boxes beneath the alignment show the level of homology. As it can be seem the pink boxes corresponding to the ß-sheets are the regions showing a higher level of homology.

FINAL REMARKS

- Sequenciation of the complete PKD1 gene is necessary to identify those changes related to the disease. The high number of changes presented along the gene made it impossible to determine the relation between nucleotide change and development of the disease. This relation could only be established for the changes classified as non sense amino acid changes, as these are stop codon or frameshift changes that affect the amino acid composition of the protein in a long extend.
- PKD1 gene is a flexible gene that allows for different SNPs without affecting protein function. By analyzing the content of the database we were able to identify a high number of silent nucleotide changes and missense amino acid changes within the gene. As some of this changes were also observed in healthy individuals it gave us a clue about the flexibility of the protein. This was also supported by the fact that polycystin 1 does not act as an enzyme where more conservation of the amino acid sequence is expected.
- Automatic analysis tools shown to be an efficient method for annotation of changes within the gene. The use of our own developed annotation tools and database aided in the fast analysis of individual gene sequences and in the organization and integration of the data.
- Integration of genotype and phenotype information into a database demonstrates to be a useful method to study the behavior of PKD1. Using the information contained in our database we look for different relations between sequence change and disease behavior and even though our preliminary results did not show any kind of relation, it is expected that we will be able to find some relations as soon as the information of genotype and phenotype increases in size and quality.
- From structural studies it was possible to propose the function of polycystin 1. We demonstrate that the use of “ab initio” structural models can be useful for the understanding on how the protein behaves and, despite the models were not accurate enough for deeper

studies, they were good to find similar structures of proteins with known function confirming in some cases what it was believed to be the function of polycystin 1 and pointing to new directions in other cases.

- The extracellular part of the protein plays a role in cell adhesion. As we have shown, the folding of the different domains that are located at the extracellular part of the protein reminds those of proteins with function in cell adhesion, protein binding and in signal receptors. The presence of 7 bladed beta-propellers made us think that this structures would interact with other 7 bladed beta-propeller at neighboring cells when polycystin 1 was placed in the basolateral membrane aiding in the maintenance of duct structure. When placed in the cilia we believe that the main role of the extracellular part of the protein is as a mechanosensor that controls the cation channel activity in the presence of urine currents.
- The intracellular part of the protein present lipoxigenase activity. This function was already described and our structural models just confirm what it was known for PLAT domain. This result also support the conclusions obtained by our approach as this could be considered as the test set of our tertiary structure study.
- The transmembrane domains of the protein form a cation channel. There are strong evidences that transmembrane domains of polycystin 1 are part of a non specific cation channel, the three dimensional structure, the conservation of some of its domains and the sequence similarity with polycystin 2 (a known cation channel) among others lead us to define this new function for polycystin 1. As shown, our results support those by Babich et al.
- The study of the Beta-propeller models allowed us to define a new structural domain (the Beta-propeller-PKD) composed by the 15 PKD domains that lay together and by the first half of REJ. It was also useful to propose a new subdivision of REJ domain in two parts, one composed by beta-sheets that form the fourth Beta propeller and a PKD fold structure, and a second part of REJ mainly composed by Alpha-helix.

- Taking a broader look at all the functions of the protein we can speculate on how the protein works. We suggest that during kidney development intra and extracelular part of the protein act as signal receptors that regulate the activity of the cation channel, controlling the Ca +2 concentration within the cell and controlling by this method the expression or repression of certain genes involved in cell duplication and specificity and that's why when a mutation occurs in the protein, cells keep dividing leading to the formation of cysts. In a mature cell we suggest that the role of the protein has also implications in the maintenance of the cell to cell interactions and also regulating gene expression.

CONCLUSIONES

1. La secuenciación completa del gen PKD1 es necesaria para poder identificar aquellos cambios en el ADN relacionados con el desarrollo de la enfermedad.

2. PKD1 se muestra como un gen flexible ya que presenta un elevado número de polimorfimos sin afectar para ello a la funcionalidad de la proteína.

3. Las herramientas de análisis automatizado se han mostrado como un método eficaz para la anotación de los múltiples cambios en la secuencia del gen.

4. La integración de la información genotípica y fenotípica en una base de datos fue de gran utilidad para la clínica y la investigación en la Poliquistosis Renal.

5. El empleo de un sistema gráfico para el estudio de las mutaciones permite la identificación de cambios presentes en multiples indíviduos de una manera rápida y eficaz.

6. Se describe la variante de secuencia 21239-gccac-21243 presente en el 80% de los individuos estudiados.

7. Se demostró que el uso de modelos estructurales de la poliquistina 1 era útil para el estudio presuntivo de la función proteica.

8. Se presentan pruebas estructurales de la función de la poliquistina 1 en la adhesión celular.

9. Se define un nuevo tipo de dominio estructural, llamado Beta hélice-PKD o “Beta-propeller-PKD”, y se postula una reorganización del modelo hasta ahora existente en la poliquistina 1.

10. Se describen tres agrupaciones o “clusters” de 5 dominios PKD cada una los cuales presenta una alta homología a nivel de estructura secundaria.

11. La estructura secundaria del dominio REJ y la alta homología de parte de ésta con los “cluster” de PKD, permiten dividir a REJ en dos subdominios diferentes; el primero de ellos sería también constituyente de una estructura “Beta-propeller-PKD”.

12. Los dominios transmembrana 6-11 probablemente forman parte de un canal catiónico no específico.

13. La presencia de 4 “beta-propellers” y de un canal catiónico en una misma proteína implicaría la interacción de ambos como mecanismo regulador del estado del poro.

BIBLIOGRAFÍA

1. Altschuh,D., Vernet,T., Berti,P., Moras,D. And Nagai,K.: Coordinated amino acid changes in homologous protein families. Protein Eng., 2, 193-199, 1988.
2. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. And Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol., 215, 403-410.
3. Ariceta G., Vila M., Arrojo L., Otero M., Pazos G., Alonso R., Cordal T., Davila S., Lens X.M.: Genetic Diagnosis of Autosomal Dominant Polycystic Kidney Disease in Children at Risk. 33th Annual Meeting of the European Pediatric Nephrology Association. Prague 9:370ª, 1999.
4. Babich, V.; Zeng, W.; Yeh, B.; Ibraghimov-Beskrovnaya, O.; Cai, Y.; Somlo, S.; Huang, C. : The N-Terminal Extracellular Domain is required for Polycystin-1-dependent channel activity. J. Biol. Chem. 2004, 279,24, June 11, 25582-25589.
5. Badenas C, Praga M, Armengol A, Tazón B, Andrés A, Morales E, Camacho Ja, Lens Xm, Davila S, Milà M, Darnell A, Torra R. Mutations in the COL4a4 and COL4a3 genes cause Familial Benign Hematuria. J Am Soc Nephrol 2002; 13: 1248-1254
6. Bernstein,F.C., Koetzle,T.F., Williams,G.J.B., Meyer,E.F., Brice,M.D. et al: The protein Data Bank: a computer based archival file for macromolecular structures. J. Mol. Biol., 112, 535-542. 1977
7. Beyer,W.A., Stein,M.L.; Smith,T.F. and Ulam,S.M. : A molecular sequence metric and evolutionary trees. Math. Biosci., 112, 535-542. 1977.
8. Bilofsky,H.S.; Burks,C., Fickett,J.W., Goad,W.B., Lewitter,F.I., Rindone,W.P., Swindell,C.D. and Tung,C.S.: The GenBank genetic sequence data bank. Nucleic Acids Res., 14,1-4. 1986
9. Bogdanova N, Markoff A, Gerke V, McCluskey M, Horst J, Dworniczak B.: Homologues to the first gene for autosomal dominant polycystic kidney disease are pseudogenes. Genomics. 2001 Jun 15;74(3):333-41.
10. Bonneau R, Strauss CE, Rohl CA, Chivian D, Bradley P, Malmstrom L,

Robertson T, Baker D. (2002) De novo prediction of three-dimensional structures for major protein families. J Mol Biol 322(1):65-78

11. Bonneau R, Tsai J, Ruczinski I, Chivian D, Rohl C, Strauss CE, Baker D. (2001) Rosetta in CASP4: progress in ab initio protein structure prediction. Proteins Suppl 5:119-26
12. Bycroft M, Bateman A, Clarke J, Hamill SJ, Sandford R, Thomas RL, Chothia C.: The structure of a PKD domain from polycystin-1: implications for polycystic kidney disease. EMBO J. 1999 Jan 15;18(2):297-305.
13. Chivian D, Kim DE, Malmstrom L, Bradley P, Robertson T, Murphy P, Strauss CEM, Bonneau R, Rohl CA, Baker D. (2003) Automated prediction of CASP-5 structures using the Robetta server. Proteins 53 Suppl 6:524-33
14. Chothia,C. And Lesk,A.M.: The relation between the divergence of sequence and structure in proteins. EMBO J., 5, 823-826. 1986.
15. Clarke,B. :Selective constraints on amino-acid substitution during the evolution of proteins. Nature, 228, 159-160. 1970
16. Cordal T., Davila S., Alonso R., Vila M., Ariceta G., Otero M., Pazos G., Arrojo L., Lens X.M.: Poliquistosis Renal de Adulto como forma predominante en la infancia. XXIX Congreso Nacional de la Sociedad Española de Nefrología. Valencia 17-20 Octubre, 1999.
17. Corpet, F.: Multiple sequence alignment with hierarchical clustering, Nucl. Acids Res., 16 (22), 10881-10890. 1988
18. Daoust M.C., Reynolds D.M., Bichet D.G., Somlo S.: Evidence for a third genetic locus for autosomal dominant polycystic kidney disease. Genomics 25:733-736, 1995.
19. Dayhoff,M.O.: Atlas of Protein Sequence and Structure, Vol. 4, Suppl. 3. National Biomedical Research Foundation, Washington, D.C., U.S.A. 1978
20. Devereux,J., Haeberli,P. And Smithies,O.: A comprehensive set of sequence analysis programs for the VAX. Nucleic Acids Res., 12, 387-395. 1984.
21. Devuyst O., A. Persu, M.S. Stoenoiu, Y. Pirson, Lens Xm, D. Chauveau. eNOS polymorphism and renal disease progression in

Autosomal Dominant Polycystic Kidney Disease. Am J Kidney Dis. 2003; 41: 1124-1125

22. Devuyst O, Persu A, Vo-Cong MT: Autosomal dominant kidney disease: modifier genes and endothelial dysfunction. . Nephrol Dial Transplant 18, 2211-2215. 2003
23. Epstein,C.J. :Non-randomness of amino-acid changes in the evolution of homologous proteins. Nature, 215, 355-359. 1967.
24. Fülöp, V., Szeltner, Z. and Polgár, L. (2000). Catalysis of serine oligopeptidases is controlled by a gating filter mechanism. EMBO. Rep. 1, 277-81
25. G. Vriend, C. Sander. Quality control of protein models: Directional atomic contact analysis. J.Appl.Cryst. 26, 47-60. 1993
26. Geng L., Segal Y., Pavlova A., Barros E.J.G., Lohning C., Lu W.N., Nigam S.K., Frischauf A.M., Reeders S.T., Zhou J.: Distribution and developmentally regulated expression of murine polycystin. American Journal of physiology-Renal physiology 41:F451-F459,1997.
27. Geng L., Segal Y., Peissel B., Deng N.H., Pei Y., Carone F., Rennke H.G., Gluksmannkuis A.M., Schneider M.C., Areicsson M., Reeders S.T., Zhou J.: Identification and localization of polycystin, the PKD1 gene product. J Clin Invest 98:2674-2682,1996.
28. Gibbs, A.J. and McIntyre,G.A.: The diagram, a method for comparing sequences. Eur. J. Biochem.,16, 1-11. 1970.
29. Gibbs,A.J.; Dale,M.B., Kinns,H.R. and MacKenzie,H.G.: The transition matrix method for comparing sequences; its use in describing and classifying proteins by their amino acid sequences. Syst. Zool., 20, 417-425. 1971.
30. Gilbert-Barness E.F., Opitz J.M., Barness La: Inheritance of kidney and urinary tract diseases. Ed. Spitzer A, Avner ED, pp 327-400. Kluwer Academic publisher, Boston, 1990.
31. Gingeras,T.R. and Roberts,R.J.: Steps toward computer analysis of nucleotide sequences. Science, 209, 1322-1328. 1980.

32. Gonzalo A., Gallego A., Tato A., Ortuno J.: Age at renal replacement therapy in autosomal dominant polycystic kidney disease. Nephron 74:620,1996.
33. Hamm,G.H. and Cameron,G.N.: The EMBL Data Library. Nucleic Acids Res. 14,5-9. 1986.
34. Higgins,D.G. and Sharp,P.M.CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. Gene 73,237-244. 1988.
35. Hogewind B.I., Veltkamp J.J., Koch C.W., De Graef J.: Genetic counselling for adult polycystic kidney disease. Ultrasound a useful tool in pre-symptomatic diagnosis?. Clin Genet 18:168-172,1980.
36. Hughes J., Ward C.J., Peral B., Aspinwall R., Clark K., Sanmillan J.I., Gamble V., Harris P.C.: The polycystic kidney disease 1 (PKD1) encodes a novel protein with multiple cell recognition domains. Nature Genet. 10:151-159, 1995.
37. Ikeda M, Guggino WB. Do polycystins function as cation channels?. Curr Opin Nephrol Hypertens. 2002 Sep;11(5):539-45.
38. Kim DE*, Chivian D*, Baker D. (2004) Protein structure prediction and analysis using the Robetta server. Nucleic Acids Res. 32 Suppl 2:W526-31
39. Kimberling W.J., Kumar S., Gabow P.A., Kenyon J.B., Connolly C.J., Somlo S.: Autosomal dominant polycystic kidney disease: localization of the second gene to chromosome 4q13-q23. Genomics 18:467-472,1993.
40. Krzywicki,A. And Slonimski,P.P.: Formal analisys of protein sequences: I. Specific long-range constraints in pair associations of amino acids. J. Theor. Biol. 17,136-158. 1967.
41. Kumánovics A, Levin G, Blount P.: Family ties of gated pores: evolution of the sensor module. The FASEB Journal. 16: 1623-1629. October 2002.
42. Laskowski, R.A., MacArthur, M.W., Moss, D.S. and Thornton, J.M: PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.* **26**, 283-291. 1993

43. Lens X.M.: Rexistro galego de nefropatías. Epidemiología de la Enfermedad Renal Poliquística Autosómica Dominante en Galicia. Impacto del tratamiento substitutivo. Nefrología 13(Supl.3): 178-180, 1993.
44. Lens Xm, Onuchic L, Wu G, Hayashi T, Daoust M, Mo-Chizuki T, Santarina Lb, Stockwin Jm, Mücher G, Becker J, Sweeny We, Avner De, Guay-Woodford, Zerres K, Somlo S, Germino Gg. An Integrated Genetic and Physical Map of the Autosomal Recessive Polycystic Kidney Disease Region. Genomics 1997; 41: 463- 466
45. Marchler-Bauer A, Anderson JB, DeWeese-Scott C, Fedorova ND, Geer LY, He S, Hurwitz DI, Jackson JD, Jacobs AR, Lanczycki CJ, Liebert CA, Liu C, Madej T, Marchler GH, Mazumder R, Nikolskaya AN, Panchenko AR, Rao BS, Shoemaker BA, Simonyan V, Song JS, Thiessen PA, Vasudevan S, Wang Y, Yamashita RA, Yin JJ, and Bryant SH: "*CDD: a curated Entrez database of conserved domain alignments*", Nucleic Acids Res. 31:383-387. 2003.
46. Moy GW, Mendoza LM, Schulz JR, Swanson WJ, Glabe CG, Vacquier VD.: The sea urchin sperm receptor for egg jelly is a modular protein with extensive homology to the human polycystic kidney disease protein, PKD1. J Cell Biol. 1996 May;133(4):809-17.
47. Needleman,S.B. and Wusch,C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol., 48, 443-453. 1970.
48. Oxana Ibraghimov-Beskrovnaya, William R. Dackowski, Lukas Foggensteiner†,Nick Coleman, Sathia Thiru, Linda R. Petry, Timothy C. Burn, Timothy D. Connors, Terence Van Raay, John Bradley, Feng Qian§,Luiz F. Onuchic, Terry J. Watnick, Klaus Piontek, Raymond M. Hakim, Gregory M. Landes, Gregory G. Germino, Richard Sandford, And Katherine W. Klinger: Polycystin: In vitro synthesis, in vivo tissue expression, and subcellular localization identifies a large membrane-associated protein. Proc. Natl. Acad. Sci. USA Vol. 94, pp. 6397–6402, June 1997.
49. Persu A, M.S. Stoenoiu, T. Messiaen, S. Davila, J-C. Robino, O. El Khattabi, D. Mourad M, Horie S, Pirson Y,Chauveau, X. Jeunemaitre,

Lens Xm, O. Devuyst. Modifier effect of eNOS in Autosomal Dominant Polycystic Kidney Disease. Hum Mol Gen. 2002; 11: 229-241

50. Persu A, Duyme M, Pirson Y, Lens Xm, Messiaen T, Breuning Mh, Chauveau D, Levy M, Grünfeld J-P, Devuyst O. Comparison between siblings and monozygotic twins supports a significant role of modifier genes in Autosomal Dominant Polycystic Kidney Disease. Kidney Int 2004; 66: 2132-2136.
51. Peters D.J., Sandkuijl L.A.: Genetic heterogeneity of polycystic kidney disease in Europe. Contrib Nephrol 97:128-139,1992.
52. Peters D.J., Spruit L., Saris J.J., Ravine D., Sandkuijl L.A., Fossdal R., Boersma J., Van Eijk R., Norby S., Constantinou-Deltas Cd., y Col.: Chromosome 4 localization of a second gene for autosomal dominant polycystic kidney disease. Nat Genet. 5:359-362,1993.
53. Pirson Y: Physiopathology of diabetic nephropathy: what we learn from transplantation. Nephrologie. 1998;19(3):105-9.
54. Ponting CP, Hofmann K, Bork P.: A latrophilin/CL-1-like GPS domain in polycystin-1. Curr Biol. 1999 Aug 26;9(16):R585-8.
55. Ramachandran, G.N., Ramakrishnan, C. and Sasisekharan, V.J.: Stereochemistry of polypeptide chain configurations, J. Mol. Biol. 7, 95-99. 1963
56. Rashin,A.A.: Locations of domains in globular proteins. Nature, 291,85-86. 1981.
57. Ravine D., Gibson R.N., Walker R.G., Sheffield L.J., Kincaid-Smith P., Danks D.M.: Evaluation of ultrasonographic diagnosis criteria for autosomal dominant polycystic kidney disease a. Lancet 343:824-827, 1994.
58. Ravine D., Walker R.G., Gibson R.N., Forrest S.M., Richards R.I., Friend K., Sheeffield L.J., Kincaid-Smith P., Danks D.M.: Phenotype and genotype heterogeneity in autosomal dominant polycystic kidney disease: Lancet 340:1330-1333, 1992.
59. Reeders S.T., Breuning M.H., Davies K.E., Nicholls R.D., Jarman A.P., Higgs D.R., Pearson P.I., Weatherall D.J.: A highly polymorphic DNA marker linked to adult polycystic kidney disease on chromosome 16. Nature 317:542-544, 1985.

60. Reynolds DM, Hayashi T, Cai Y, Veldhuisen B, Watnick T, Lens XM, Mochizuki T, Quian F, Fossdal R, Coto E, Wu G, Breuning NH, Germino GG, Peters D, Somlo S. Aberrant splicing in the PKD2 gene as a Cause of Polycystic Kidney Disease. J Am Soc Nephrol 1999; 10: 2342- 2351.
61. Rezende W., Parreira Ks, García-González M.A., Riveira E, Banet JF, Lens X.M. Homozygosity for uromodulin disorders (Familial Juvenil Hyperuricemic Nephropathy and Autosomal Dominant Medullary Cystic Kidney Disease-type 2). Kidney Int 2004; 66: 558-563
62. Rezende W., Parreira Ks, Banet JF, Outeda P, Barrio-Lucia V., Lens X.M. A novel pattern of mutation in Uromodulin disorders. (enviado para publicación)
63. Rossmann,M.G. and argos,P.: Three-dimensional coordinates from stereo diagrams of molecular structures. Acta Crystallogr., 36, 819-823. 1980.
64. Sackin,M.J.: Cross association: a method of comparing protein sequences. Biochem. Genet., 5,287-313. 1971.
65. Sayle,R.A.; and Milner-White,E.J.: RASMOL: biomolecular graphichs for all. Trends Biochem. Sci., 20, 374-374. 1995.
66. Shatsky M, Nussinov R, Wolfson HJ.: FlexProt: alignment of flexible protein structures without a predefinition of hinge regions. J Comput Biol. 2004;11(1):83-106.
67. Shatsky M, Nussinov R, Wolfson HJ.: FlexProt: alignment of flexible protein structures without a predefinition of hinge regions. J. Comput. Biol. 2004; 11(1):83-106.
68. Simons KT, Kooperberg C, Huang E, Baker D.: Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol. 1997 Apr 25;268(1):209-25.
69. Simons KT, Kooperberg C, Huang E, Baker, D. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulate anealing and Bayesian scoring functions. J Mol Biol 268:209-25
70. Simons KT, Ruczinski I, Kooperberg C, Fox B, Bystroff C, Baker D.

(1999) Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. Proteins 34(1) 82-95

71. Smith,T.F. and Waterman,M.S.: Comparison of biosequences. Adv. Appl. Math., 2, 482-489. 1981.
72. Stayner, C. and J. Zhou. Polycystin channels and kidney disease Trends in Pharmacological Sciences. 22:543-546. 2001.
73. THE INTERNATIONAL POLYCYSTIC KIDNEY DISEASE CONSORTIUM: Polycystic kidney disease: the complete structure of the PKD1 gene and its protein. Cell 81:289-298,1995.
74. Tinoco,I.; Uhlenbeck,O.C. and Levine,M.D.: Estimation of secondary structure in ribonucleic acids. Nature, 230, 362-367. 1971.
75. Thompson,J.D., Gibson,T.J., Plewniak,F., Jeanmougin,F. and Higgins, D.G.: The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Research, 24:4876-4882. 1997
76. Torra R., Badenas C., Darnell A., Nicolau C., Volpini V., Revert L., Estivill X.: Linkage, clinical features, and prognosis of autosomal dominant polycystic kidney disease types 1 and 2. J. Amer. Soc. Nephrol 7:2142-2151,1996.
77. Torres VE.: Extrarenal manifestations of autosomal dominant polycystic kidney disease. Am J Kidney Dis. 1999 Dec;34(6):xlv-xlviii
78. Tsiokas L, Kim E, Arnould T, Sukhatme VP, Walz G.: Homo- and heterodimeric interactions between the gene products of PKD1 and PKD2. Proc Natl Acad Sci U S A. 1997 Jun 24;94(13):6965-70.
79. Vega BT, Badenas C, Ars E, Lens XM, Milà M, Darnell A, Torra A. Autosomal recessive Alport's syndrome and benign familial hematuria are collagen type IV diseases. Am J Kidney Dis. 2003 Nov; 42(5): 952-9
80. Viribay M, Hayashi T, Telleria D, Mochizuki T, Reynolds DM, , Alonso R, Lens XM, Moreno F, Harris P, Somlo S, San Millan JL. Novel stop and frameshifting mutations in the Autosomal Dominant Polycystic Kidney Disease 2 (PKD2) Gene. Hum Gen 1997; 101: 229- 234.

81. Vriend, G. : WHAT IF: a molecular modeling and drug design program. *J. Mol. Graphics* 8, 52-56. 1990
82. Watnick TJ, Piontek KB, Cordal TM, Weber H, Gandolph MA, Qian F, Lens XM**,** Neumann HPH, Germino GG. An unusual pattern of mutation in the replicated portion of PKD1 is revealed by use of a novel strategy for mutation detection. Hum Mol Gen 1997; 6:1473-1481
83. Weston BS, Bagneris C, Price RG, Stirling JL.: The polycystin-1 C-type lectin domain binds carbohydrate in a calcium-dependent manner, and interacts with extracellular matrix proteins in vitro. Biochim Biophys Acta. 2001 May 31;1536(2-3):161-76.
84. Wu,T.T.; Fitch,W.M. and Margoliash,E.: The information content of protein amino acid sequences. Ann. Rev. Biochem., 43, 539-566. 1974

WEB REFERENCES

Crystallography: http://www-structure.llnl.gov/Xray/index_intro.html

NMR: http://www.cryst.bbk.ac.uk/PPS2/projects/shirra/html/intro.htm

NMR: http://www.cis.rit.edu/htbooks/nmr/chap-1/chap-1.htm

Modelling: http://www.expasy.org/swissmod/course/course-index.htm

PERL: http://perl.com

Bio-PERL: http://www.bioperl.org

BLAST: ftp://ftp.ncbi.nih.gov/blast/

EMBOSS: http://www.hgmp.mrc.ac.uk/Software/EMBOSS/

MySQL: http://www.mysql.com

PHP: http://www.php.net

JAVA: http://java.sun.com/

NNPREDICT: http://www.cmpharm.ucsf.edu/ñomi/nnpredict.html

PROF: http://www.aber.ac.uk/~phiwww/prof/

Sspro: http://www.igb.uci.edu/tools/scratch/

PHDsec: http://cubic.bioc.columbia.edu/predictprotein

Consensus: http://www.bork.embl-heidelberg.de/Alignment/consensus.html

Rosetta: http://www.bioinfo.rpi.edu/~bystrc/hmmstr/server.php

Robetta: http://robetta.bakerlab.org/

Biotech: http://biotech.ebi.ac.uk:8400/

PDB: http://www.rcsb.org/pdb/

DALI: http://www.ebi.ac.uk/dali/

FlexProt: http://bioinfo3d.cs.tau.ac.il/FlexProt/

Multalin: http://prodes.toulouse.inra.fr/multalin/multalin.html

APPENDIX

Consensus Secondary Structure Sequence of Polycystin 1

CCCCCCHHHHHHHHHHHHHHHHHCCCCCCCCCCCCCCCCCCCCCCCE
EECCCCCCCCCCCCCCCCCCCCCHCCCCHHHHEHHHHHHHHHHHHHE
ECCCCCCCCCCHHHHHHCHCCCCEEECCCCCCCCCCCCCHCHHHHHHHCC
CECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCECCCCCCCCCCCE
EEEEHHHHCCCCCCCCCCCCCCCCCCCCCCHHCHHCCCEECCCCCCCCCC
CCEEEECCCCCCCCCCCCCCCCCCCCCEEECCCCCCCCCEECCCCCCCCCC
CHHHHCCCCCCCCCCCCECCCCCCCCEECCCCCCCCCCCECCCCCCEEEEE
EEHHCCCEECCCCCEEECCCHHEEECCCCCCCCCCCCCCCCEEECCCCCC
CHHHHHHHEHCCCCCCCCCCCCCCCCCCCECCCCCCCEEEEEHHHHHHHHH
HHHHHHHHHHHHHHHCCCHHHHHHHHEEHHCCCCEEEEEECCCCCCCCCC
CCCCCCCCHHCCCCCCCCCCCCCCCCCEEEECCCCCCCCCCCCCCCCCCE
EEEECCCCCCCCCCHHEEECCCCCCCCCCCCCCCCCCCCCCCCCCCCEEE
ECCCCCCCCHHHHHEHHHCCCCCCCCCCHHHHHHHHHHHHCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCECCCCCCCCCCCCCCCCCCC
CCCCCCCHHHHHHHHHHECCCCCCCEEEEEECCCCCCCECCCCCEEEECC
CCCCCEEECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCEEEH
HHHHHHHHHHHEEECCCCCCCCCCCCCCEEEEEECCCCCCCCCCCEEEEE
CCCCCCCEEEECCCCCCCEEECCCCCCEEEEECCCCCCCCCCCCCCCC
CCCCECCCCCCEEECCCCCCCCCCCCCCCEEEEEECCCCCCEEEEEEEE
ECCCCCCCEEEECCCCCCCECCCCCCCCCHHEEEEEEECCCCECCCCCC
EEEECCCCCCCCCEEEEEEEEECCCEEEEEEECCCCCCEEEEEEEEE
EEHCCCCCEECCCCCEECCCCHHHHHECEEECHHHEEEEECCCC
HEEECCCCCCCCCCCCCCCCEEEEECCCCCECCCCCEEEEEECCC
CHEECCCEEEECCCCCEEEECCCEEECCCEEECCCCCCCCCEEEE
EECCCCCCCCCCCCCCCCCECCCCEEEEEEECCCCCCCHEEEEEE
EEHCCCCCCCEEECCCCCEEEEEECCCCCCEEEECCCCCCEECCCC
EEEEEEECCCCEEEECCCCCHHEEEEEEEEEECCCCCCCCCC
CCCEEEEECCCCCCEEECCCCCCCCCCCCEECCCCECEECCCCCEEE
EEEECCCCCECEEEECCCCCCCCCCCCCCCEEEEHCCCHHHECCCC
CCCCEEECCCCCCCCCCCCCCCEEEEECCCCEEEEEECCCCCC
CCCEEEECCCEEEEEECCCCCCECCCCEEECCCCCCCCEEECCC
CCCCCCCCCEEECCCCCEEEEECCCCCCCCCCEEEEEECCEEE
ECCCCEEECCCEEEEEECCCCCEEEEECCCCCCCCCCEEEEE
ECCCEEEEEEECCCCCCCCEEEEEEEEHHCHEECCCCCCCCCCE
EEEEECCCCCCEEEEECCCCCCCCCCCCCCCEEEECCCEEEEEE
ECCCCCCEEECCCCCCCCCEEECCCCCCCCCCEEEEEECCCCE
EEEEEECCCCCCCCCCCCECCCCCCCCEEEECCCCCCCCCEEEE
EECCCCEEECCCCCCCEEECCCCCCECCECCCCCCCEEECCCC
CCCCEEEECCCCCEEEECCCCCCCEEEEEEEEECCCCCEEEEEEC
CCCCCCCEEEEEEHCCCEEEEEECCCCCCCCCCCCCCCCCCC
CEEEECCCCCCCCCCHHEEEEEECCCCCCCCCCCCCCCCCCEE
EECCCCHEEEEECCEECCCCEEECCCCCEECCCCCEEEEEE
ECCCCCCCEEEEEECCHHEEECCCCCCCCEEEECCCCCCEEE
EECCCCCCCCCCCCCCCCCCCCCCCCCEEEEECCCCEEEEEEE
EEECCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
EEEECCCCCCCCCEECCHHHHHHHCCCHHEEECCCCCCECCEEEE
EECCCCCCCCCCCEECCCCCCCCCCCCCCCCCCCCCCCEEEEEEECC

CCCCCCEEEEEEECCCCCCCEEECCCCCEEECCCCCEEECCCCCCCCC
CCCCCCCCCCEEEEECCCCCCCCCCEECCCCCCCCCECCCHHHHCCCC
EEEEEEEEEECCCCCCCCCEEEEECCCCEEEEEEECCCCCEEEEECC
EEEEEECEEECCCCCCCCCEEEEEECCCEEEEECCCCCCCCCHHEEEE
ECCEECCCCCCEEEEEECCCCCCCCCEEEECCCCCCCCCCCCCECCCC
CEEEEEEEEEECCCCCCCCCCCCHHHHHHHHHHCCCCCCCEEEEECC
CCCCCCCCCCCCCCCEECEEEEECCCCHHHHHHHHCECCECCCCCC
CCCCEEEEECCCCCCHHHHCCCCCCCCEEHHHHHHHHHHHHHHHHH
CCCCCCCCHHHHHHHHHHHHHEEEEECCCCCHHHHHHHHHHCCCC
CCEEHHHHHHHHHHHHHHHHHHHHCCCCCCCCCCCCCCEEEECCCE
EEECCCCCCCCCCCCCCCCCCHHHHHHHHHHHHHHHHHHHCCCCC
CCCCCCHHHHHHCCCCCCCCEEEEECCCCCCCCCCCCCHHHHHHHCC
HHHHEEEEECCCCCCCCCEEECCEECHHHHHHHHHCCCCCCCHHHCC
CCCCEEEECCCCCCCCCCCCCCCCCCCCCEEECCCCCCEEEEECCCC
CCCCEEEEEEEECCCCCCCCCCCCEEEEEECCCCCCCCCCCCCCCCC
CCCCCCCCCCCEEEECCCCCCCCCCEEEECCCCCEEEEEEECCCEEE
EHHHCCCCCEEHHCCCCCCCCCCCCCEEEEEEHHHHCCCCECCCCC
EEEECCCCCCCCEEEEEEHHHHHHHHHHHHHHHHHHHCCCCCCCCCC
CCCCCEEEEEEEECCCCCCCCCCEEEEEEEECCCCCCCCCCCCCCC
CCCCCEEEEECCCCCCCCCEEEEEEECCCCCCCCCHEEEEEEECCCC
CCEEEEHCCEEEECECCCCCEEEEEEHHCHHHHHHHHHHHHHHHHCC
CCCEEEEECCCCCCCCCCCCCCCCHHHHHHEEHHCCCCEEEECCCCC
CCCCCCCCCCCCCCCCEEEEEEEEEEEEHHHHHHHHHHCCCCCCCC
CCCCCCCCCEEEECCCCCCCCCCCCEEEECCCCCCHHHHHHCCCCCEC
CCCCEEECCCCCCCCCCCCCCCCCCCCCCHHHHHHHCCCCCCCCCC
CCCCCCCCCCCCCCCCCCHHHHHHHHHCCCCCCCCCCCCCCHHHH
HCCCCCCCCCHHHHHHHHHCCCCCCCCCCCCCCCHHHHHCCCCHHHC
HHHCCCCHHHHCCCCCHHHEEEEEEEEEEEEECCCCCCCEEEEEEHCC
HHHHHHCCCCCCHHHHHHHHHHHHHHCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
CCCCCCCCCCHHHHHHHHHHHCCCHHHHHHCCCCCCHHHHHHCCCEE
CCCCCCCCCCCCCCHEEEEECCCCCCCCCCCCCEECCCCCCCCCCCCC
CCCCCCCCCCCEEEECCCCCCCCCCCCEEEECCCCCEEECCCCHHHHH
HHHHHHHHCCCCCCCCHEEEEEEECCCCCCCCHEEEEEEECCCCCCHH
CHHECCEEEEEEHCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHCCCH
HHHHHCHHHHHHHHHHHHHHHHHHHHHCCCHHHHHHHHCCCCCCCC
CCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCHHHHH
HHHHHHHCEEEEEEHHHHHHHHHEEECCCCCCHHHHHHHHEEECCCCC
CCCCCCCCCCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCC
CCCCCCCCCHHHHHHHHHHHHHHHCCCCHHHHHEEEEECCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCEEECCCCCCCCCCC
HHHHHHHHHHHHHHHHCCCCHHHHHHHHHHHHCCCCCCCCCCCCCC
CCCCCCCCCCCHHHHHHCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CorrectSeq.java

```
package correctseq;

/**
 * <p>Title: CorrectSeq </p>
 * <p>Description: Programa para corregir secuencia en formato fasta de una
 * manera rapida y eficaz</p>
 * <p>Copyright: Copyright (c) 2003</p>
 * <p>Company: Julio Fernandez Banet</p>
 * @author Julio Fernandez Banet
 * @version 1.0
 */

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;

public class CorrectSeq extends Frame{
  FileDialog fd= new FileDialog(this);
  MenuBar menu1 = new MenuBar();
  Menu mnArchivo = new Menu();
  MenuItem mnAbrir = new MenuItem();
  MenuItem mnCerrar = new MenuItem();
  MenuItem mnGuardar = new MenuItem();
  Panel panel2 = new Panel();
  Panel panel3 = new Panel();
  GridLayout gridLayout1 = new GridLayout();
  Label Posicion = new Label();
  TextField textPosicion = new TextField();
  Label Cambio = new Label();
  TextField textCambio = new TextField();
  TextArea textSecuencia = new TextArea();
  Button btEdit = new Button();
  Label Nombre = new Label();
  TextField textnombre = new TextField();
  Hashtable datos=new Hashtable();
  Label Inicio = new Label();
  TextField txtInicio = new TextField();
  Label Fin = new Label();
  TextField txtFin = new TextField();

  public static void main(String[] args){
    CorrectSeq cs= new CorrectSeq();
    cs.setSize(600,400);
    cs.setResizable(false);
    cs.show();
  }
```

```
public CorrectSeq(){
  enableEvents(AWTEvent.WINDOW_EVENT_MASK);
  try{
    jbInit();
  }
  catch(Exception e){
    e.printStackTrace();
  }
}
public void processWindowEvent(WindowEvent e){
  super.processWindowEvent(e);
  if(e.getID()==WindowEvent.WINDOW_CLOSING){
    mnCerrar_actionPerformed(null);
  }
}
private void jbInit() throws Exception{
  this.setMenuBar(menu1);
  mnAbrir.setLabel("Abrir");
  mnAbrir.setShortcut(new MenuShortcut(79));
  mnAbrir.addActionListener(new java.awt.event.ActionListener(){
    public void actionPerformed(ActionEvent e){
      mnAbrir_actionPerformed(e);
    }
  });
  mnCerrar.setLabel("Cerrar");
  mnCerrar.setShortcut(new MenuShortcut (83));
  mnCerrar.addActionListener(new java.awt.event.ActionListener(){
    public void actionPerformed(ActionEvent e){
      mnCerrar_actionPerformed(e);
    }
  });
  mnGuardar.setLabel("Guardar");
  mnGuardar.setShortcut(new MenuShortcut (KeyEvent.VK_S, true));
  mnGuardar.addActionListener(new java.awt.event.ActionListener(){
    public void actionPerformed(ActionEvent e){
      mnGuardar_actionPerformed(e);
    }
  });
  mnArchivo.setLabel("Archivo");
  menu1.setHelpMenu(mnArchivo);
  panel2.setLayout(gridLayout1);
  Posicion.setAlignment(1);
  Posicion.setBackground(Color.orange);
  Posicion.setText("Posicion");
  textPosicion.setText("Indica la posicion a cambiar");
  Cambio.setAlignment(1);
  Cambio.setBackground(Color.orange);
  Cambio.setText("Cambio");
  textCambio.setText("Indica el cambio a realizar");
  panel2.setFont(new java.awt.Font("DialogInput", 0, 12));
```

```
    gridLayout1.setColumns(2);
    gridLayout1.setRows(5);
    textSecuencia.setColumns(50);
    textSecuencia.setRows(10);
    textSecuencia.setText("Secuencia a modificar");
    btEdit.setLabel("Editar");
    btEdit.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(ActionEvent e) {
        btEdit_actionPerformed(e);
      }
    });
    Nombre.setAlignment(1);
    Nombre.setBackground(Color.yellow);
    Nombre.setText("Nombre Secuencia");
    textnombre.setText("Introduce aqui el nombre de tu secuencia");
    Inicio.setAlignment(1);
    Inicio.setBackground(new Color(255, 205, 205));
    Inicio.setComponentOrientation(null);
    Inicio.setText("Primera base a guardar");
    Fin.setAlignment(1);
    Fin.setBackground(new Color(255, 205, 205));
    Fin.setText("Ultima base a guardar");
    menu1.add(mnArchivo);
    mnArchivo.add(mnAbrir);
    mnArchivo.add(mnCerrar);
    mnArchivo.add(mnGuardar);
    this.add(panel2,  BorderLayout.NORTH);
    this.add(panel3,  BorderLayout.CENTER);
    panel3.add(textSecuencia, null);
    panel3.add(btEdit, null);
    panel2.add(Nombre, null);
    panel2.add(textnombre, null);
    panel2.add(Posicion, null);
    panel2.add(textPosicion, null);
    panel2.add(Cambio, null);
    panel2.add(textCambio, null);
    panel2.add(Inicio, null);
    panel2.add(txtInicio, null);
    panel2.add(Fin, null);
    panel2.add(txtFin, null);
    }

void mnAbrir_actionPerformed(ActionEvent e){
     String archivo;
     fd.setMode(fd.LOAD);
     fd.show();
    if(fd.getFile()==null){
       return;
  }else{
   int start=0;
```

```
    String line, element;
    int counter=0;
    archivo = fd.getDirectory()+fd.getFile();
    textSecuencia.setText(fd.getDirectory()+fd.getFile()+"\n");
      try{
     File f= new File(archivo);
     BufferedReader br= new BufferedReader(new FileReader(f));
     while((line=br.readLine())!=null){
       start++;
       if(start>1){
         StringTokenizer st=new StringTokenizer(line);
         while(st.hasMoreTokens()){
           element=st.nextToken().toString();
           char[] c=element.toCharArray();
           for(int k=0;k<c.length; k++){
             if(c[k]!=' '){
               String symbol=String.valueOf(c[k]);
               datos.put(new Integer(counter),symbol);
               counter++;
             } //end of if
           } //end of for loop
         } //end of while
       } //end of first if
     } //end of first while
     int j=0;
    }catch(IOException ex){}
  }
}
 void mnGuardar_actionPerformed(ActionEvent e){
     int salto=0;
     String secuencia="";
     fd.setMode(fd.SAVE);
     fd.show();
     if(fd.getFile()==null)return;
     try{
     FileOutputStream fo=new
     FileOutputStream(fd.getDirectory()+"\\"+fd.getFile());
     secuencia=">";
     secuencia+=textnombre.getText()+"\n";
     textSecuencia.append("\nGuardando\n");

     for (int i=Integer.parseInt(txtInicio.getText())- 1;
           i<Integer.parseInt(txtFin.getText());i++){
       secuencia+=datos.get(new Integer(i));
       salto++;
       if(salto%60==0){
       secuencia+="\n";
       } //end of if
     } //end of for loop
     textSecuencia.append("Archivo guardado\n");
```

```
    fo.write(secuencia.getBytes());
    fo.close();

  }catch(IOException ex){
  System.out.println("Error al escribir archivo");}
}
void mnCerrar_actionPerformed(ActionEvent e){
    System.exit(0);
}


  void btEdit_actionPerformed(ActionEvent e) {
    String change=textCambio.getText();
    int pos=(Integer.parseInt(textPosicion.getText()))-1;
    textSecuencia.append("\nBase original: ");
    textSecuencia.append((String)datos.get(new Integer(pos)));
    datos.put(new Integer(pos),change);
    textSecuencia.append("\nBase introducida: ");
    textSecuencia.append((String)datos.get(new Integer(pos)));
    textSecuencia.append("\n");
  }
}
```

ChangeAndOrder.pl

```
#!/usr/bin/perl -w

##############################################
# Format file name and stores flies from same individue in a direcctory
# ChangeAndOrder.pl  V.0.5
# By Julio Fernandez Banet
# Last modified 19/05/2003
# V.0.5 added regular expresion for abi sequence files
##############################################

use File::Copy;

$NewName=('');

# Check that the user introduces the necesary arguments
if (!$ARGV[0]){
  print "please introduce the directory where sequence files are stored\n";
  print "Example: perl ChangeAndOrder.pl
/users/juliofer/Documents/STQWork/\n";
  exit;
}

chdir("$ARGV[0]") or die "Could not open directory";
  @Allseqs = glob('*');
```

```
  for($i=0; $i<=$#Allseqs; $i++){
    $SeqName= $Allseqs[$i];
    $SeqName=~ tr/a-z/A-Z/;

# Gives the same format to the names of the files
    if
($SeqName=~/NP*(\s*\d+\(*\d*\)*)\s*(EX\s*\d+)\s*\d*\W*\w*\s*(F\s*\d*)\s*\w*/){
      $NewName="$1$2$3.seq";
    }
    elsif
($SeqName=~/NP*(\s*\d+\(*\d*\)*)\s*(EX\s*\d+)\s*\d*\W*\w*\s*(R\s*\d*)\s*\w*/){
      $NewName="$1$2$3.seq";
    }
    elsif
($SeqName=~/NP*(\s*\d+\(*\d*\)*)\s*(F\s*\d*)\s*\d*\W*\w*\s*(EX\s*\.*\d+)\s*\w*
/){
      $NewName="$1$3$2.seq";
    }
    elsif
($SeqName=~/NP*(\s*\d+\(*\d*\)*)\s*(R\s*\d*)\s*\d*\W*\w*\s*(EX\s*\.*\d+)\s*\w*
/){
      $NewName="$1$3$2.seq";
    }
    elsif ($SeqName=~/NP*(\s*\d+\(*\d*\)*)\s*(\w+\s*\d*)\s*\w*/){
      $NewName="$1$2.seq";
    }
    elsif ($SeqName=~/NP*\d+/){
      print "$SeqName is a directory\n";
      next;
    }
    elsif ($SeqName=~/\w+\_\w+\_(\d+EX\w+)\_\w*/){
      $NewName="$1.seq";
    }
    else {print "$SeqName\n";}
    print "$NewName\n";

# Place all the files from the same individue in a common directory
    $NewName=~ tr/\ //d;  # removes all white spaces in the name of the
sequence
    @DirName= split(/[\ _.BEF\(]/, $NewName);
    $Dir=$DirName[0];
    mkdir($Dir);
    move ("$SeqName", "$Dir/$NewName");
    $NewName=('');
  }
```

AnotateDifs.pl

```
#!/usr/bin/perl

######################
# Blast to sequences and annotate differences betwen them
# By Julio Fernandez Banet
# AnotateDifs.pl V.1.4
# V.1.0 First public version
# V.1.1 Added annotation of protein domain nomenclature
# V.1.2 Add name of output sequence to the list of output variable anotated
(i.e. N11Ex11F2.seq.out)
# V.1.3 Bug Corrected that gives wrong mRNA position when there was an
indel in the Wildtype sequence.
# V.1.4 Added Gap open penalty equals 0  so big insertions or deletions are
also displayed
# V.1.5 Only first alignment is considered to take the mRNA coordinates
# Last modified dd/mm/yy  25/08/2004
######################

# Call BioPerl Modules
use Bio::AlignIO;
use Bio::SeqIO;
use Bio::Tools::BPbl2seq;
use Bio::Tools::Run::StandAloneBlast;
use Bio::Tools::Blast::HSP;
use IO::Handle;

STDOUT->autoflush(1);
STDERR->autoflush(1);

# Check that the user introduces the necesary arguments
if (!$ARGV[0] && !$ARGV[1] && !$ARGV[2]){
  print "please introduce the directory where sequence files are stored\n";
  print "Example: perl AnotateDifs.pl /home/usuario/sequences/
  /home/usuario/WildType_sequences/pkd1DNAcds
  /home/usuario/WildType_sequences/pkd1mRNAFasta\n";
  exit;
}

# Create some necessary variables
$directory=$ARGV[0];
$WildDNA=$ARGV[1];
$WildRNA=$ARGV[2];


# Creates a list of individual directories
chdir("$directory") or die "Could not open directory";
@AllDirs=glob('*');
```

```
# Creates the file where mutation lists are going to be stored
#open(OUT, ">>/home/usuario/Mutations/MutDNA.txt");
open(OUT,">>/tmp/MutDNA.txt");

#Open each directory and creates list of files
foreach $Dir (@AllDirs){
  chdir("$Dir");
  @AllSeqs=glob('*.seq');
  foreach $Seq (@AllSeqs){
    print "$Seq\n";

  #Get 2 sequences to compare
  # Get the Wild-Type gene sequence
  my $Gene=Bio::SeqIO->new(-file=>"$WildDNA",
                  '-format'=>'Fasta');
  my $Seq2=$Gene->next_seq();


  # Get the individual sequence
  my $Seq_in =Bio::SeqIO->new(-file=> $Seq, '-format'=>'Fasta');
  my $Seq1=$Seq_in->next_seq();

if($Seq1 ne ""){

  # Run bl2seq with the two sequences
    $factory=Bio::Tools::Run::StandAloneBlast-
>new('program'=>'blastn','G'=>'0','outfile'=>"$Seq.out");
    my $bl2seq_report=$factory->bl2seq($Seq1, $Seq2);

  # Take the result from the bl2seq and parse it
    my $report = Bio::Tools::BPbl2seq->new(-file=> "$Seq.out",-
display_id=>"$Seq", -report_type =>'blastn');
    $name=$report->sbjctName;
    $report->sbjctLength;

    if ($name ne ""){ # if alignment is not null
 $firstaln=0;
      while(my $hsp = $report->next_feature ) {
                                                                                #
Select only the first alginment from the blast report
        if( $hsp->length>100 && $firstaln==0){ # Select only alignments longer
that 100 bases
                $firstaln=1;
                $Query= $hsp->querySeq();  #Sequence from individual
                $Subject= $hsp->sbjctSeq();# PKD1 gen sequence
                $Homolog= $hsp->homologySeq(); # Cigar sequence of the
alignment
          &cDNAcoor($Subject); # Call the subroutine to get the coordinates of
the changes
```

```
                                        # compared to mRNA seq instead of gene
seq
        $Hit_Beg= $hsp->hit->start;
             $Hit_End= $hsp->hit->end;
             $Query_Beg= $hsp->query->start;

             if( $Subject =~ /\-/ ) {
#                print "subject has gaps\n";#prints not so usefull info to the
screen
            }

            if( $Query =~ /\-/ ) {
#                print "query has gaps\n"; #prints not so usefull info to the screen
            }
#             print "$Query\n"; #prints not so usefull info to the screen
#             print "$Subject\n"; #prints not so usefull info to the screen


# Check if alignment is forward or reverse and makes correction to calculate
change position
        $hit_strand = $hsp->hit->strand;
        if ($hit_strand == 1){
          $cDNABegin=$cBegin;
        }elsif ($hit_strand == -1){
          $cDNABegin=$cEnd;
        }
 #       print "$query_strand\n";
 #       print "$hit_strand\n";

#             print $Hit_Beg,"\t",$Hit_End."\t",$Seq."\t",$cDNABegin."\n";

             # Call the Sequence comparison subroutine and creates a report of
             # diference betwen wild-type gene and individual
             &Compare($Dir, $Query, $Subject, $Homolog, $Hit_Beg, $Hit_End,
$cDNABegin,
             $hit_strand, $QDNAbeq, $QDNAend, $Query_Beg, $Seq);
             print OUT $Output; # Writes the changes to a file
            }

     } #end of while
   }# end of if $name
      # some descriptors
      #
      # query   agatgtggtgggag
      # Homolog ||||||||||||||
      # Subject agatgtggtgggag
      #
   $bl2seq_report->close();
   $report->close();
   }#End of if($Seq1)
```

```
  }# End of foreach $Seq
  chdir("$directory") or die "Could not open directory";

}# End of foreach $Dir
close OUT;
print "Programa ejecutado con exito\n";

#############################################################
# Subroutine for base change annotation

sub Compare{
  $Output=('');
  $cDNAPos=0;
  $protPos=0;
  $position=0; # Show global position where mismatch ocurred
  $place=-1; # Shows position where mismatch detected
  $correction=1;# Corrects position when there is an indel in the sequence
  $IndCorrect=0;
  $OurfilePosition=0;

# Change strings into arrays
  @Query=split (//, $Query);
  @Subject=split(//, $Subject);
  @Homolog=split(//, $Homolog);

  foreach $cigar(@Homolog){
    if ($cigar eq "|"){
      $place++;
    }
    if ($cigar eq "\ "){
      $place++;
      $Ind=$Query[$place];#Individual DNA seq position
      $Gen=$Subject[$place];# Wild type gene DNA seq position

      if ($Gen eq "-"){
        $correction++;
      }
      if ($Ind eq "-"){
        $IndCorrect++;
      }

# Position in individual fasta sequence
      $OurFilePosition=$Query_Beg+($place-$IndCorrect);

# Calculates the position where change occurred
      if ($hit_strand == 1){ # if forward alignment
        $position=$Hit_Beg+($place-$correction);
        $cDNAPos=$cDNABegin+($place-$QDNAbeg)+1;
      }
      elsif ($hit_strand == -1){ # if reverse alginment
```

```
        $position=$Hit_End-($place-$correction);
        $cDNAPos=$cDNABegin-($place-$QDNAbeg)+1;
      }
 #     print "$cDNAPos\n";
# 211 bp correction before the mRNA first ATG. This correction is
# necesary to estimate the amino acid position affected by the nlucleotide
change
# Divided by 3 to calculate the a.a. position
      $protPos=(($cDNAPos-212)/3);
# Split the a.a. result to transform the result into an integer
      $protPos=~/(\d+)\.*(\d*)/;
      if($2){
      $protPos=$1+1;
      }else{
      $protPos="$1";
      }

# Call subroutine to annotate the type of change
      $NewInd=&Symbol($Gen, $Ind);

# Call the subroutine to indicate the exon affected by the change
      $NewPosition=&Exon($position);

# If change ocurred in an intron turn  mRNA position and estimated
# a.a position values to zero
      if ($NewPosition=~/\d+\tIntron/){
        $protPos=0;
        $cDNAPos=0;
      }

 # Call subroutine to annotate protein which protein domain corresponds
 # to the position where the a.a sequence was affected
      &Domain($cDNAPos);

#print "$cDNAPos\n";

#Return all the values

$Output=$Output.$Dir."\t".$NewPosition."\t".$cDNAPos."\t".$Gen."\t".$NewInd
."\t".$protPos."\t"
      .$OurFilePosition."\t".$Seq.".out\n";
    }
  }
}

############################################################
# sub routine for correct base annotation

sub Symbol{
```

```
  if($Gen eq "a"){
    if($Ind eq "m" || $Ind eq "c"){$Ind="c\tTransversion";}
    elsif($Ind eq "r" || $Ind eq "g"){$Ind="g\tTransition";}
    elsif($Ind eq "w" || $Ind eq "t"){$Ind="t\tTransversion";}
    elsif($Ind eq "-"){$Ind="-\tDeletion";}
    else{ $Ind ="n\tUnknown";}
  }
  elsif($Gen eq "c"){
    if($Ind eq "m" || $Ind eq "a"){$Ind="a \tTransversion";}
    elsif($Ind eq "s" || $Ind eq "g"){$Ind="g \tTransversion";}
    elsif($Ind eq "y" || $Ind eq "t"){$Ind="t \tTransition";}
    elsif($Ind eq "-"){$Ind="- \tDeletion";}
    else{ $Ind ="n \tUnknown";}
  }
  elsif($Gen eq "g"){
    if($Ind eq "r" || $Ind eq "a"){$Ind="a \tTransition";}
    elsif($Ind eq "s" || $Ind eq "c"){$Ind="c \tTransversion";}
    elsif($Ind eq "k" || $Ind eq "t"){$Ind="t \tTransversion";}
    elsif($Ind eq "-"){$Ind="- \tDeletion";}
    else{ $Ind ="n \tUnknown";}
  }
  elsif($Gen eq "t"){
    if($Ind eq "w" || $Ind eq "a"){$Ind="a \tTransversion";}
    elsif($Ind eq "y" || $Ind eq "c"){$Ind="c \tTransition";}
    elsif($Ind eq "k" || $Ind eq "g"){$Ind="g \tTransversion";}
    elsif($Ind eq "-"){$Ind="- \tDeletion";}
    else{ $Ind ="n \tUnknown";}
  }
  elsif($Gen eq "-"){
    if($Ind ne "a" || $Ind ne "c" || $Ind ne "g" || $Ind ne "t"){
      $Ind=$Ind."\tInsertion";
    }else{$Ind=$Ind."\tInsertion";}
  }
  return $Ind;
}

############################################################
# Subroutine to indicate if the change is within an exon

sub Exon{

  SWITCH:{
    if ($position >3648 && $position<3862) {$position=$position."\t1"; last
SWITCH;}
    if ($position >19903 && $position<19974) {$position=$position."\t2"; last
SWITCH;}
    if ($position >20095 && $position<20167) {$position=$position."\t3"; last
SWITCH;}
    if ($position >20435 && $position<20605) {$position=$position."\t4"; last
SWITCH;}
```

```
   if ($position >20818 && $position<21492) {$position=$position."\t5"; last
SWITCH;}
   if ($position >21608 && $position<21792) {$position=$position."\t6"; last
SWITCH;}
   if ($position >22227 && $position<22448) {$position=$position."\t7"; last
SWITCH;}
   if ($position >22636 && $position<22753) {$position=$position."\t8"; last
SWITCH;}
   if ($position >23162 && $position<23289) {$position=$position."\t9"; last
SWITCH;}
   if ($position >23704 && $position<23903) {$position=$position."\t10"; last
SWITCH;}
   if ($position >24346 && $position<25111) {$position=$position."\t11"; last
SWITCH;}
   if ($position >25988 && $position<26120) {$position=$position."\t12"; last
SWITCH;}
   if ($position >26317 && $position<26493) {$position=$position."\t13"; last
SWITCH;}
   if ($position >26807 && $position<26941) {$position=$position."\t14"; last
SWITCH;}
   if ($position >27406 && $position<31024) {$position=$position."\t15"; last
SWITCH;}
   if ($position >31244 && $position<31393) {$position=$position."\t16"; last
SWITCH;}
   if ($position >32327 && $position<32471) {$position=$position."\t17"; last
SWITCH;}
   if ($position >32598 && $position<32868) {$position=$position."\t18"; last
SWITCH;}
   if ($position >32961 && $position<33185) {$position=$position."\t19"; last
SWITCH;}
   if ($position >33251 && $position<33410) {$position=$position."\t20"; last
SWITCH;}
   if ($position >33801 && $position<33952) {$position=$position."\t21"; last
SWITCH;}
   if ($position >36954 && $position<37098) {$position=$position."\t22"; last
SWITCH;}
   if ($position >37701 && $position<38330) {$position=$position."\t23"; last
SWITCH;}
   if ($position >38624 && $position<38780) {$position=$position."\t24"; last
SWITCH;}
   if ($position >38961 && $position<39213) {$position=$position."\t25"; last
SWITCH;}
   if ($position >39337 && $position<39533) {$position=$position."\t26"; last
SWITCH;}
   if ($position >41027 && $position<41198) {$position=$position."\t27"; last
SWITCH;}
   if ($position >41285 && $position<41428) {$position=$position."\t28"; last
SWITCH;}
   if ($position >41528 && $position<41733) {$position=$position."\t29"; last
SWITCH;}
```

```
    if ($position >41823 && $position<41950) {$position=$position."\t30"; last
SWITCH;}
    if ($position >43609 && $position<43726) {$position=$position."\t31"; last
SWITCH;}
    if ($position >43816 && $position<43856) {$position=$position."\t32"; last
SWITCH;}
    if ($position >44081 && $position<44275) {$position=$position."\t33"; last
SWITCH;}
    if ($position >44352 && $position<44496) {$position=$position."\t34"; last
SWITCH;}
    if ($position >47384 && $position<47502) {$position=$position."\t35"; last
SWITCH;}
    if ($position >47581 && $position<47783) {$position=$position."\t36"; last
SWITCH;}
    if ($position >47856 && $position<48050) {$position=$position."\t37"; last
SWITCH;}
    if ($position >48501 && $position<48640) {$position=$position."\t38"; last
SWITCH;}
    if ($position >49002 && $position<49114) {$position=$position."\t39"; last
SWITCH;}
    if ($position >49405 && $position<49547) {$position=$position."\t40"; last
SWITCH;}
    if ($position >49687 && $position<49813) {$position=$position."\t41"; last
SWITCH;}
    if ($position >49996 && $position<50171) {$position=$position."\t42"; last
SWITCH;}
    if ($position >50418 && $position<50709) {$position=$position."\t43"; last
SWITCH;}
    if ($position >50784 && $position<50919) {$position=$position."\t44"; last
SWITCH;}
    if ($position >51003 && $position<51308) {$position=$position."\t45"; last
SWITCH;}
    if ($position >51398 && $position<52883) {$position=$position."\t46"; last
SWITCH;}
    else {$position=$position."\tIntron";}
  }
return $position;
}

#################################################################
# Gives the coordinates of the changes in the cDNA
# It performs an alignment betwen the wildtype gene segment, obtain from the
# previous alignment between the wildtype and the individual seq, and the
# wildtype mRNA sequence
sub cDNAcoor{
  use Bio::Seq;

  $firstseq=0;
  $cBegin=0;
  $cEnd=0;
```

```
  # Get the Wild_type gene DNA segment from the previous alignment
  # and take away all the indels
  $SbjctNoIndels=$Subject;
  $SbjctNoIndels=~ s/\-//g;

  # Get the cDNA Wild-Type sequence
  my $cDNA=Bio::SeqIO->new(-file=>"$WildRNA",
                   '-format'=>'Fasta');
  my $Seq3=$cDNA->next_seq();

  my $Seq4=Bio::Seq->new( -display_id => 'PKD1_gene',
                     -seq=>"$SbjctNoIndels");

  #my $Seq4=$Wild->seq();
  #print $Seq4;

  # Run bl2seq with
  my $factory=Bio::Tools::Run::StandAloneBlast->new('program'=>'blastn',
'outfile'=>"$Seq.Coor.out");
  my $bl2seq_coor=$factory->bl2seq($Seq3, $Seq4);

  my $newreport = Bio::Tools::BPbl2seq->new(-file=> "$Seq.Coor.out");

  $exists=$newreport->sbjctName;
  $length=$newreport->sbjctLength;

  if($exists){
  while(my $hsp = $newreport->next_feature) {

    if ($firstseq==0){ # Select only the first alginment from the blast report
      $cBegin= $hsp->query->start;  #cDNA sequence start
      $cEnd= $hsp->query->end;  #cDNA sequence end
      $QDNAbeg= $hsp->hit->start; #Gen sequence start
      $QDNAend= $hsp->hit->end; #Gen sequence end
      $firstseq=1;
    }
  }
  }
  $newreport->close();
  $bl2seq_coor->close();

  return ($cBegin, $cEnd, $QDNAbeg, $QDNAend, $length);

}
#############################################################
# Subroutine to indicate the domain to which the CDNA position belongs
sub Domain{

  SWITCH:{
```

```
   if($cDNAPos>305 && $cDNAPos<406){$cDNAPos=$cDNAPos."\tLRR N-
FLANK"; last SWITCH;}
   if($cDNAPos>425 && $cDNAPos<493){$cDNAPos=$cDNAPos."\tLRRI";
last SWITCH;}
   if($cDNAPos>500 && $cDNAPos<568){$cDNAPos=$cDNAPos."\tLRRII";
last SWITCH;}
   if($cDNAPos>587 && $cDNAPos<751){$cDNAPos=$cDNAPos."\tLRR C-
FLANK"; last SWITCH;}
   if($cDNAPos>758 && $cDNAPos<931){$cDNAPos=$cDNAPos."\tWSC";
last SWITCH;}
   if($cDNAPos>1019 && $cDNAPos<1270){
     if($cDNAPos>1125 && $cDNAPos<1144){$cDNAPos=$cDNAPos."\tPKD
domain 1 core"; last SWITCH;}
     $cDNAPos=$cDNAPos."\tPKD domain 1"; last SWITCH;}
   if($cDNAPos>1424 && $cDNAPos<1813){$cDNAPos=$cDNAPos."\tC-
Type lectin"; last SWITCH;}
   if($cDNAPos>2126 && $cDNAPos<2224){$cDNAPos=$cDNAPos."\tLDL-A
like"; last SWITCH;}
   if($cDNAPos>2740 && $cDNAPos<2998){
     if($cDNAPos>2846 && $cDNAPos<2869){$cDNAPos=$cDNAPos."\tPKD
domain 2 core"; last SWITCH;}
     $cDNAPos=$cDNAPos."\tPKD domain 2"; last SWITCH;}
   if($cDNAPos>2999 && $cDNAPos<3205){
     if($cDNAPos>3110 && $cDNAPos<3127){$cDNAPos=$cDNAPos."\tPKD
domain 3 core"; last SWITCH;}
     $cDNAPos=$cDNAPos."\tPKD domain 3"; last SWITCH;}
   if($cDNAPos>3259 && $cDNAPos<3580){
     if($cDNAPos>3374 && $cDNAPos<3393){$cDNAPos=$cDNAPos."\tPKD
domain 4 core"; last SWITCH;}
     $cDNAPos=$cDNAPos."\tPKD domain 4"; last SWITCH;}
   if($cDNAPos>3581 && $cDNAPos<3838){
     if($cDNAPos>3692 && $cDNAPos<3712){$cDNAPos=$cDNAPos."\tPKD
domain 5 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 5"; last SWITCH;}}
   if($cDNAPos>3839 && $cDNAPos<4069){
     if($cDNAPos>3938 && $cDNAPos<3957){$cDNAPos=$cDNAPos."\tPKD
domain 6 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 6"; last SWITCH;}}
   if($cDNAPos>4127 && $cDNAPos<4342){
     if($cDNAPos>4193 && $cDNAPos<4213){$cDNAPos=$cDNAPos."\tPKD
domain 7 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 7"; last SWITCH;}}
   if($cDNAPos>4343 && $cDNAPos<4600){
     if($cDNAPos>4451 && $cDNAPos<4471){$cDNAPos=$cDNAPos."\tPKD
domain 8 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 8"; last SWITCH;}}
   if($cDNAPos>4601 && $cDNAPos<4843){
     if($cDNAPos>4710 && $cDNAPos<4729){$cDNAPos=$cDNAPos."\tPKD
domain 9 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 9"; last SWITCH;}}
```

```
   if($cDNAPos>4850 && $cDNAPos<5095){
     if($cDNAPos>4955 && $cDNAPos<4972){$cDNAPos=$cDNAPos."\tPKD
domain 10 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 10"; last SWITCH;}}
   if($cDNAPos>5099 && $cDNAPos<5356){
     if($cDNAPos>5207 && $cDNAPos<5227){$cDNAPos=$cDNAPos."\tPKD
domain 11 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 11"; last SWITCH;}}
   if($cDNAPos>5357 && $cDNAPos<5608){
     if($cDNAPos>5465 && $cDNAPos<5488){$cDNAPos=$cDNAPos."\tPKD
domain 12 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 12"; last SWITCH;}}
   if($cDNAPos>5611 && $cDNAPos<5863){
     if($cDNAPos>5720 && $cDNAPos<5732){$cDNAPos=$cDNAPos."\tPKD
domain 13 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 13"; last SWITCH;}}
   if($cDNAPos>5864 && $cDNAPos<6109){
     if($cDNAPos>5972 && $cDNAPos<5988){$cDNAPos=$cDNAPos."\tPKD
domain 14 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 14"; last SWITCH;}}
   if($cDNAPos>6116 && $cDNAPos<6385){
     if($cDNAPos>6227 && $cDNAPos<6244){$cDNAPos=$cDNAPos."\tPKD
domain 15 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 15"; last SWITCH;}}
   if($cDNAPos>6386 && $cDNAPos<6604){
     if($cDNAPos>6491 && $cDNAPos<6511){$cDNAPos=$cDNAPos."\tPKD
domain 16 core"; last SWITCH;}
     else{$cDNAPos=$cDNAPos."\tPKD domain 16"; last SWITCH;}}
   if($cDNAPos>6923 && $cDNAPos<8500){$cDNAPos=$cDNAPos."\tREJ";
last SWITCH;}
   if($cDNAPos>9245 && $cDNAPos<9391){$cDNAPos=$cDNAPos."\tGPS";
last SWITCH;}
   if($cDNAPos>9434 && $cDNAPos<9514){$cDNAPos=$cDNAPos."\tTM 1";
last SWITCH;}
   if($cDNAPos>9563 &&
$cDNAPos<9871){$cDNAPos=$cDNAPos."\tPLAT/LH2"; last SWITCH;}
   if($cDNAPos>10061 && $cDNAPos<10108){$cDNAPos=$cDNAPos."\tTM
2"; last SWITCH;}
   if($cDNAPos>10181 && $cDNAPos<10242){$cDNAPos=$cDNAPos."\tTM
3"; last SWITCH;}
   if($cDNAPos>10871 && $cDNAPos<10939){$cDNAPos=$cDNAPos."\tTM
4"; last SWITCH;}
   if($cDNAPos>10946 && $cDNAPos<11011){$cDNAPos=$cDNAPos."\tTM
5"; last SWITCH;}
   if($cDNAPos>11027 && $cDNAPos<11027){$cDNAPos=$cDNAPos."\tTM
6"; last SWITCH;}
   if($cDNAPos>11213 && $cDNAPos<11278){$cDNAPos=$cDNAPos."\tTM
7"; last SWITCH;}
   if($cDNAPos>11897 && $cDNAPos<11962){$cDNAPos=$cDNAPos."\tTM
8"; last SWITCH;}
```

```
   if($cDNAPos>12017 && $cDNAPos<12085){$cDNAPos=$cDNAPos."\tTM
9"; last SWITCH;}
   if($cDNAPos>12293 && $cDNAPos<12358){$cDNAPos=$cDNAPos."\tTM
10"; last SWITCH;}
   if($cDNAPos>12374 && $cDNAPos<12439){$cDNAPos=$cDNAPos."\tTM
11"; last SWITCH;}
   if($cDNAPos>12788 &&
$cDNAPos<12955){$cDNAPos=$cDNAPos."\tCoiled-coil"; last SWITCH;}
   else{$cDNAPos=$cDNAPos."\tInterdomain";}
  }
return $cDNAPos;
}
```

ProtTrans.pl

```
#!/usr/bin/perl

#####################################
# A program to translate individual DNA seqs 6 frame, get the correct
translation
# and find the changes betwen wild-type and subject sequence
# protTrans.pl v.1.0
# By Julio Fernandez Banet
# Last modified dd/mm/yy 03/12/2003
# V.0.1 First functional version
# V.0.2 a.a. position bug fixed (wrong position if indel in sequence)
# V.0.3 Added predicted affected secondary structure for each amino acid
change
# V.0.4 Added amino acid characteristics
# v.1.0 Added the name of the outputfile of the alignment to the list of
characteristics that are annotated.
#####################################

# Get an EMBOSS program to work
use Bio::Factory::EMBOSS;
use Bio::SeqIO;
use Bio::AlignIO;
use IO::Handle;

STDOUT->autoflush(1);
STDERR->autoflush(1);

# Check that the user introduces the necesary arguments
if (!$ARGV[0] || !$ARGV[1] || !$ARGV[2]){
  print "please introduce the directory where sequence files are stored\n";
  print "Example: perl protTrans.pl /home/usuario/sequences
/home/usuario/WildType/prot /home/usuario/Wildtype/PKD1Struct.txt\n";
  exit;
}
```

```
# Create some necesary variables
$directory=$ARGV[0];
$WildProt=$ARGV[1];
$structfile=$ARGV[2];

# Load the list of predicted secondary structure for each amino acid position to
an array
open(IN,$structfile);
@secondstruct=<IN>;
foreach $secondstruct(@secondstruct){
$secondary=$secondary.$secondstruct;
}
@struct=split(//,$secondary);

# Create a list of all individual directories
chdir("$directory") or die "Could not open directory";
@AllDirs=glob('*');

#Creates the files where the protein changes are going to be stored
open(OUT, ">>/tmp/MutProt.txt");


# Open a directory and creates a list of files
foreach $Dir (@AllDirs){
  chdir($Dir) or  die "Could not open directory";
  @proteins=glob('*.seq');

  foreach $protein(@proteins){
  #  print "$protein\n";
     $f = Bio::Factory::EMBOSS -> new(); # Factory for transeq
     $g = Bio::Factory::EMBOSS -> new(); # Factory for supermatcher

  # get an EMBOSS application  object from the factory
     $transeq = $f->program('transeq');
     $supermatcher =$g->program('supermatcher');

  # Define outfiles for both programs
     my $transeqoutfile = "$protein.prot";
#    print "$transeqoutfile\n";

     my $smoutfile = "$protein.supermatch";
#    print "$smoutfile\n";

  # Run transeq program to geet the six frmae translation of individual
sequences
     $transeq->run({ '-sequence' => $protein,
                  '-outseq'   => $transeqoutfile,
                  '-frame' => '6'});

  #Get the set of sequences to compare
```

```
 # Get the Wild-Type gene sequence
  $seqa = "$WildProt";

 # Get the individual sequence
   $seqb = "$transeqoutfile";

 #Run supermatcher to get the correct frame of the six translated
   $supermatcher->run({ '-aseq' => $seqa,
                        '-bseq' => $seqb,
                        '-outfile' => $smoutfile,
                        '-gapopen' => '10.0',
                        '-gapextend' => '0.5'});


 # now you might want to get the alignment
 $alnin = new Bio::AlignIO(-format => 'emboss',
                           -file   => $smoutfile);

    while( $aln = $alnin->next_aln ) {
      if ($aln->percentage_identity > 70){ # Filter bad quality alignments
# Get the polycystin info from the alginment
        $seq1= $aln->get_seq_by_pos(1);
        $pkd1= $seq1->seq;
        $pkd1Start= $seq1->start;
        $pkd1End=$seq1->end;
# Get the individual a.a. seq info from the alignment
        $seq2= $aln->get_seq_by_pos(2);
        $indiv= $seq2->seq;
        $indivStart=$seq2->start;
        $indivEnd=$seq2->end;

# Get the homology line
        $Homology = $aln->match_line;

# Get the pkd1 protein sequence and place each element in an array
        @pkd=split (//, $pkd1);
# Get the individual protein sequence to an array
        @ind=split(//, $indiv);
# Get the homology sequence to an array
        @Homolog=split(//, $Homology);

        $place = -1;
        $Output=('');
        $ChangeType=('');
        $position=0;
        $correction=0;

        foreach $Homolog (@Homolog){
          if ($Homolog eq "*"){ # if not a.a. change
            $place++;
```

```
        }else{ # if a.a. change
          $place++;
          $IndSym=$ind[$place];
          $Symbol=$IndSym;

# Call subroutine to add a.a properties
          &Characts($Symbol);
          $IndSym=$Symbol;
          $pkdSym=$pkd[$place];
          $pkdSym=uc($pkdSym); # to uppercase
          if($pkdSym eq "-"){ #
            $correction++;
          }
          $Symbol=$pkdSym;
# Call subroutine to add a.a. properties
          &Characts($Symbol);
          $pkdSym=$Symbol;
          $position=$pkd1Start+($place-$correction);
# annotate type of change based on homology line symbol
          if ($Homolog eq ":"){
          $ChangeType="Conservative";
          }elsif ($Homolog eq "."){
          $ChangeType="Non-Conservative";
          }elsif ($Homolog eq " "){
          $ChangeType="Nonsense";
          }# end of if else bucle
 #          print $Homolog;
 #          print $ChangeType;
# Call subroutine to add secondary structure that corresponds to the affected
possition
         #&Garnier($place, $secondary);
         &Garnier($place, $struct);

$Output=$Output.$Dir."\t".$position."\t".$IndSym."\t".$pkdSym."\t".$ChangeTy
pe."\t"
         .$structure."\t".$protein.".supermatch\n";
         }
       } # end of foreach $homolog
 #       print $Output;
       print OUT $Output; # Save mutation list to file
     } # end of if percentage_identity

    } # end of while

  }# End of foreach $protein

# Return to main directory
  chdir("$directory") or die "Could not open directory";
}# End of foreach $Dir
close OUT;
```

```
#####################################################################
##
# Consensus secondary structure

sub Garnier{
$structure='';
$value='';
#  @struct=split(//,$secondary);
  $value=$struct[$place];
  if ($value eq "H"){
    $structure="Helix";
  }elsif($value eq "T"){
    $structure="Loop";
  }elsif($value eq "E"){
    $structure="Sheet";
  }elsif($value eq "C"){
    $structure="Coil";
  }
  return $structure;
}

#####################################################################
#
# Amino acid caracteristics

sub Characts{

  SWITCH:{
    if($Symbol eq "G" || $Symbol eq "g"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aliphatic, Tiny"; last SWITCH}
    if($Symbol eq "A" || $Symbol eq "a"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aliphatic, Tiny"; last SWITCH}
    if($Symbol eq "V" || $Symbol eq "v"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aliphatic, Small"; last SWITCH}
    if($Symbol eq "L" || $Symbol eq "l"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aliphatic"; last SWITCH}
    if($Symbol eq "I" || $Symbol eq "i"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aliphatic"; last SWITCH}
    if($Symbol eq "P" || $Symbol eq "p"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aliphatic, Cyclic, Small"; last SWITCH}
    if($Symbol eq "F" || $Symbol eq "f"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, Aromatic"; last SWITCH}
    if($Symbol eq "Y" || $Symbol eq "y"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, H-Bonding, Ionizable"; last SWITCH}
    if($Symbol eq "W" || $Symbol eq "w"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, H-Bonding, Aromatic"; last SWITCH}
    if($Symbol eq "S" || $Symbol eq "s"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding, Tiny"; last SWITCH}
```

```
    if($Symbol eq "T" || $Symbol eq "t"){ $Symbol=$Symbol."\tPolar,
hydrophilic, Aliphatic, Tiny"; last SWITCH}
    if($Symbol eq "N" || $Symbol eq "n"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding, Small"; last SWITCH}
    if($Symbol eq "Q" || $Symbol eq "q"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding"; last SWITCH}
    if($Symbol eq "C" || $Symbol eq "c"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, H-Bonding, SulfurContaining, Ionizable, Cs-s, Ch-h,
ChargedAtNeutralpHAcidic"; last SWITCH}
    if($Symbol eq "M" || $Symbol eq "m"){ $Symbol=$Symbol."\tNonpolar,
hydrophobic, SulfurContaining"; last SWITCH}
    if($Symbol eq "D" || $Symbol eq "d"){ $Symbol=$Symbol."\tPolar,
hydrophilic, Ionizable, H-Bonding, Small, ChargedAtNeutralpHAcidic"; last
SWITCH}
    if($Symbol eq "E" || $Symbol eq "e"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding, Ionizable, ChargedAtNeutralpHAcidic"; last SWITCH}
    if($Symbol eq "H" || $Symbol eq "h"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding, Ionizable(Aromatic), ChargedAtNeutralpHBasic"; last
SWITCH}
    if($Symbol eq "K" || $Symbol eq "k"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding, Ionizable, ChargedAtNeutralpHBasic"; last SWITCH}
    if($Symbol eq "R" || $Symbol eq "r"){ $Symbol=$Symbol."\tPolar,
hydrophilic, H-Bonding, Ionizable, ChargedAtNeutralpHBasic"; last SWITCH}
    if($Symbol eq "*"){ $Symbol=$Symbol."\tStop Codon"; last SWITCH}
    else{$Symbol=$Symbol."\tUnknown"; last SWITCH}
  }
 return $Symbol;
}
```

Create.PKDGnosis.DB

```
DROP DATABASE IF EXISTS PKDGnosys;

CREATE DATABASE IF NOT EXISTS PKDGnosys;

use PKDGnosys;

create table InDNA (Nind int(8), DNApos int(8), Exon int(3), mRNApos int(6),
Domain char(20),

GenSym char(2), InSym char(2), ChType char(15), PredProtpos int(5),
ABIPos int(4), dnaFile char(25));

create table InProt (Nind int(8), ProtPos int(5), InSym char(2), InAAProp
char(95),

GenSym char(2), GenAAProp char(95), ChType char(20), Struct char(6),
aaFile char(35));

create table Silent(Nind int(8), DNApos int(8), Exon int(3), mRNApos int(6),
Domain char(20),

GenSym char(2), InSym char(2), ChType char(15), PredProtpos int(5),
ABIPos int(4), dnaFile char(25));

create table NoSilent(Nind int(8), DNApos int(8), Exon int(3), mRNApos int(6),
Domain char(20),

GenSym char(2), InSym char(2), ChType char(15), PredProtpos int(5),
ABIPos int(4), dnaFile char(25));

create table Phenotype (Apellidos char(20),      nombre char(10), Diagnostic
char(35), afecto char(30), Familia int(10),

Nind int(6), ObsFeno char(30), Nacemento char(10), Dx char(10), Exitus
char(25), Observado char(30), Protocol char(30));

create table FamiliasPKD (NFamilia int(5), NombreFamilia char(20), Gene
int(5), Mutaciones char(30), Link decimal(6),

IRCT int(3), Vascula char(2), Infantil char(2), Hepatico char(2), obserxen
char(20), obserfen char(20), comment char(20),

afacer char(20));

LOAD DATA LOCAL INFILE "/tmp/MutDNA.txt" into table InDNA;

LOAD DATA LOCAL INFILE "/tmp/MutProt.txt" into table InProt;

LOAD DATA LOCAL INFILE "/tmp/NEFX11.txt" into table Phenotype;

LOAD DATA LOCAL INFILE "/tmp/FamiliaPKD.txt" into table FamiliasPKD;
```

CreateIndexes

```
use PKDGnosys;

Create index IndPheno on Phenotype(Nind);
Create index indexFamilia on Phenotype(Familia);
Create index NumFam on Phenotype(Familia);
Create index NumFamInd on Phenotype(Familia, Nind);
Create index NFamPKD on FamiliasPKD(NFamilia);
Create index Patient on InDNA(Nind);
Create index DNACHType on InDNA(chType);
Create index DNAposition on InDNA(DNApos);
Create index mRNAposition on InDNA(mRNApos);
Create index Exones on InDNA(Exon);
Create index Domains on InDNA(Domain);
Create index Patient2 on InProt(Nind);
Create index ProtPosition on InProt(Protpos);
Create index ProtCHType on InProt(ChType);
Create index SNPS on InDNA(Nind, PredProtpos);
Create index Mutation on InProt(Nind, ProtPos);

insert into Silent Select InDNA.* FROM InDNA LEFT JOIN InProt ON
InDNA.Nind=InProt.Nind AND InDNA.PredProtpos=InProt.ProtPos
WHERE InProt.ProtPos IS NULL GROUP BY InDNA.Nind, InDNA.DNApos,
InDNA.PredProtpos;

insert into NoSilent Select InDNA.* FROM InDNA LEFT JOIN InProt ON
InDNA.Nind=InProt.Nind AND InDNA.PredProtpos=InProt.ProtPos
WHERE InProt.ProtPos IS NOT NULL GROUP BY InDNA.Nind,
InDNA.DNApos, InDNA.PredProtpos;

Create index SilentInd on Silent(Nind);
Create index SilentchType on Silent(chType);
Create index SilentDNApos on Silent(DNApos);
Create index SilentmRNA on Silent(mRNApos);
Create index SilentExon on Silent(Exon);
Create index SilentDomain on Silent(Domain);
Create index SilentIndProt on Silent(Nind, PredProtpos);

Create index NoSilentInd on NoSilent(Nind);
Create index NoSilentchType on NoSilent(chType);
Create index NoSilentDNApos on NoSilent(DNApos);
Create index NoSilentmRNA on NoSilent(mRNApos);
Create index NoSilentExon on NoSilent(Exon);
Create index NoSilentDomain on NoSilent(Domain);
Create index NoSilentIndProt on NoSilent(Nind, PredProtpos);
```

PKDGnosys.phtml

```
<!-- Plantilla paginaweb -->
<?php
  function Head($HeadTitle)
  {
?>

<html>
  <head>
  <?php
    echo("<title>PKDGnosys. PKD1 mutation database. $HeadTitle</title>");
  ?>
 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
 <meta name="keywords" content="PKD1, polycystin, mutation, database,
ADPKD">
 <meta name="description" content="PKD1 mutation database contains
information of changes in DNA sequence from 200 patients of ADPKD
compared with the wildtype gene described in GenBank">
 <meta name="author" content="mpjfb@usc.es">
 <STYLE TYPE="text/css">
<!--
  .menu {
    position:absolute;
          visibility:hidden;
    background-color: white;
    layer-background-color: white;
    color: black;
    border-style: solid;
    border-color: black;
    border-width: 1px;
    padding: 3px;
    font-size : 12px;
    font-family: "arial", "helvetica";
  }

 .menu A:hover {text-decoration: underline; color: red;}
 .menu A {text-decoration: none; color: black;}
-->
</STYLE>

<SCRIPT language="javascript">

// Objeto de deteccion del navegador
function DetectorNavegador() {
  this.NS4 = document.layers;
  this.IE4 = document.all;
  this.DOM = document.getElementById;
  this.DHTML = this.NS4 || this.IE4 || this.DOM;
}
```

```
var soporta = new DetectorNavegador();
var menu = new Array();
var menuActivo = null;

// Objeto Menu
function activarMenu() {
  if (soporta.DHTML && menuActivo != this) {
    if (menuActivo) menuActivo.ocultar();
    menuActivo = this;
    this.mostrar();
  }
}

function mostrarMenu() {
  eval(this.capaRefStr + this.estiloRefStr + '.visibility = "visible"');
  if (soporta.DOM)
        this.domRef.style.display = "block";
}

function ocultarMenu() {
  eval(this.capaRefStr + this.estiloRefStr + '.visibility = "hidden"');
}

function cambiarPosicionMenu(top, left) {
  eval(this.capaRefStr + this.estiloRefStr + this.topRefStr + ' = top');
  eval(this.capaRefStr + this.estiloRefStr + this.leftRefStr + ' = left');
  if (soporta.DOM)
        this.domRef.style.display = "none";
}

function Menu(capaID, top, left, width) {
  this.activar = activarMenu;
  this.mostrar = mostrarMenu;
  this.ocultar = ocultarMenu;
  this.cambiarPosicion = cambiarPosicionMenu;
  if (soporta.DOM) {
    this.domRef = document.getElementById(capaID);
        this.domRef.style.width = width;
    this.domRef.style.display = "none";
  }
  this.capaRefStr = (soporta.NS4) ?
    'document["'+capaID+'"]' :
    ((soporta.IE4) ? 'document.all["'+capaID+'"]' : 'this.domRef');
  this.estiloRefStr = (soporta.NS4) ? '' : '.style';
  this.topRefStr = (soporta.IE4) ? '.pixelTop' : '.top';
  this.leftRefStr = (soporta.IE4) ? '.pixelLeft' : '.left';
  this.cambiarPosicion(top, left);
}
```

```
// Manejo de eventos
function ocultarMenuActivo(e) {
  if (menuActivo) {
    menuActivo.ocultar();
    menuActivo = null;
  }
}

// Inicializacion
function inicializar() {
  if (soporta.DHTML) {
    if (soporta.NS4)
      document.captureEvents(Event.MOUSEUP);
    document.onmouseup = ocultarMenuActivo;
  }
  menu[0] = new Menu("menu0", 100, 5, 100);
  menu[1] = new Menu("menu1", 100, 93, 120);
  menu[2] = new Menu("menu2", 100, 200, 310);
}

window.onload = inicializar;
</SCRIPT>
 </head>
  <body LINK="#0000FF" VLINK="#0000FF" bgcolor="#CCFFFF"
background="#00FFFF">

<!--  header  -->
<?php }
  function Banner(){
?>

<!--  Banner  -->
<table width="960" border="0" cellspacing="0" cellpadding="0"
bgcolor="#33CCFF">
<tr valign="top">
      <td width="120" height="60" align="center" valign="top"><a
href="./index.html"><img src="images/Logo.gif" width="76" height="60"
border="0"></a></td>
      <td width="840" height="60"><img src="images/Banner.jpg" width="840"
height="60" border="0"></td>
    </tr>
  </table>

<?php }
  function LinkBar(){
?>
<!--  LinkBar  -->
<DIV id="menu0" CLASS="menu">
  <A HREF="index.php">Main</A><BR>
  <A HREF="tutorial.php">tutorial</A><BR>
```

```
</div>

<DIV id="menu1" CLASS="menu">
  <A HREF="TheLab.php">The Lab</A><BR>
  <A HREF="Staff.php">The Staff</A><BR>
</div>

<DIV id="menu2" CLASS="menu">
  <A HREF="http://www.pkdcure.org/">About PKD</A><BR>
  <A
HREF="http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?33359212:NCBI:55242
44">PKD1 DNA sequence</A><BR>
  <A
HREF="http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=nucl
eotide&list_uids=4505832&dopt=GenBank">PKD1 mRNA
sequence</A><BR>
  <A
HREF="http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=prot
ein&list_uids=1730587&dopt=GenPept">PKD1 Protein sequence</A><BR>
  <A
HREF="http://uwcmml1s.uwcm.ac.uk/uwcm/mg/search/120293.html">HGMD
</A><BR>
</div>

<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0"
WIDTH="960" BGCOLOR="#9933CC">
<TD WIDTH="120">
<a href="index.php" style="text-decoration: none; color: white;"
onmouseover="if (menu[0]) menu[0].activar();" >PKDGnosys</A>
</TD>

<TD WIDTH="120">
<a href="TheLab.php" style="text-decoration: none; color: white;"
onmouseover="if (menu[1]) menu[1].activar();" >The Lab</A>
</TD>

<TD WIDTH="200">
<a href="Links.php" style="text-decoration: none; color: white;"
onmouseover="if (menu[2]) menu[2].activar();" >PKD external links</A>
</TD>

<TD width="70%" HEIGHT="25"> </TD>
<?php }
  function Body(){
?>

<!--  Body  -->

    <table width="960" border="0" cellspacing="0" cellpadding="0">
     <tr>
```

```
 <?php }
  function LeftBar1(){
 ?>

   <!-- left bar  -->
  <td width="120" valign="top">
     <table border="0">
    </table>
    <p><br>
   </td>
<?php }
 function RightBar(){
?>

  <!--  Right Bar  -->
    <td width="15" valign="top"></td> <!--//Skip the overlaping between
regions-->
    <td width="840" valign="top">

<?php }
 function PageFoot(){
?>

   </td>
   </tr>

   </table>

   <table width="640" border="0" cellspacing="0" cellpadding="0">
    <tr><td width="115"> </td>
    <td width="525"><HR><span class="Bodytext">This Web site is created,
developed, and maintained by the Laboratorio de Nefrolog&iacute;a, CHUS,
Santiago de Compostela (Spain).<br>
   If you have suggestions or corrections.<br>
   Please send your feedback using our form at: <a
href="feedback.php">PKDGnosys Support form</a><br>
   Or mail us at: <a href="mailto:mpjfb@usc.es?subject=PKDGnosys
support">PKDGnosys mail</a>
<br>

   Last Update: 25/09/2003 version 1.0
</span>
</td></tr>
   </table>
</body>
</html>

<?php }
?>
```

Index.php

```
<?php include("PKDGnosys.phtml")
?>
<?php Head("Database Query");
    Banner();
    LinkBar();
    Body();
    LeftBar1();
    RightBar();
?>

<?php
$db = mysql_connect("localhost", "user", "password"); //Connect to the
database
mysql_select_db("PKDGnosys", $db);  //Select the database to use
?>

 Search PKD1 database

<form name='query' method=post Action="RESPONSEGraph.php">
  <table width="400" border="1">
    <tr>
      <td>Input Exon Name</td>
      <td>
        <select name="Exon" size="1">
          <option value="">
          <option value="1">1
          <option value="2">2
          <option value="3">3
          <option value="4">4
          <option value="5">5
          <option value="6">6
          <option value="7">7
          <option value="8">8
          <option value="9">9
          <option value="10">10
          <option value="11">11
          <option value="12">12
          <option value="13">13
          <option value="14">14
          <option value="15">15
          <option value="16">16
          <option value="17">17
          <option value="18">18
          <option value="19">19
          <option value="20">20
          <option value="21">21
          <option value="22">22
          <option value="23">23
          <option value="24">24
```

```
        <option value="25">25
        <option value="26">26
        <option value="27">27
        <option value="28">28
        <option value="29">29
        <option value="30">30
        <option value="31">31
        <option value="32">32
        <option value="33">33
        <option value="34">34
        <option value="35">35
        <option value="36">36
        <option value="37">37
        <option value="38">38
        <option value="39">39
        <option value="40">40
        <option value="41">41
        <option value="42">42
        <option value="43">43
        <option value="44">44
        <option value="45">45
        <option value="46">46
      </select>
    </td>
  </tr>
  <tr>
    <td>Input Family Number</td>
    <td>
     <select name="Family" size="1">
      <option value="">
<?php
  $SearchFamilies ="SELECT Individue.NFamily
                    FROM InDNA, Individue
                    WHERE InDNA.Nind=Individue.Nind
                    GROUP BY Individue.NFamily";

  $Families= mysql_query($SearchFamilies,$db);

  while($Fam= mysql_fetch_row($Families)){
  printf("<option value='%s'>%s",$Fam[0],$Fam[0]);
  }

?>
      </select>
    </td>
  </tr>
  <tr>
    <td>Input Individual number</td>
    <td>
      <select name="Individual" size="1">
```

```
        <option value="">
       <?php
   $SearchIndividue ="SELECT InDNA.Nind
                                FROM InDNA GROUP BY InDNA.Nind";
   $Individue= mysql_query($SearchIndividue,$db);

   while($Ind= mysql_fetch_row($Individue)){
   printf("<option value='%s'>%s",$Ind[0],$Ind[0]);
   }

?>
       </select>
     </td>
   </tr>
   <tr>
     <td>Type of DNA change</td>
     <td>
       <select name="Mutation" size="1">
         <option value="">
         <option value="Unknown">Unknown
         <option value="Insertion">Insertion
         <option value="Deletion">Deletion
         <option value="Transition">Transition
         <option value="Transversion">Transversion
       </select><br>
       <input type=checkbox name=filter value="false">Filter "base -> n"
changes
     </td>
   </tr>
   <tr>
     <td>DNA mutation change position</td>
     <td>
       <input type="text" name="DNApos">
     </td>
   </tr>
   </table>
   <table>
   <tr><input type=checkbox name=onlySNPS value="false">Select only
changes that don't change a.a chain (Silent polymorfisms)</td>
   </tr>
   <tr><input type=checkbox name=noIntron value="false">Don't display
Intron information</td>
   </tr>
   <br><br>
   <tr>
     <td><FONT color = "#FF0000"> Choose this if you want to see
information of the amino acid chain.</FONT></td>
   </tr>
   <tr>
```

```
    <td><input type=checkbox name=aainfo value="true"
onclick=mi_onclick()>Select changes that also cause a.a. change </td>
      <script type="text/javascript">
       function mi_onclick(){
        if(document.query.aainfo.checked){
         document.query.Domain.disabled=false;
         document.query.Struct.disabled=false;
         document.query.aachange.disabled=false;
        }else{
         document.query.Domain.disabled=true;
         document.query.Struct.disabled=true;
         document.query.aachange.disabled=true;
        }
       }
      </script>
  </tr>
  </table>
  <table width="400" border="1">
  <tr>
   <td>Protein Domain</td>
   <td>
    <select name="Domain" size="1" disabled=true>
     <option value="">
     <option value="Interdomain">Interdomain
     <option value="LRR N-FLANK">LRR N-FLANK
     <option value="LRRI">LRRI
     <option value="LRRII">LRRII
     <option value="LRR C-FLANK">LRR C-FLANK
     <option value="WSC">WSC
     <option value="PKD domain 1">PKD domain 1
     <option value="PKD domain 1 core">PKD domain 1 core
     <option value="C-Type lectin">C-Type lectin
     <option value="LDL-A like">LDL-A like
     <option value="PKD domain 2">PKD domain 2
     <option value="PKD domain 2 core">PKD domain 2 core
     <option value="PKD domain 3">PKD domain 3
     <option value="PKD domain 3 core">PKD domain 3 core
     <option value="PKD domain 4">PKD domain 4
     <option value="PKD domain 4 core">PKD domain 4 core
     <option value="PKD domain 5">PKD domain 5
     <option value="PKD domain 5 core">PKD domain 5 core
     <option value="PKD domain 6">PKD domain 6
     <option value="PKD domain 6 core">PKD domain 6 core
     <option value="PKD domain 7">PKD domain 7
     <option value="PKD domain 7 core">PKD domain 7 core
     <option value="PKD domain 8">PKD domain 8
     <option value="PKD domain 8 core">PKD domain 8 core
     <option value="PKD domain 9">PKD domain 9
     <option value="PKD domain 9 core">PKD domain 9 core
     <option value="PKD domain 10">PKD domain 10
```

```
      <option value="PKD domain 10 core">PKD domain 10 core
      <option value="PKD domain 11">PKD domain 11
      <option value="PKD domain 11 core">PKD domain 11 core
      <option value="PKD domain 12">PKD domain 12
      <option value="PKD domain 12 core">PKD domain 12 core
      <option value="PKD domain 13">PKD domain 13
      <option value="PKD domain 13 core">PKD domain 13 core
      <option value="PKD domain 14">PKD domain 14
      <option value="PKD domain 14 core">PKD domain 14 core
      <option value="PKD domain 15">PKD domain 15
      <option value="PKD domain 15 core">PKD domain 15 core
      <option value="PKD domain 16">PKD domain 16
      <option value="PKD domain 16 core">PKD domain 16 core
      <option value="REG">REG
      <option value="GPS">GPS
      <option value="TM1">TM1
      <option value="TM2">TM2
      <option value="TM3">TM3
      <option value="TM4">TM4
      <option value="TM5">TM5
      <option value="TM6">TM6
      <option value="TM7">TM7
      <option value="TM8">TM8
      <option value="TM9">TM9
      <option value="TM10">TM10
      <option value="TM11">TM11
      <option value="Coiled-coil">Coiled-coil
    </select>
  </td>
</tr>
<tr>
  <td>Amino Acid Change type</td>
  <td>
    <select name="aachange" size="1" disabled=true>
      <option value="">
      <option value="Conservative">Conservative
      <option value="Non-Conservative">Non-Conservative
      <option value="Missense">Missense
    </select>
  </td>
</tr>
<tr>
  <td>Secondary Structure</td>
  <td>
    <select name="Struct" size="1" disabled=true>
      <option value="">
      <option value="Helix">Helix
      <option value="Sheet">Sheet
      <option value="Loop">Loop
      <option value="Coil">Coil
```

```
      </select>
    </td>
  </tr>
  </table>
  <table>
  <tr>
    <td>Select how to order the information:</td>
  </tr>
  </table>
  <table width="400" border="1">
  <tr>
    <td><input type=radio name=ordenar value="orFamily">Family</td>
    <td><input type=radio name=ordenar value="orInd">Individual</td>
    <td><input type=radio name=ordenar value="orExon">Exon</td>
  </tr>
  <tr>
    <td><input type=radio name=ordenar value="orDNA">DNA position</td>
    <td><input type=radio name=ordenar value="ormRNA">mRNA
position</td>
    <td><input type=radio name=ordenar value="orProt">a.a. position</td>
  </tr>
  </table>
  <table>
  <tr>
    <td>
      <input type="submit" name="Submit" value="Send request">
    </td>
    <td>
      <input type="reset" value="Clear Form" name="reset">
    </td>
  </tr>
 </table>
</form>

<?php PageFoot();
?>
```

ResponseGraph.php

```
<?php include("PKDGnosys.phtml");
?>
<?php Head("Sequence Query");
Banner();
Linkbar();
Body();
RightBar(); ?>

<!---- Database Query ---->
<?php

$origin=$_SERVER["QUERY_STRING"];
if($origin!="") $next=1;
$variables=explode("&",$origin);
for($i=0;$i<count($variables);$i++){
$tab=explode("=",$variables[$i]);
$tabla1=str_replace("%20"," ","$tab[1]");
$$tab[0]=$tabla1;
}
if (!$next){
$Exon= $_POST[ 'Exon'];
$Family= $_POST[ 'Family'];
$Individual= $_POST[ 'Individual'];
$Mutation= $_POST[ 'Mutation'];
$Domain= $_POST[ 'Domain'];
$aachange= $_POST[ 'aachange'];
$Struct= $_POST[ 'Struct'];
$aainfo= $_POST[ 'aainfo'];
$ordenar= $_POST[ 'ordenar'];
$DNApos= $_POST[ 'DNApos'];
$Filter= $_POST[ 'filter'];
$onlySNPS= $_POST[ 'onlySNPS'];
$noIntron= $_POST[ 'noIntron'];

}
// if comes from SearchVisual
if(!$MaxRecordsPerPage) $MaxRecordsPerPage=25;

//Limit the number of record displayed avoid server overload
if ($MaxRecordsPerPage>75)

ErrorMessage("Wrong: you have requested to many records");

$db = mysql_connect("localhost", "user", "password"); //Connect to the
database
mysql_select_db("PKDGnosys", $db);  //Select the database to use
```

```
//Generate the SQL command for doing a select from the DataBase to display
results in the results table
if($onlySNPS){
$searchStmt = "SELECT Silent.Nind,
Silent.DNApos,
Silent.Exon,
Silent.mRNApos,
Silent.Domain,
Silent.GenSym,
Silent.InSym,
Silent.ChType,
Silent.PredProtpos,
Silent.ABIPos,
Individue.NFamily,
Silent.dnaFile
FROM Silent, Individue
WHERE Silent.Nind!=\"\" AND
Silent.Nind=Individue.Nind AND ";

if ($Exon) {$searchStmt .="Silent.Exon=\"$Exon\" AND ";}//if variable exists
if ($Individual)   {$searchStmt .="Silent.Nind=\"$Individual\" and ";} //if the
variable exist
if ($Mutation)     {$searchStmt .="Silent.ChType=\"$Mutation\" and ";}//if the
variable exists
if ($Family)    {$searchStmt .="Individue.NFamily=\"$Family\" and ";}//if
variable exists
if ($Domain)    {$searchStmt .="Silent.Domain=\"$Domain\" and ";}//if variable
exists
if ($DNApos)   {$searchStmt .="Silent.DNApos=\"$DNApos\" and ";}
if ($Filter) {$searchStmt .="Silent.InSym!=\"n\" and ";}
if ($noIntron) {$searchStmt .="Silent.Exon!=\"Intron\" and ";}


// Query for text results presentation
$stmt = substr($searchStmt, 0, strlen($searchStmt)-4);
if ($ordenar==orDNA) $stmt .="order by Silent.DNApos";
if ($ordenar==orInd)  $stmt .="order by Silent.Nind";//order the result by
Individual
if ($ordenar==orFamily) $stmt .="order by Silent.NFamily";//order the result by
family
if ($ordenar==orExon) $stmt .="order by Silent.Exon";
if ($ordenar==ormRNA) $stmt .="order by Silent.mRNApos";
}else{
  if ($aainfo){   //show only changes where aa change is also confirmed
  $searchStmt = "SELECT InDNA.Nind,
                        InDNA.DNApos,
                        InDNA.Exon,
                        InDNA.mRNApos,
                        InDNA.Domain,
                        InDNA.GenSym,
```

```
                        InDNA.InSym,
                        InDNA.ChType,
                        InDNA.PredProtpos,
                        InDNA.ABIPos,
                        Individue.NFamily,
                        InDNA.dnaFile,
                        InProt.ProtPos,
                        InProt.InSym,
                        InProt.InAAProp,
                        InProt.GenSym,
                        InProt.GenAAProp,
                        InProt.ChType,
                        InProt.Struct,
                        InProt.aaFile
                        FROM InDNA, InProt, Individue
                        WHERE InDNA.Nind=InProt.Nind AND
                        InDNA.Nind=Individue.Nind AND
                        InDNA.PredProtpos=InProt.ProtPos AND ";
  }else{
  $searchStmt = "SELECT InDNA.Nind,
  InDNA.DNApos,
  InDNA.Exon,
  InDNA.mRNApos,
  InDNA.Domain,
  InDNA.GenSym,
  InDNA.InSym,
  InDNA.ChType,
  InDNA.PredProtpos,
  InDNA.ABIPos,
  Individue.NFamily,
  InDNA.dnaFile
  FROM InDNA, Individue
  WHERE InDNA.Nind!=\"\" AND
  InDNA.Nind=Individue.Nind AND ";
}

if ($Exon) {$searchStmt .="InDNA.Exon=\"$Exon\" AND ";}//if variable exists
if ($Individual)   {$searchStmt .="InDNA.Nind=\"$Individual\" and ";} //if the
variable exist
if ($Mutation)     {$searchStmt .="InDNA.ChType=\"$Mutation\" and ";}//if the
variable exists
if ($Family)    {$searchStmt .="Individue.NFamily=\"$Family\" and ";}//if
variable exists
if ($Domain)    {$searchStmt .="InDNA.Domain=\"$Domain\" and ";}//if variable
exists
if ($aachange)    $searchStmt .="InProt.ChType=\"$aachange\" and ";//if
variable exists
if ($Struct)    $searchStmt .="InProt.Struct=\"$Struct\" and ";//if variable exists
if ($DNApos)   {$searchStmt .="InDNA.DNApos=\"$DNApos\" and ";}
if ($Filter) {$searchStmt .="InDNA.InSym!=\"n\" and ";}
```

```
if ($noIntron) {$searchStmt .="InDNA.Exon!=\"Intron\" and ";}


// Query for text results presentation
$stmt = substr($searchStmt, 0, strlen($searchStmt)-4);
if ($aainfo) $stmt .="group by InDNA.Nind, InDNA.PredProtPos ";
if ($ordenar==orDNA) $stmt .="order by InDNA.DNApos";
if ($ordenar==orInd)  $stmt .="order by InDNA.Nind";//order the result by
Individual
if ($ordenar==orFamily) $stmt .="order by InDNA.NFamily";//order the result
by family
if ($ordenar==orExon) $stmt .="order by InDNA.Exon";
if ($ordenar==ormRNA) $stmt .="order by InDNA.mRNApos";
if ($ordenar==Prot)  $stmt .="order by InProt.ProtPos";

}
// Query to get the total number of result
$numberofresult = mysql_query($stmt, $db);

// Get the total number for results
$total_records = mysql_num_rows($numberofresult);

// redefine search to limit it
if (!$initialRecord)    $initialRecord=0;

$stmt = $stmt. " limit $initialRecord,$MaxRecordsPerPage";

$result = mysql_query($stmt,$db);



//**************************************************
// if number of records is 0 do not present the next
//**************************************************

if (!$total_records) {

SearchCriteria($Exon, $Family, $Individual, $Mutation, $Domain, $aachange,
$Struct, $DNApos, $Filter, $noIntron, $onlySNPS); //call to the funtion
ErrorMessage("No records found for the selected parameters");
}

SearchCriteria($Exon, $Family, $Individual, $Mutation, $Domain, $aachange,
$Struct, $DNApos, $Filter, $noIntron, $onlySNPS);

echo"<HR>\n";

echo"<!----- Result graphics (GRAPH) ----->\n";
```

```
echo "<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0
WIDTH=\"900\" bordercolor=\"#000000\">\n";

echo"<br>";
if(!$QueryToUse){
echo  "<img
src=\"IMAGES/graphics.php?Exon=$Exon&Family=$Family&Individual=$Indiv
idual&Mutation=$Mutation&DNApos=$DNApos&Domain=$Domain&Filter=$Fil
ter&aainfo=$aainfo&onlySNPS=$onlySNPS\">\n";
}else {
echo  "<img src=\"IMAGES/graphics.php?$QueryToUse\">\n";
}
echo "<br>\n";
echo "<br>\n";

echo"<!---------- Result Report (HEAD) ------->\n";

echo "<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=1
WIDTH=\"900\" bordercolor=\"#000000\">\n";

//define header of the report
echo "<tr><td bgcolor=\"#CCFFCC\" width=50><b> <font size=-1><center>
Individual </center></font></b></td>\n";
echo "<td bgcolor=\"#CCFFFF\" width=\"50\"><b> <font size=-1><center>
Family </center></font></b></td>\n";
//echo "<td bgcolor=\"#FFFFGG\" width=\"50\"><b><font size=-
1><center> ABI file position </center></font></b></td>\n";
echo "<td bgcolor=\"#CCCCFF\" width=\"50\"><b> <font size=-
1><center> DNA position </center></font></b></td>\n";
echo "<td bgcolor=\"#CCCCFF\" width=\"50\"><b> <font size=-
1><center> mRNA position </center></font></b></td>\n";
echo "<td bgcolor=\"#CCCCFF\" width=\"50\"><b> <font size=-
1><center> Exon </center></font></b></td>\n";
echo "<td bgcolor=\"#CCCCFF\" width=\"50\"><b> <font size=-
1><center> Wild-type nucleotide </center></font></b></td>\n";
echo "<td bgcolor=\"#CCCCFF\" width=\"50\"><b> <font size=-
1><center> Individual nucleotide </center></font></b></td>\n";
echo "<td bgcolor=\"#CCCCFF\" width=\"50\"><b> <font size=-
1><center> Nucleotide change type </center></font></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\" width=\"50\"><b> <font size=-
1><center> predicted amino acid position </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> DNA
alignment </center></b></td>\n";

if($aainfo){
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> amino
acid position </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> Protein
Domain </center></b></td>\n";
```

```
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> Wild-type
amino acid </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> Individual
amino acid </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> Wild-type
amino acid properties </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> Individual
amino acid properties </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> amino
acid change type </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-
1><center> Secondary structure predicted </center></b></td>\n";
echo "<td bgcolor=\"#FFCCFF\"><b> <font size=-1><center> aa
alingment</center></b></td>\n";

}
echo "</tr>\n";


// each value for the report list
$RecordsCounter=0;

echo"<!---------- Result Report (VALUES) ------->\n";
while ($myquery = mysql_fetch_row($result)) {
$Indiv=$myquery[0];
$fam=$myquery[10];
$RecordsCounter++;
printf("<td bgcolor=\"#AAFFAA\" width=\"5\"><font size=-1><center><a
href='phenotype.php?Individual=%s'> %s</center></font></td>\n",$Indi
v,$myquery[0]);//NInd
printf("<td bgcolor=\"#AAFFDD\" width=\"5\"><font size=-1><center><a
href='treeGIF/%s'> %s</center></font></td>\n",$fam,$myquery[10]);//N
Family
//printf("<td bgcolor=\"#FFFF33\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[9]);//ABIpos
printf("<td bgcolor=\"#AAAAFF\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[1]);//DNApos
printf("<td bgcolor=\"#AAAAFF\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[3]);//mRNApos
printf("<td bgcolor=\"#AAAAFF\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[2]);//Exon
printf("<td bgcolor=\"#AAAAFF\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[5]);//GenSym
printf("<td bgcolor=\"#AAAAFF\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[6]);//IndSym
printf("<td bgcolor=\"#AAAAFF\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[7]);//ChType
printf("<td bgcolor=\"#FFCCCC\" width=\"5\"><font size=-
1><center> %s</center></font></td>\n",$myquery[8]);//PretProtpos
```

```
printf("<td bgcolor='#FFCCCC'> <center><input type=button
value=\"View\"onclick=location='seqs/%s/%s'></center></td>",$myquery[0],$
myquery[11]);//DNA alignment

if($aainfo){
printf("<td bgcolor=\"#FFCCCC\"><font size=-
1><center> %s</center></font></td>\n",$myquery[12]);//ProtPos
printf("<td bgcolor=\"#FFCCCC\"><font size=-
1><center> %s</center></font></td>\n",$myquery[4]);//Domain
printf("<td bgcolor=\"#FFCCCC\"><font size=-
1><center> %s</center></font></td>\n",$myquery[15]);//GenSym
printf("<td bgcolor=\"#FFCCCC\"><font size=-
1><center> %s</center></font></td>\n",$myquery[13]);//IndSym
printf("<td bgcolor=\"#FFCCCC\"><font size=-
2><center> %s</center></font></td>\n",$myquery[16]);//Genaaprop
printf("<td bgcolor=\"#FFCCCC\"><font size=-
2><center> %s</center></font></td>\n",$myquery[14]);//Inaaprop
printf("<td bgcolor=\"#FFCCCC\"><font size=-
1><center> %s</center></font></td>\n",$myquery[17]);//ChType
printf("<td bgcolor=\"#FFCCCC\"><font size=-
1><center> %s</center></font></td>\n",$myquery[18]);//Struct
printf("<td bgcolor='#FFCCCC'> <center><input type=button
value=\"View\"onclick=location='seqs/%s/%s'></center></td>",$myquery[0],$
myquery[19]);//aa Alignment

}
printf("</tr>\n");
}

echo "</table>\n";

echo"<HR>";

echo "<br>Records Found: <font
color=\"#CC11AA\"><b>$total_records</b></font> \n
";

if ($total_records> $MaxRecordsPerPage)
echo "Requested <font
color=\"#CC11AA\"><b>$MaxRecordsPerPage</b></font>\n";

echo "Interval (<font color=\"#CC11AA\"><b>",$initialRecord+1;
echo" - ",$initialRecord+$RecordsCounter;
echo"</b></font>)\n";
echo "Displayed <font
color=\"#CC11AA\"><b>$RecordsCounter</b></font><br>\n";

//calculate number of pages needing links
$pages=intval($total_records/$MaxRecordsPerPage);
```

```
if ($total_records%$MaxRecordsPerPage) $pages++;  //has a remainder so
add one page

//////////////////


$QueryToUse ="";

//if($SV) $QueryToUse .="SV=Y&";
if($Exon) $QueryToUse .="Exon=$Exon&";
if($Family) $QueryToUse .="Family=$Family&";
if($Individual) $QueryToUse .="Individual=$Individual&";
if($Mutation) $QueryToUse .="Mutation=$Mutation&";
if($DNApos) $QueryToUse .="DNApos=$DNApos&";
if($Domain) $QueryToUse .="Domain=$Domain&";
if($aachange) $QueryToUse .="aachange=$aachange&";
if($Struct) $QueryToUse .="Struct=$Struct&";
if($aainfo) $QueryToUse .="aainfo=$aainfo&";
if($ordenar) $QueryToUse .="ordenar=$ordenar&";
if($Filter) $QueryToUse .="Filter=$Filter&";
if($noIntron) $QueryToUse .="noIntron=$noIntron&";
if($onlySNPS) $QueryToUse .="onlySNPS=$onlySNPS&";


if($MaxRecordsPerPage) $QueryToUse
.="MaxRecordsPerPage=$MaxRecordsPerPage&";

$previousInitialRecord= $initialRecord;

// if $pages >1 present differents pages
if ($pages>1){
echo"Go to page:<br> ";
for ($i=1;$i<=$pages;$i++){

$initialRecord=$MaxRecordsPerPage*($i-1);

if ($previousInitialRecord!= $initialRecord){
echo"<a HREF=\"RESPONSEGraph.php?$QueryToUse".
"initialRecord=$initialRecord\" class=\"leftbarSubclass\">[$i] 
</a>\n";}
else {
echo"<a HREF=\"RESPONSEGraph.php?$QueryToUse".
"initialRecord=$initialRecord\" class=\"iconselected\">[$i] </a
>\n";
}
}

}

PageFoot();
```

```
//////////

function ErrorMessage($toDisplay){
echo("<p><br><center>$toDisplay</center>\n
<p><br><center><span class=\"iconselected\"><a href=\"index.html\" cl
ass=\"iconselected\">Try again</A></span></center>\n") ;

PageFoot();
exit() ;
}

//////////

function SearchCriteria($Exon, $Family, $Individual, $Mutation, $Domain,
$aachange, $Struct, $DNApos, $Filter, $noIntron, $onlySNPS){

echo"<b>Search parameters:</b><br>";

if($Exon){
echo("Exon number: ");
echo("<font color=\"#CC11AA\"><b>$Exon </b></font>");
}
if($Family){
echo("Family number: ");
echo("<font color=\"#CC11AA\"><b>$Family </b></font>");
}
if($Individual){
echo("Individual number: ");
echo("<font color=\"#CC11AA\"><b>$Individual </b></font>");
}
if($DNApos){
echo("DNA change position: ");
echo("<font color=\"#CC11AA\"><b>$DNApos </b></font>");
}
if($Mutation){
echo("Nucleotide change type: ");
echo("<font color=\"#CC11AA\"><b>$Mutation </b></font>");
}
if($Domain){
echo("Protein Domain: ");
echo("<font color=\"#CC11AA\"><b>$Domain </b></font>");
}
if($aachange){
echo("Amino acid change type: ");
echo("<font color=\"#CC11AA\"><b>$aachange </b></font>");
}
if($Struct){
echo("Protein Secondary Structure: ");
echo("<font color=\"#CC11AA\"><b>$Struct </b></font>");
```

```
}
if($Filter){
echo(" Filtered for Unknown changes");
}
if($onlySNPS){
echo(" Silent changes only");
}
if($noIntron){
Echo(" No intron information displayed");
}
}

?>
```

Graph.php

```
<?php
Header("Content-type: image/png");

$db = mysql_connect("localhost", "user", "password"); //Connect to the
database
mysql_select_db("PKDGnosys", $db);  //Select the database to use


if ($onlySNPS){
//Generate the SQL command for doing a count from the DataBase
  $countquery = "SELECT COUNT(distinct Silent.Nind, Silent.Exon) FROM
Silent, Individue
  WHERE Silent.Nind!=\"\" AND Silent.Exon!=\"Intron\" AND
Silent.Nind=Individue.Nind AND ";

//Generate the SQL command for doing a select from the DataBase to display
the graphics
  $graphquery="SELECT Silent.Nind, Silent.DNApos, Silent.Exon FROM
Silent, Individue
  WHERE Silent.Nind!=\"\" AND Silent.Exon!=\"Intron\" AND
Silent.Nind=Individue.Nind AND ";

if ($Exon) {//if variable exists
                        $countquery .="Silent.Exon=\"$Exon\" AND ";
                        $graphquery.="Silent.Exon=\"$Exon\" AND ";}
if ($Individual)   { //if the variable exist
                        $countquery .="Silent.Nind=\"$Individual\" AND ";
                        $graphquery.="Silent.Nind=\"$Individual\" AND ";}
if ($Mutation)      {//if the variable exists
                        $countquery .="Silent.ChType=\"$Mutation\" AND ";
                        $graphquery.="Silent.ChType=\"$Mutation\" AND ";}
if ($Family)    {//if variable exists
                        $countquery .="Individue.NFamily=\"$Family\" AND ";
                        $graphquery.="Individue.NFamily=\"$Family\" AND ";}
```

```
if ($Domain)    {//if variable exists
                        $countquery .="Silent.Domain=\"$Domain\" AND ";
                        $graphquery.="Silent.Domain=\"$Domain\" AND ";}
if ($Filtro){
        $countquery .="Silent.InSym!=\"n\" AND ";
        $graphquery .="Silent.InSym!=\"n\" AND ";}

//Count the number of different individues within the query

$CountString = substr($countquery, 0, strlen($countquery)-4);
$CountString .="order BY Silent.Exon, Silent.Nind";


// Query for graphical representation
$graphinfo = substr($graphquery, 0, strlen($graphquery)-4);
$graphinfo .="order by Silent.Exon, Silent.Nind "; //best order for graphical
display

}else if ($aainfo){
//Generate the SQL command for doing a count from the DataBase
  $countquery = "SELECT COUNT(distinct NoSilent.Nind, NoSilent.Exon)
FROM NoSilent, Individue
  WHERE NoSilent.Nind!=\"\" AND NoSilent.Exon!=\"Intron\" AND
NoSilent.Nind=Individue.Nind AND ";

//Generate the SQL command for doing a select from the DataBase to display
the graphics
  $graphquery="SELECT NoSilent.Nind, NoSilent.DNApos, NoSilent.Exon
FROM NoSilent, Individue
  WHERE NoSilent.Nind!=\"\" AND NoSilent.Exon!=\"Intron\" AND
NoSilent.Nind=Individue.Nind AND ";

if ($Exon) {//if variable exists
                        $countquery .="NoSilent.Exon=\"$Exon\" AND ";
                        $graphquery.="NoSilent.Exon=\"$Exon\" AND ";}
if ($Individual)   { //if the variable exist
                        $countquery .="NoSilent.Nind=\"$Individual\" AND ";
                        $graphquery.="NoSilent.Nind=\"$Individual\" AND ";}
if ($Mutation)      {//if the variable exists
                        $countquery .="NoSilent.ChType=\"$Mutation\" AND ";
                        $graphquery.="NoSilent.ChType=\"$Mutation\" AND ";}
if ($Family)    {//if variable exists
                        $countquery .="Individue.NFamily=\"$Family\" AND ";
                        $graphquery.="Individue.NFamily=\"$Family\" AND ";}
if ($Domain)    {//if variable exists
                        $countquery .="NoSilent.Domain=\"$Domain\" AND ";
                        $graphquery.="NoSilent.Domain=\"$Domain\" AND ";}
if ($Filtro){
        $countquery .="NoSilent.InSym!=\"n\" AND ";
        $graphquery .="NoSilent.InSym!=\"n\" AND ";}
```

```
//Count the number of different individues within the query

$CountString = substr($countquery, 0, strlen($countquery)-4);
$CountString .="order BY NoSilent.Exon, NoSilent.Nind";


// Query for graphical representation
$graphinfo = substr($graphquery, 0, strlen($graphquery)-4);
$graphinfo .="order by NoSilent.Exon, NoSilent.Nind "; //best order for
graphical display


}else{
//Generate the SQL command for doing a count from the DataBase
  $countquery = "SELECT COUNT(distinct InDNA.Nind, InDNA.Exon) FROM
InDNA, Individue
  WHERE InDNA.Nind!=\"\" AND InDNA.Exon!=\"Intron\" AND
InDNA.Nind=Individue.Nind AND ";

//Generate the SQL command for doing a select from the DataBase to display
the graphics
  $graphquery="SELECT InDNA.Nind, InDNA.DNApos, InDNA.Exon FROM
InDNA, Individue
  WHERE InDNA.Nind!=\"\" AND InDNA.Exon!=\"Intron\" AND
InDNA.Nind=Individue.Nind AND ";

if ($Exon) {//if variable exists
                        $countquery .="InDNA.Exon=\"$Exon\" AND ";
                        $graphquery.="InDNA.Exon=\"$Exon\" AND ";}
if ($Individual)   { //if the variable exist
                        $countquery .="InDNA.Nind=\"$Individual\" AND ";
                        $graphquery.="InDNA.Nind=\"$Individual\" AND ";}
if ($Mutation)      {//if the variable exists
                        $countquery .="InDNA.ChType=\"$Mutation\" AND ";
                        $graphquery.="InDNA.ChType=\"$Mutation\" AND ";}
if ($Family)    {//if variable exists
                        $countquery .="Individue.NFamily=\"$Family\" AND ";
                        $graphquery.="Individue.NFamily=\"$Family\" AND ";}
if ($Domain)    {//if variable exists
                        $countquery .="InDNA.Domain=\"$Domain\" AND ";
                        $graphquery.="InDNA.Domain=\"$Domain\" AND ";}
if ($Filtro){
        $countquery .="InDNA.InSym!=\"n\" AND ";
        $graphquery .="InDNA.InSym!=\"n\" AND ";}

//Count the number of different individues within the query

$CountString = substr($countquery, 0, strlen($countquery)-4);
$CountString .="order BY InDNA.Exon, InDNA.Nind";
```

```
// Query for graphical representation
$graphinfo = substr($graphquery, 0, strlen($graphquery)-4);
$graphinfo .="order by InDNA.Exon, InDNA.Nind "; //best order for graphical
display
}



$info = mysql_query($graphinfo, $db);
$counter = mysql_query($CountString, $db);
  $mycount= mysql_fetch_row($counter);
  $valor=$mycount[0];
  $Exon='0';
  $Individue='0'; // initialize the variable that show if mutation data comes from
the same individual
  $fin=10000;
  $init=0;
  $ImageHeight=$valor*10;
  $HeightBeg=1;
  $HeightEnd=10;

  $im=ImageCreate(900,$ImageHeight);
  $textcolor= ImageColorAllocate($im,255,255,255);
  $background= ImageColorAllocate($im,0,0,150);
  ImageFilledRectangle($im,0,0,900,$ImageHeight,$background);
  $mutcolor= ImageColorAllocate($im,255,0,0);
  $normalcolor= ImageColorAllocate($im,0,0,255);


while ($myquery = mysql_fetch_row($info)){
  $individuo=$myquery[0];
  $mutacion=$myquery[1];
  $ActualExon=$myquery[2];
  if(ActualExon=='Intron'){
  }
  else{
    if($Exon==$ActualExon){
      if($Individue==$individuo){
        $mut=($mutacion-$inicio)*800;
        $mutpos= floor($mut/$div);
        $mutpos2= $mutpos+50;
        $width=$mutpos2-1;
        ImageFilledRectangle($im,$init,$HeightBeg,$width,$HeightEnd-
2,$normalcolor);
        ImageFilledRectangle($im,$width,$HeightBeg,$mutpos2,$HeightEnd-
2,$mutcolor);
        $init=50+$mutpos+1;
      }
```

```
      else{
        ImageFilledRectangle($im,$init,$HeightBeg,850,$HeightEnd-
2,$normalcolor);
        $HeightBeg=$HeightBeg+10;
        $HeightEnd=$HeightEnd+10;
        $mut=($mutacion-$inicio)*800;
        $mutpos= floor($mut/$div);
        $mutpos2= $mutpos+50;
        $width=$mutpos2-1;
        ImageFilledRectangle($im,50,$HeightBeg,$width,$HeightEnd-
2,$normalcolor);
        ImageString($im,1,855,$HeightBeg,$individuo,$textcolor);
        ImageString($im,1,25,$HeightBeg,$individuo,$textcolor);
        ImageFilledRectangle($im,$width,$HeightBeg,$mutpos2,$HeightEnd-
2,$mutcolor);
        $init=50+$mutpos+1;
        $Individue=$individuo;
      }
    }else{
      if($Exon!='0'){
        //ImageString($im,1,2,0,$valor,$textcolor);
        ImageFilledRectangle($im,$init,$HeightBeg,850,$HeightEnd-
2,$normalcolor);
        $HeightBeg=$HeightBeg+10;
        $HeightEnd=$HeightEnd+10;
      }
      $Exon=$ActualExon;
      if($ActualExon=='1'){ $inicio=3648; $fin=3862; }
      elseif($ActualExon=='2'){ $inicio=19903; $fin=19974; }
      elseif($ActualExon=='3'){ $inicio=20095; $fin=20167; }
      elseif($ActualExon=='4'){ $inicio=20435; $fin=20605; }
      elseif($ActualExon=='5'){ $inicio=20818; $fin=21492; }
      elseif($ActualExon=='6'){ $inicio=21608; $fin=21792; }
      elseif($ActualExon=='7'){ $inicio=22227; $fin=22448; }
      elseif($ActualExon=='8'){ $inicio=22636; $fin=22753; }
      elseif($ActualExon=='9'){ $inicio=23162; $fin=23289; }
      elseif($ActualExon=='10'){ $inicio=23704; $fin=23903; }
      elseif($ActualExon=='11'){ $inicio=24346; $fin=25111; }
      elseif($ActualExon=='12'){ $inicio=25988; $fin=26120; }
      elseif($ActualExon=='13'){ $inicio=26317; $fin=26493; }
      elseif($ActualExon=='14'){ $inicio=26807; $fin=26941; }
      elseif($ActualExon=='15'){ $inicio=27406; $fin=31024; }
      elseif($ActualExon=='16'){ $inicio=31244; $fin=31393; }
      elseif($ActualExon=='17'){ $inicio=32327; $fin=32471; }
      elseif($ActualExon=='18'){ $inicio=32598; $fin=32868; }
      elseif($ActualExon=='19'){ $inicio=32961; $fin=33185; }
      elseif($ActualExon=='20'){ $inicio=33251; $fin=33410; }
      elseif($ActualExon=='21'){ $inicio=33801; $fin=33952; }
      elseif($ActualExon=='22'){ $inicio=36954; $fin=37098; }
      elseif($ActualExon=='23'){ $inicio=37701; $fin=38330; }
```

```
    elseif($ActualExon=='24'){ $inicio=38624; $fin=38780; }
    elseif($ActualExon=='25'){ $inicio=38961; $fin=39213; }
    elseif($ActualExon=='26'){ $inicio=39337; $fin=39533; }
    elseif($ActualExon=='27'){ $inicio=41027; $fin=41198; }
    elseif($ActualExon=='28'){ $inicio=41285; $fin=41428; }
    elseif($ActualExon=='29'){ $inicio=41528; $fin=41733; }
    elseif($ActualExon=='30'){ $inicio=41823; $fin=41950; }
    elseif($ActualExon=='31'){ $inicio=43609; $fin=43726; }
    elseif($ActualExon=='32'){ $inicio=43816; $fin=43856; }
    elseif($ActualExon=='33'){ $inicio=44081; $fin=44275; }
    elseif($ActualExon=='34'){ $inicio=44352; $fin=44496; }
    elseif($ActualExon=='35'){ $inicio=47384; $fin=47502; }
    elseif($ActualExon=='36'){ $inicio=47581; $fin=47783; }
    elseif($ActualExon=='37'){ $inicio=47856; $fin=48050; }
    elseif($ActualExon=='38'){ $inicio=48501; $fin=48640; }
    elseif($ActualExon=='39'){ $inicio=49002; $fin=49114; }
    elseif($ActualExon=='40'){ $inicio=49405; $fin=49547; }
    elseif($ActualExon=='41'){ $inicio=49687; $fin=49813; }
    elseif($ActualExon=='42'){ $inicio=49996; $fin=50171; }
    elseif($ActualExon=='43'){ $inicio=50418; $fin=50709; }
    elseif($ActualExon=='44'){ $inicio=50784; $fin=50919; }
    elseif($ActualExon=='45'){ $inicio=51003; $fin=51308; }
    elseif($ActualExon=='46'){ $inicio=51398; $fin=52883; }
   //  elseif($ActalExon=='Intron'){$fin=1;}
   //  else{$fin=1;}
    $div=$fin-$inicio;
    $mut=($mutacion-$inicio)*800;
    $mutpos= floor($mut/$div);
    $mutpos2=$mutpos+50;
    $width=$mutpos2-1;
    ImageFilledRectangle($im,50,$HeightBeg,$width,($HeightEnd-
2),$normalcolor);
    ImageString($im,1,855,$HeightBeg,$individuo,$textcolor);
    ImageString($im,1,880,$HeightBeg,$ActualExon,$textcolor);
    ImageString($im,1,25,$HeightBeg,$individuo,$textcolor);
    ImageString($im,1,1,$HeightBeg,$ActualExon,$textcolor);
    ImageFilledRectangle($im,$width,$HeightBeg,$mutpos2,($HeightEnd-
2),$mutcolor);
    $init=50+$mutpos+1;
    $Individue=$individuo;
    }
  }
}
  ImageFilledRectangle($im,$init,$HeightBeg,850,$HeightEnd-
2,$normalcolor);
  Imagepng($im);
  ImageDestroy($im);

?>
```

Phenotype.php

```
<?php include("PKDGnosys.phtml");
?>
<?php Head("Sequence Query");
Banner();
Linkbar();
Body();
RightBar(); ?>

<!---- Database Query ---->
<?php

$db = mysql_connect("localhost", "user", "password"); //Connect to the
database
mysql_select_db("PKDGnosys", $db);  //Select the database to use


//Generate the SQL command for doing a select from the DataBase to display
results in the results table

  $searchStmt = "SELECT Phenotype.Apellidos, Phenotype.nombre,
Phenotype.Diagnostic, Phenotype.afecto,

Phenotype.Familia,Phenotype.Nind,Phenotype.ObsFeno,Phenotype.Naceme
nto,Phenotype.Dx,Phenotype.Exitus,
  Phenotype.Observado,Phenotype.Protocol,FamiliasPKD.Gene,
FamiliasPKD.Link,FamiliasPKD.IRCT,FamiliasPKD.Vascula,

FamiliasPKD.Infantil,FamiliasPKD.Hepatico,FamiliasPKD.obserxen,FamiliasP
KD.obserfen,FamiliasPKD.comment,FamiliasPKD.afacer
  FROM Phenotype, FamiliasPKD
  WHERE
  FamiliasPKD.NFamilia=Phenotype.Familia AND ";

$searchStmt.= "Phenotype.Nind=\"$Individual\"";

$result = mysql_query($searchStmt,$db);
$numresult=$result;

// Get the total number for results
$total_records = mysql_num_rows($numresult);



//*************************************************
// if number of records is 0 do not present the next
//*************************************************

if (!$total_records) {
```

```
SearchCriteria($Individual); //call to the funtion
ErrorMessage("No clinical information found for the selected individual");
}

SearchCriteria($Individual);

echo"<HR>\n";
echo "<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=1
WIDTH=\"900\" bgcolor=\"#FFFFFF\" bordercolor=\"#000000\">\n";

echo"<!---------- Result Report ------->\n";

while ($myquery = mysql_fetch_row($result)) {
echo "<tr><td bgcolor=\"#FFFFFF\" width=50><b> <font size=+1><center>
Individual Number</center></font></b></td></tr>\n";
printf("<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><center> %s</center></font></td></tr>\n",$myquery[5]);//NInd

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b> <font size=+1><center>
Family Number</center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2></td></tr>\n");
//printf("<td bgcolor=\"#FFFFCC\" width=\"50\"> </td></tr>\n");
printf("<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><font
size=0><center> %s</center></font></td>\n",$myquery[4]);//NumFamil
y
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2> </td></tr>\n");


echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center> Born date </center></font></b></td>\n";
echo "<td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center> Dx date </center></font></b></td>\n";
echo "<td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center> Exitus date </center></font></b></td></tr>\n";
printf("<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><center> %s</center></font></td>\n",$myquery[7]);//Nacement
o
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><center> %s</center></font></td>\n",$myquery[8]);//Dx
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><center> %s</center></font></td></tr>\n",$myquery[9]);//exitus

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+1><center>  Afecto </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2><font
size=0><left> %s</left></font></td></tr>\n",$myquery[3]);//ESRD


echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center>  Gene </center></font></b></td>\n";
```

```
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><left> %s</left></font></td></tr>\n",$myquery[12]);//Gene

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+1><center>  Diagnostico </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2><font
size=0><left> %s</left></font></td></tr>\n",$myquery[2]);//Diagnostic

echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center>  Protocol </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><left> %s</left></font></td></tr>\n",$myquery[11]);//Protocol

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+1><center>  Observado Fenotipo </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2><font
size=0><left> %s</left></font></td></tr>\n",$myquery[6]);//Obsfeno

echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center>  Observaciones
generales</center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><left> %s</left></font></td></tr>\n",$myquery[10]);//Observado

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+1><center>  Link (LOD-Score) </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2><font
size=0><left> %s</left></font></td></tr>\n",$myquery[13]);//Link

echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center>  IRCT (ESRD) </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><left> %s</left></font></td></tr>\n",$myquery[14]);//IRCT

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+1><center>  Vascular </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2><font
size=0><left> %s</left></font></td></tr>\n",$myquery[15]);//vascula

echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+1><center>  Infantil </center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFFF\" width=\"50\"><font
size=0><left> %s</left></font></td></tr>\n",$myquery[16]);//Infantil


echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+1><center>  Hepatico</center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2><font
size=0><left> %s</left></font></td></tr>\n",$myquery[17]);//Hepatico
```

```
echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+2><center> Observaciones Genotipo Familia
</center></font></b></td></tr>\n";
printf("<tr><td bgcolor=\"#FFFFFF\" width=\"50\" colspan=3><font
size=0><center> %s</center></font></td></tr>\n",$myquery[18]);//Obs
erxen

echo "<tr><td bgcolor=\"#FFFFCC\" width=\"50\"><b><font
size=+2><center> Observaciones Fenotipo Familia
</center></font></b></td>\n";
printf("<td bgcolor=\"#FFFFCC\" width=\"50\" colspan=2></td></tr>\n");
printf("<tr><td bgcolor=\"#FFFFCC\" width=\"50\"colspan=3><font
size=0><left> %s</left></font></td></tr>\n",$myquery[19]);//Obserfen

echo "<tr><td bgcolor=\"#FFFFFF\" width=\"50\"><b><font
size=+2><center> Comentarios, Por
hacer</center></font></b></td></tr>\n";
printf("<tr><td bgcolor=\"#FFFFFF\" width=\"50\" colspan=3><font
size=0><center> %s</center></font></td></tr>\n",$myquery[20]);//Com
ment
printf("<tr><td bgcolor=\"#FFFFFF\" width=\"50\" colspan=3><font
size=0><center> %s</center></font></td></tr>\n",$myquery[21]);//Afac
er

}

echo "</table>\n";
```

Código IUPAC:

A – A Adenina
C – C Citosina
G – G Guanina
T – T Timina
R – G o A puRina
Y – T o C pYrimidina
M – A o C aMino
K – G o T Keto
S – G o C Fuerte interacción (3 puentes de hidrogeno).
W – A o T Débil iteracción (2 puentes de hidrógeno).
H – A, C o T no existe G y H es la siguiente letra en el alfabeto.
B – G, T o C no hay A y B es la siguiente letra del alfabeto.
V – G, C o A (no hay T->U) V es la siguiente letra a la U.
D – G, A o T noy hay C, D está después de C.
N – A, C, G o T Cualquiera (aNy).

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Autosomal Dominant Polycystic Kidney Disease (ADPKD) is one of the most frequent monogenic inherited diseases.

PKD1 is a complex gene with 46 exons and ~50kb. The protein that encodes (polycystin 1) contains transmembrane, extra and intracellular domains.

A set of tools was developed to study the PKD1 gene and its protein. First step was to implement the PERL-BioPERL program to automate the analysis of PKD1 gene sequences from ADPKD affected individuals, and to annotate every change found within the sequence. Second step was to integrate genotypic information in a relational database. By the use of these tools it was possible to annotate more than 384 nucleotide changes. It was also possible to identify new sequence variants.

Tertiary structure prediction methods were used for a better understanding of the role of polycystin 1. Due to its complexity (36 different domains) the protein was modeled using “ab initio” methods and with the obtained structures was possible to suggest at least two different functions for polycystin 1.

The extracellular part was suggested to play a role in cell adhesion. This was supported by the folds obtained for the different PKD domains as they appear to be organized in 7-bladed beta-propellers, structure that was found in many proteins with cell adhesion and binding functions.

For the transmembrane domains it was possible to find similarities with proteins that form cation channels and leads to the suggestion that polycystin 1 would be part of a multimeric non specific cation channel.