

Projecte fi de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Eina d'anàlisi i visualització de grafs de compatibilitat i cicles de trasplantament per a suport dels KEP (Kidney Exchange Programs)

Document: Memòria

Alumne: Maria Sunyer Rigau

Tutor: Francesc Castro i Anton Bardera
Departament: Informàtica, Matemàtica Aplicada i Estadística
Àrea: Llenguatges i Sistemes Informàtics

Convocatòria (mes/any): Juny 2021

PROJECTE FI DE GRAU

**Eina d'anàlisi i visualització de
grafs de compatibilitat i cicles de
trasplantament per a suport dels
KEP (Kidney Exchange Programs)**

Autor:

Maria SUNYER RIGAU

Juny 2021

Grau en Enginyeria Informàtica

Tutors:

Francesc CASTRO

Anton BARDERA

Resum

Els Kidney Exchange Programs (KEP) gestionen els procediments de trasplantament de ronyó creuats amb donant viu. Aquests programes permeten a pacients en necessitat d'un ronyó que tenen algun familiar o amic proper disposat a donar-lo però que no són compatibles, trobar una altra parella de pacient-donant en la mateixa situació a on si són compatibles de forma creuada. El donant vinculat a un pacient ofereix un seu ronyó a canvi que el "seu" pacient en rebí un provinent d'un altre donant compatible. En alguns casos, hi ha pacients que tenen més d'un donant vinculat o també hi ha casos de donants altruistes.

En aquest projecte, s'ha desenvolupat una aplicació web de suport als equips mèdics dels KEP's que permet la visualització de les parelles pacient-donant i els possibles trasplantaments que es podrien realitzar en format de graf. La visualització s'ha desenvolupat en JavaScript utilitzant la llibreria D3.js. En aquesta aplicació es permet seleccionar diferents nodes per visualitzar en detall la informació d'aquell pacient i els seus donants vinculats, conjuntament amb les arestes i nodes amb les que interactuen. També es pot seleccionar una aresta per veure en detall els trasplantaments que representa, la seva direcció i la seva puntuació.

A l'aplicació web també es pot visualitzar el graf de cicles del conjunt de pacients. Els cicles són un conjunt de trasplantaments que permetrien a un grup de pacients realitzar una cadena de trasplantaments a on tots rebrien un ronyó. Aquests cicles poden ser de 2 o 3 parelles de pacients. En cas de tenir un conjunt de cicles que formin una solució es pot visualitzar quins serien els trasplantaments que es realitzarien. La visualització permet veure en detall els cicles d'un node (direcció, puntuació, etc.) i també es pot seleccionar un cicle en concret per veure la informació en detall.

Un altre dels objectius del projecte és l'anàlisi del graf, mitjançant mesures de teoria de grafs i de teoria de la informació, i crear un model de regressió logística per predir si una parella pacient-donant acabarà sent escollida en un cicle de trasplantaments. Concretament, s'han aplicat diferents mesures de centralitat de teoria de grafs, com el grau d'un node o la mesura "betweenness centrality", amb la hipòtesi que un node, com més central és, més probabilitats té de ser escollit per a un trasplantament. Les mesures de teoria de la informació escollides són mesures que modelen un passeig aleatori (random walk) del graf, com la sorpresa entròpica o la informació mútua. Les diferents mesures s'han calculat en Python, i en el cas de les mesures de teoria de grafs s'han calculat amb el suport de la llibreria NetworkX.

Amb aquestes mesures conjuntament amb mesures mèdiques, com ara el grup sanguini i el panel-reactive antibody (PRA), s'ha modelat un model de

regressió logística. El model resultant és un model que prediu la probabilitat d'una parella pacient-donant d'acabar sent seleccionada per un trasplantament i les mesures que utilitza són: el grau de sortida, Closness Centrality, Betweenness Centrality, el nombre de donants amb grup sanguini B i la sorpresa entròpica. El model s'ha calculat usant R i s'ha utilitzat la llibreria rpy2 per poder fer les prediccions en Python.

La predicció del model conjuntament amb algunes de les mesures es poden visualitzar en el graf amb els colors dels nodes. També hi ha un apartat d'estadístiques a on es poden visualitzar aquestes mesures i algunes dades mèdiques en forma d'histograma.

L'aplicació web té un sistema d'autenticació d'usuaris i aquests poden guardar i gestionar un conjunt de fitxer JSON amb la informació dels pacients, dels donants, de les compatibilitats entre ells i els possibles cicles a realitzar.

Agraïments

Per començar vull agrair molt especialment als tutors del meu projecte final de grau Francesc Castro i Anton Bardera, pel seu acompanyament i ajuda constant durant el desenvolupament del projecte.

També vull agrair la meva família pel seu suport incondicional.

Índex

| | | |
|----------|--|-----------|
| 1 | Introducció | 1 |
| 1.1 | Objectius | 2 |
| 2 | Viabilitat | 3 |
| 3 | Metodologia | 5 |
| 4 | Planificació | 7 |
| 5 | Marc de treball i conceptes previ | 9 |
| 5.1 | Trasplantament renal | 9 |
| 5.2 | Teoria de grafs | 13 |
| 5.2.1 | Degree Centrality | 14 |
| 5.2.2 | Eigenvector Centrality | 14 |
| 5.2.3 | Closeness Centrality | 14 |
| 5.2.4 | Betweenness Centrality | 14 |
| 5.2.5 | Trophic levels | 15 |
| 5.3 | Teoria de la informació | 15 |
| 5.3.1 | Probabilitat estacionaria | 15 |
| 5.3.2 | Entropia i sorpresa entròpica | 16 |
| 5.3.3 | Informació mútua | 16 |
| 5.4 | Regressió logística | 17 |
| 5.5 | Visualització de les dades | 18 |
| 6 | Requisits del sistema | 19 |
| 6.1 | Requisits | 20 |
| 7 | Estudi i decisions | 23 |
| 7.1 | Framework | 23 |
| 7.1.1 | Django | 23 |
| 7.2 | Visualització del graf | 25 |
| 7.3 | Mesures del graf | 25 |
| 7.3.1 | Regressió logística | 26 |
| 8 | Anàlisi i disseny del sistema | 27 |
| 8.1 | Model de dades | 27 |
| 8.1.1 | Fitxer de pacients | 27 |
| 8.1.2 | Fitxer de donants | 28 |

| | | |
|-----------|---|-----------|
| 8.1.3 | Fitxer de compatibilitats | 28 |
| 8.1.4 | Fitxer de cicles | 29 |
| 8.2 | Interfície d'usuari | 30 |
| 8.3 | Diagrama i fitxes de casos d'ús | 30 |
| 9 | Implementació i proves | 35 |
| 9.1 | Aplicació web | 35 |
| 9.2 | Visualització del graf | 38 |
| 9.2.1 | Graf de compatibilitats | 38 |
| 9.2.2 | Graf de cicles | 43 |
| 9.2.3 | Estadístiques | 45 |
| 9.3 | Mesures del graf | 46 |
| 9.4 | Regressió logística | 50 |
| 10 | Implantació i resultats | 53 |
| 10.1 | Regressió logística | 53 |
| 10.2 | Aplicació web | 62 |
| 11 | Conclusions | 65 |
| 12 | Treball futur | 67 |
| | Bibliografia | 69 |

CAPÍTOL 1

Introducció

Els Kidney Exchange Programs (KEP) gestionen els procediments de trasplantaments de ronyó creuats amb donant viu, on un donant vinculat a un pacient ofereix un ronyó a canvi que el “seu” pacient en rebí un provinent d’un altre donant compatible.

Un trasplantament de ronyó és un tractament quirúrgic que consisteix a implantar un ronyó sa d’un donant viu o mort en un pacient amb malaltia renal crònica en fase final. L’objectiu del trasplantament és, entre altres coses, restaurar les funcions renals, netejar toxines de la sang, crear un equilibri hidroelectrolític (equilibri entre la quantitat d’aigua i minerals), promoure la síntesi hormonal i evitar així la diàlisi. En una donació amb donant viu, el ronyó prové d’una persona sana (membre de la família o amic) que dona voluntàriament l’òrgan a un ésser estimat. Es realitzen diversos estudis mèdics sobre el donant i el receptor abans de la donació per confirmar la viabilitat de la donació i la compatibilitat entre ells. Els trasplantaments de ronyó es realitzen en pacients amb insuficiència renal quan els ronyons no estan fent bé la seva feina, cosa que provoca que el cos acumuli aigua i residus.

Els trasplantaments de ronyó amb donant viu han anat en augment durant els últims anys en la majoria de països del primer món, ja que presenten diferents avantatges enfront dels trasplantaments amb donant cadàver:

- Major taxa de supervivència.
- Reducció del temps d’espera.
- Les darreres tècniques suposen un baix risc en cas de donant viu.

El primer punt es deu al fet que l’òrgan té una "qualitat exquisida", perquè qui el dona és una persona sana i a més perquè l’operació es pot planificar al detall, cosa inviable quan l’òrgan prové d’una persona morta.

Aquest treball es desenvolupa en un marc de col·laboració entre una sèrie de professors de l’àrea de Llenguatges i Sistemes Informàtics (LSI), entre els quals hi ha els tutors del treball, en Francesc Castro i l’Anton Bardera, i l’Organización Nacional de Transplantes (ONT). L’objectiu bàsic és el de desenvolupar una eina per visualitzar i analitzar els possibles trasplantaments, que pugui ser utilitzada pel personal mèdic en els diferents programes KEP.

1.1 Objectius

De forma més detallada, els objectius d'aquest projecte són:

- Desenvolupar una eina de visualització i anàlisi en format de pàgina web, que permeti facilitar al personal mèdic la comprensió del pool de dades de donants i receptors amb les seves relacions de compatibilitat i els cicles de trasplantament a priori viables.

Dins d'aquest objectiu més general, es va dividir en un conjunt de subobjectius:

- Desenvolupar la visualització dels possibles trasplantaments en forma de graf.
 - Desenvolupar la visualització dels possibles cicles de trasplantaments dins del graf.
 - Crear interaccions amb els nodes del graf per poder visualitzar en detall la informació d'una parella pacient-donant i els possibles trasplantaments a on podrien participar.
 - Crear interaccions amb les arestes per poder visualitzar en detall els possibles trasplantaments.
 - Permetre visualitzar en el graf mesures mèdiques, de teoria de la informació i de teoria de grafs.
 - Crear una aplicació web a on incorporar la visualització del graf que permeti tenir usuaris i que aquests puguin guardar-hi fitxers JSON per visualitzar-los.
- Analitzar el pool de dades mitjançant mesures de teoria de grafs i teoria de la informació i utilitzar aquestes mesures per realitzar un model de regressió logística que permeti predir la probabilitat d'un pacient d'acabar sent seleccionat per a un trasplantament.

Aquest objectiu es va dividir en el següent conjunt de subobjectius:

- Aplicar diferents mesures de Teoria de Grafs en el graf i analitzar la seva rellevància en modelar el model de regressió logística.
- Aplicar diferents mesures de Teoria de la informació en el graf i analitzar la seva rellevància en modelar el model de regressió logística.
- Modelar el model de regressió logística per predir la probabilitat d'una parella pacient-donant de ser escollida en un dels cicles de trasplantament.

CAPÍTOL 2

Viabilitat

Aquest projecte s'emmarca en un conveni de col·laboració amb l'ONT en el qual participa la UdG, per tant no hi ha cap pressupost establert.

Per a la realització del projecte no hi ha requisits de hardware més enllà d'un ordinador amb accés a Internet (s'ha utilitzat el de l'autor del projecte). Pel que fa al software, s'han fet servir les següents eines:

- Per al desenvolupament de l'aplicació web, s'ha utilitzat programari lliure com ara Django (web framework en Python) i D3.js (Llibreria de JavaScript per a la visualització de dades).
- Per a l'anàlisi estadístic, s'ha utilitzat programari lliure com R i NetworkX (Llibreria de Python per a l'anàlisi de grafs).
- Per al desenvolupament del software, s'han utilitzat els IDE de JetBrains, concretament PyCharm per al desenvolupament en Python i IntelliJ IDEA per a la pàgina web. Gràcies al fet que l'autor és un estudiant, l'accés a aquests IDE ha estat gratuït. També s'ha utilitzat R-Studio per al treball amb R.

La utilització de D3.js per la visualització de les dades en forma de graf suposa una limitació pel que fa al nombre de nodes i arestes que es poden visualitzar (aproximadament un límit de 1000 nodes). En el nostre cas, com que és una aplicació web pensada per ser implementada a l'ONT i aquesta organització treballa en l'àmbit espanyol sabem que el nombre de parelles pacient-donant no superarà aquests límit. Si en un futur es volen unir els conjunts de parelles de pacients amb altres països, es podria arribar a superar el límit.

Un altre element necessari per al desenvolupament del projecte són les dades reals (anonimitzades) dels pools de pacients i donants. Gràcies al citat conveni entre la UdG i l'ONT s'ha pogut accedir a aquestes dades de forma totalment gratuïta.

Estimarem el cost d'inversió que ha suposat el desenvolupament d'aquest programari. Per fer-ho ens basarem amb el que determina el vigent Conveni Col·lectiu d'empreses de consultoria, estudis i informàtica estatal, que fixa una jornada laboral anual de 1.800 hores.

Un sou brut anual realista per un programador júnior seria 22.000,00 € bruts a l'any, als quals s'hi hauria de sumar un 31,5 per cent anual en concepte de despeses de seguretat social de l'empresa, la qual cosa suposa una despesa anual de 28.930,00 €, als quals hi hauríem d'afegir les despeses indirectes que serien imputables a la feina (amortització dels equips de treball bàsicament), i les despeses generals de l'empresa o organització (local, llum, assegurances, personal de suport, etc.).

En empreses de consultoria el coeficient K que multiplica les despeses laborals acostuma a ser superior a 2,0 però en el nostre cas suposarem una empresa de menors dimensions, per la qual cosa seria més realista un coeficient proper a 1,5.

Aplicant aquests supòsits ens surt un cost hora d'uns 24,00 €. El desenvolupament ha portat 400 hores de feina, la qual cosa suposa un cost d'inversió de 9.600,00 €.

El desenvolupament ha estat realitzat per l'estudiant, fet que suposa un cost "nul" perquè no ha cobrat pel desenvolupament de projecte.

La implantació de l'aplicació web en un servidor tindria un preu aproximat de 40€ mensuals. Però en utilitzar els mateixos servidors de l'ONT, aquests costos es redueixen significativament.

En conseqüència, podem dir que els costos econòmics que suposa el desenvolupament del projecte són nuls, i per tant, el projecte és viable.

CAPÍTULO 3

Metodología

Per al desenvolupament d'aquest projecte s'ha utilitzat una metodologia àgil, de l'estil de la metodologia Scrum.

Scrum és una metodologia àgil i col·laborativa apropiada per a desenvolupament en equips d'entre 3 i 9 persones que es reparteixen la feina. Es fan lliurament parcials periòdicament, de forma que el client veu com avança el projecte.

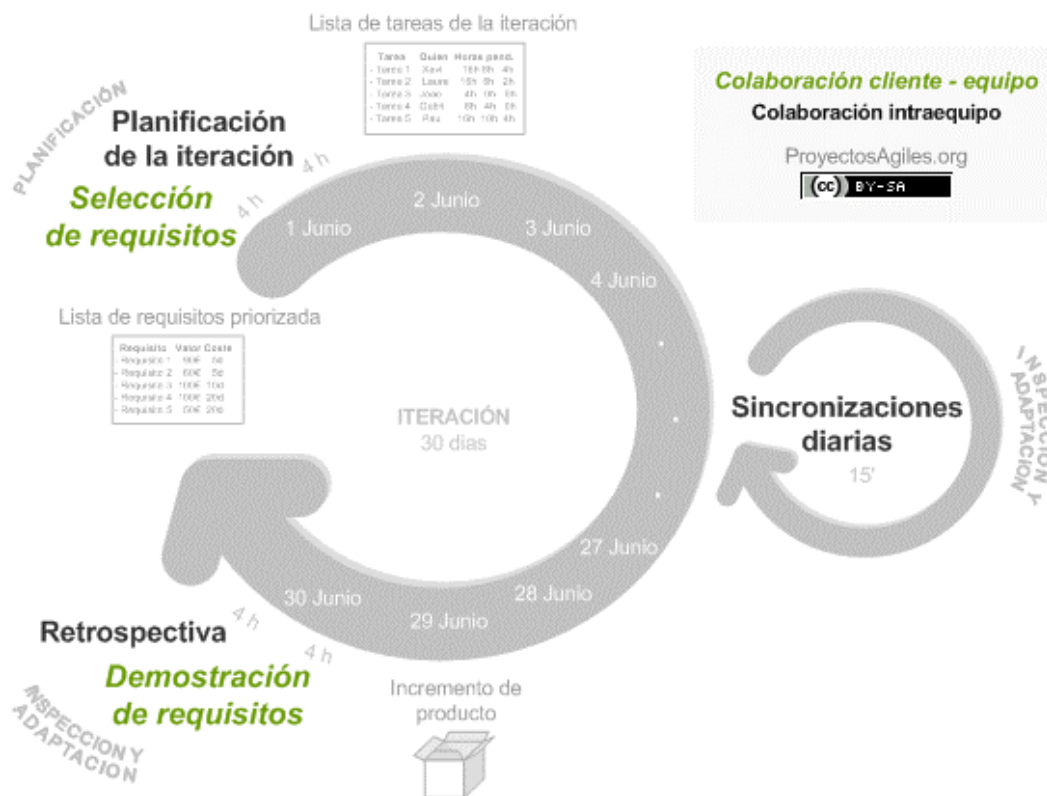


Figura 3.1: Esquema metodològic Scrum

La metodologia Scrum consisteix a organitzar el projecte en diferents cicles temporals a on es reparteixen les diferents tasques de forma equilibrada. Al final del cicle, es posa en comú tota la feina realitzada i es comprova si s'acompleixen els requisits establerts a l'inici del cicle.

No puc dir que hagi emprat la metodologia Scrum perquè, en ser un treball individual, no ha calgut realitzar cap repartiment de tasques ni posar en comú la feina feta al final de cada cicle. A l'inici del projecte, ens vam marcar uns objectius que volíem assolir a l'acabament i, durant el desenvolupament, s'han dut a terme reunions setmanals amb els professors tutors, a les quals s'ha analitzat la feina feta i s'han programat les tasques a fer per a la següent setmana.

Les reunions setmanals han sigut reunions curtes (aproximadament 30 min) i han tingut la següent dinàmica:

- Primer miràvem quina feina s'havia d'haver realitzat durant la setmana i es posava en comú la feina feta, els dubtes i problemes que havien sorgit i si feia falta, es buscaven solucions als possibles problemes apareguts.
- Seguidament es comentava la feina acabada i les conclusions que se'n podien treure.
- I per acabar, s'analitzava la feina que quedava per fer i s'assignaven les tasques a realitzar durant la següent setmana.

Durant la setmana, es duïen a terme les tasques assignades i en cas de sorgir dubtes es contactava amb els professors tutors.

CAPÍTOL 4

Planificació

Inicialment es van repartir les tasques i subtasques a realitzar, dins el projecte, durant un període de 40 setmanes.

Les tasques a realitzar eren:

- Anàlisi de requeriments (2 setmanes).
- Desenvolupament de la visualització (15 setmanes):
 - Creació graf compatibilitats D3.js (5 setmanes).
 - Interacció amb els nodes (2 setmanes).
 - Interacció amb les arestes (2 setmanes).
 - Creació graf de cicles (4 setmanes).
 - Creació apartat estadístiques (2 setmanes).
- Creació aplicació web amb usuaris (4 setmanes).
- Anàlisi del graf amb mesures de Teoria de Grafs (4 setmanes).
- Anàlisi del graf amb mesures de Teoria de la Informació (4 setmanes).
- Creació model de regressió logística (4 setmanes).
- Elaboració de la documentació i la presentació del treball (7 setmanes).

La planificació inicial s'ha anat seguint durant el desenvolupament de projecte amb alguna variació. El desenvolupament de la interacció amb els nodes va allargar-se fins a una duració de 4 setmanes, i en canvi l'elaboració de la documentació i la presentació del treball s'han realitzat en 5 setmanes.

Marc de treball i conceptes previ

El projecte s'emmarca en diferents línies de treball que convergeixen en un objectiu comú.

En primer lloc, parlarem dels trasplantaments renals. Analitzarem les necessitats i els avantatges i inconvenients que comporten els diferents tipus de trasplantaments, fent èmfasi en el que és l'objectiu d'aquest projecte, el trasplantament creuat amb donant viu.

En segon lloc, farem una introducció a la Teoria de Grafs, emfatitzant en les mesures de Teoria de Grafs que puguin tenir una major aplicabilitat al nostre treball.

En tercer lloc, parlarem de la Teoria de la Informació, i presentarem certs aspectes d'aquesta teoria que poden tenir utilitat per al nostre projecte.

En quart lloc descriurem com funcionen els models de regressió logística, els quals també seran utilitzats en la fase d'implementació i proves per tal d'analitzar quines variables són més útils per preveure la possible inclusió d'un pacient en un intercanvi de ronyons.

Finalment i en cinquè lloc, explicarem quins mètodes de visualització de dades han estat aplicats al projecte per tal d'explicar, de forma gràfica, les relacions de compatibilitat potencial entre donants i pacients així com els possibles cicles i cadenes de trasplantament.

5.1 Trasplantament renal

Moltes persones, arreu del món, pateixen insuficiència renal (és a dir, una disfunció important que afecta als seus dos ronyons). Aquesta situació empitjora molt la seva qualitat de vida, fent-los totalment dependents dels mecanismes de diàlisi, i minva també la seva esperança de vida.

En aquests casos, rebre un ronyó trasplantat s'ha mostrat, des de ja fa molts anys, com a una solució bastant efectiva. En general (els procediments varien en funció del país), quan una persona necessita un trasplantament, s'apunta a una llista d'espera per rebre un ronyó d'un donant compatible. En aquests casos, normalment es tracta d'un donant cadàver: quan una persona donant mor, els òrgans que estan en bon estat es trasplanten a un pacient compatible de la llista d'espera.

En el cas dels trasplantaments de ronyó, aquesta no és l'única via, ja que també és possible realitzar trasplantaments amb un donant viu. El pacient pot tenir una (o més) persones properes que ofereixin un ronyó -atès que es pot viure perfectament amb només un d'aquests òrgans- al seu germà, parella, pare, fill, amic, etc. Els trasplantaments amb donant viu tenen diversos avantatges envers els trasplantaments amb donants cadàver i amb el temps han guanyat força, especialment en determinats països arreu del món:

- Millors resultats: la bona salut del donant i la possibilitat de fer totes les proves de compatibilitat abans del trasplantament ajuden a una major compatibilitat HLA (entre els antígens del donant i els anticossos del receptor) i a augmentar la supervivència del pacient. El fet de tenir un donant viu, evita els efectes secundaris que pateix el ronyó a causa de la mort del donant i redueix el temps que l'òrgan no rep flux sanguini arterial (temps que no pot ser superior a 24-30 hores). A més en el cas de donant viu, aquest pot tenir una edat més semblant a la del pacient, fet difícil d'aconseguir amb els donants cadàver, ja que solen ser persones grans.
- Millora en la seguretat del donant: amb el temps s'ha anat millorant les tècniques per l'extracció poc invasiva, fet que fa que els donants tinguin un baix índex de complicacions.
- Avanços científics: s'han solucionat problemes de compatibilitats de pacients i donants mitjançant tècniques de dessensibilització (reducció d'anticossos) que permeten la realització d'un major nombre de trasplantaments.

En el cas dels trasplantaments amb donant viu, si hi ha la sort que la persona que s'ha ofert com a donant és compatible amb el potencial receptor, el trasplantament tira endavant. Tanmateix, en moltes ocasions, es produeixen incompatibilitats que fan a priori impossible la planificació de l'operació. En general es tracta o d'incompatibilitats a nivell de grup sanguini i/o d'antígens del donant vs. anticossos del receptor, tot i que també poden haver-hi altres inconvenients com ara la diferència d'edat.

En aquests casos en què es presenten incompatibilitats, i des de fa al voltant de dues dècades aproximadament, nombrosos estats han implantat programes de trasplantament creuat, als quals el pacient i el(s) "seu(s)" donant(s) entren en el pool d'espera del corresponent programa de trasplantament creuat amb l'esperança de trobar una altra parella donant-pacient en què sigui possible fer un trasplantament creuat, el que sovint s'anomena un cicle de trasplantament de mida 2. No és difícil estendre aquesta idea a cicles de mida 3 i inclús de mida

superior, tot i que a major mida la logística es va complicant molt, ja que és necessari un alt nivell d'organització per realitzar els trasplantaments de forma simultània, per tal d'evitar que alguna de les parelles es desdigni enmig del procés. Cal notar que a l'estat espanyol l'ONT només contempla, de moment, cicles de mida 2 i 3.

D'altra banda, existeixen els anomenats donants altruistes o samaritans, que ofereixen un seu ronyó sense tenir cap vincle amb cap potencial receptor. Aquesta mena de donants permeten realitzar cadenes de trasplantaments, ja que poden donar el seu ronyó a un pacient del pool de trasplantament creuat, i llavors el donant vinculat al pacient trasplantat dona el seu ronyó a un nou pacient del pool i així successivament. Es poden planificar cadenes de major llargada que els cicles, i no és tan imprescindible la simultaneïtat de les operacions, ja que en aquest cas cap pacient pot quedar "penjat" degut a alguna renúncia. El donant del darrer receptor de la cadena pot assolir un rol de nou donant altruista o bé, amb més immediatesa, donar el seu ronyó al pool de pacients que esperen un donant cadàver. Cal dir, però, que a l'estat espanyol el nombre de donants altruistes és reduït (unes poques unitats per any), en comparació amb altres llocs com ara el Regne Unit o els Països Baixos.

Concretant, podem dir que al cas espanyol (ONT) es realitzen 3 tipus diferents de trasplantaments creuats:

- Intercanvi (cicle) de mida 2. Es tracta del procediment amb una logística més simple: un trasplantament creuat a on hi intervenen 2 parelles pacient-donant (A i B), on el pacient de la parella A rep del donant B i el pacient B rep del donant A. Aquests són els trasplantaments que necessiten menys recursos per realitzar-se.

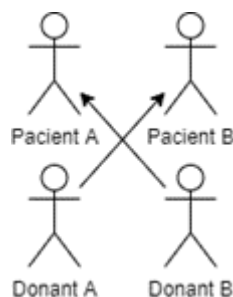


Figura 5.1: Esquema gràfic dels cicles de mida 2

- Intercanvi (cicle) de mida 3. En aquest cas hi ha implicades 3 parelles pacient-donant (A, B i C), on el donant A donaria al pacient B, el donant B donaria al pacient C i el donant C donaria al pacient A. Aquest intercanvi és més difícil de realitzar que el de mida 2, ja que requereix més recursos (3 quiròfans, per exemple).

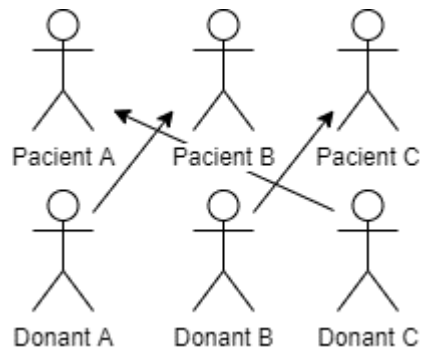


Figura 5.2: Esquema gràfic dels cicles de mida 3

- Cadena iniciada per un donant altruista. En aquest cas, el ronyó del donant altruista és trasplantat al receptor d'una parella (diguem-ne A), el donant A aportaria el seu ronyó al pacient d'una altra parella (diguem-ne B) i així successivament fins arribar a un punt en què la cadena quedi interrompuda i resti un donant pendent, el qual podrà romandre al pool com a nou donant altruista per iniciar una nova cadena o bé, per tal d'evitar que tinguin temps de desdir-se, se'l pot fer donar el seu ronyó de forma immediata a algun receptor de la llista d'espera de donant cadàver.

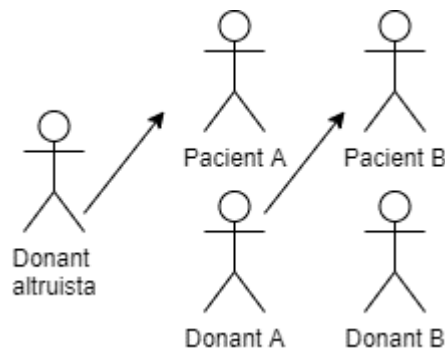


Figura 5.3: Esquema gràfic de una cadena de trasplantaments

Per tal de planificar els cicles d'intercanvi i les cadenes descrits en els anteriors punts, el procediment bàsic consisteix en optimitzar una puntuació o *score* associada a cada conjunt de cicles i/o cadenes. Aquesta puntuació és calculada a base de sumar la puntuació corresponent a cada cicle (o cadena), la qual a la vegada s'obté sumant les puntuacions de cada trasplantament potencial contingut en el cicle (o cadena).

Per tant, a cada possible trasplantament viable (és a dir, teòricament compatible) se li assigna a priori una puntuació, la qual s'obté a base d'una sèrie de criteris, bàsicament mèdics, entre els que es troben, per exemple, el nivell

d'anticossos del receptor, els grups sanguinis de receptor i donant o el temps que porta el receptor a la llista d'espera. Cal dir que el càlcul de la puntuació es realitza a partir d'una fórmula que, al cas de l'estat espanyol, ha estat aprovada per un comitè i que no pot ser modificada sense l'acord del comitè en qüestió. En altres paraules, a efectes d'aquest projecte, aquests *scores* han de ser considerats com a un simple *input*.

D'altra banda, cal remarcar que, tal i com s'ha indicat per omissió, el càlcul de la combinació òptima de cicles i/o cadenes que seran seleccionats per programar trasplantaments queda fora dels objectius d'aquest projecte.

5.2 Teoria de grafs

Un graf descriu la relació que hi ha entre un conjunt de nodes mitjançant arestes. Cada node d'un graf representa quelcom (una ciutat, una estació, etc.) i cada aresta representa la relació entre dos nodes (una carretera, etc.).

Hi ha diferents tipus de grafs segons el tipus d'arestes que tenen. El graf més bàsic és el graf no dirigit, en aquest tipus les arestes només representen l'existència d'una relació (amb un pes) entre dos nodes. Els grafs més comuns són els grafs dirigits, aquests tipus de graf té arestes que representen un enllaç entre dos nodes en una direcció en concret amb un pes. Per últim tenim els multigrafs dirigits, aquests són grafs dirigits que poden tenir més d'una aresta amb la mateixa direcció entre dos nodes.

En el nostre cas, cada node representa un pacient i els seus donants vinculats i cada aresta representa un possible trasplantament. Cada aresta té un pes que representarà la puntuació i una direcció que representa en quin sentit seria el trasplantament (donant -> pacient). La representació del pool de dades està en format d'un graf multi dirigit, ja que les arestes tenen una direcció i ens podem trobar en el cas que dos nodes tinguin més d'una aresta en la mateixa direcció, ja que pot haver-hi més d'un donant per pacient.

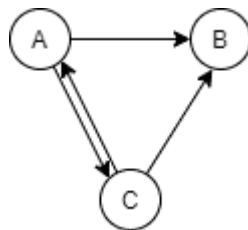


Figura 5.4: Exemple de graf dirigit

En un graf, podem calcular diferents mesures teòriques per analitzar la relació entre els nodes. Per facilitar el càlcul d'aquestes mesures, en alguns casos

representarem el graf en forma de graf dirigit en lloc de multi dirigit. Per fer-ho només tindrem en compte l'aresta amb pes més gran entre dos nodes en cas d'haver-hi més d'un donant.

Les mesures de teoria de grafs que hem utilitzat són mesures de centralitat. Aquestes mesures permeten definir la centralitat d'un node respecte al graf.

El concepte de centralitat es refereix a la importància o prominència dels nodes dins d'un graf. Hi ha diverses mesures o índexs de centralitat per determinar i comparar quantitativament la importància relativa d'un node dins de l'estructura definida pel graf.

La nostra hipòtesis és que un node com més central és, més probabilitats té de ser escollit per un trasplantament el pacient corresponent a aquest node.

5.2.1 Degree Centrality

La primera mesura de centralitat que hem utilitzat és el grau d'un node. Podem definir el grau com el número d'arestes d'un node. En els grafs dirigits, podem diferenciar el grau d'entrada (indegree), a on es valora el número d'arestes que entren en el node (en el nostre cas són el número de possibles trasplantaments a on el pacient del node rep un ronyó) i el grau de sortida (outdegree) a on es valora el número d'arestes que surten del node.

Totes aquestes mesures es poden calcular tenint en compte la puntuació de l'aresta o no.

5.2.2 Eigenvector Centrality

L'eigenvector és una mesura de la importància d'un node en relació amb el graf que pertany. Es considera important un node en el graf segons les seves arestes, el pes d'aquestes i la importància dels nodes als quals van dirigides. És una mesura que es basa en el càlcul del vector propi referent al valor propi més gran possible.

5.2.3 Closeness Centrality

És una mesura de centralitat dels nodes dels grafs a on es mesura la mida del camí més curt entre un node amb tots els nodes del graf. Es pot calcular tenint en compte el pes de les arestes o no.

5.2.4 Betweenness Centrality

És una mesura de centralitat dels nodes del graf que es defineix com el número de camins que passen per un node a l'hora de calcular tots els camins mínims

entre dos nodes del graf. Igual que algunes de les mesures anteriors, es pot tenir en compte el pes de les arestes o no.

5.2.5 Trophic levels

És una mesura de centralitat per grafs dirigits a on s'ordena el graf de manera que les arestes tinguin una direcció semblant, intentant que els nodes amb només arestes de sortida quedin a un costat i els nodes amb només arestes d'entrada a l'altre. Un cop ordenat el graf, a cada node se li assigna un nivell segons la nova posició. Aquesta mesura s'inspira en les cadenes tròfiques que s'observen a la natura.

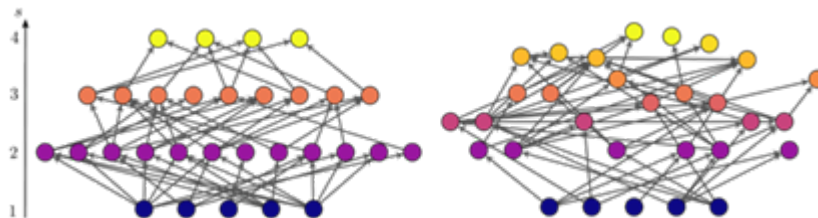


Figura 5.5: Exemple gràfic de la mesura Trophic Levels

5.3 Teoria de la informació

Les mesures de Teoria de la informació que hem utilitzat calculen la probabilitat en diferents escenaris d'estar en un node determinat. Són mesures que modelen un passeig aleatori (random walk) del graf. També hi ha mesures del graf a on es calcula la interconnectivitat del graf.

Un dels elements bàsics per calcular aquestes mesures, són les probabilitats de transició entre els nodes. És a dir, si en un moment t estic en un node, quina probabilitat tinc d'estar en un altre node en el moment $t+1$. Per calcular aquestes probabilitats, s'utilitza la matriu d'adjacències del graf normalitzada de manera que cada fila suma un total d'1.

5.3.1 Probabilitat estacionària

La probabilitat estacionària, és la probabilitat d'estar en un node i es caracteritza en la seva invariabilitat en el temps. Aquesta probabilitat estacionària correspondrà al vector propi amb valor propi igual a 1 de la matriu de transició d'estats transposada, ja que la probabilitat estacionària en un moment t ha de ser la mateixa que la probabilitat en el moment $t + 1$.

5.3.2 Entropia i sorpresa entròpica

A partir de la probabilitat estacionaria es pot calcular l'entropia del graf, què és la magnitud que mesura la informació continguda en un flux de dades, és a dir la informació que ens aporta:

$$H(X) = - \sum_{i=1}^N p_s(x_i) \log(p_s(x_i)). \quad (5.1)$$

A on $p_s(x_i)$ és la probabilitat estacionaria del node i .

La contribució de cada un dels nodes a l'entropia ve donat de manera natural per la sorpresa entròpica, que es pot mesurar de dos formes diferents:

$$E(x_i) = -\log p_s(x_i) \quad (5.2)$$

I també es pot multiplicar per la probabilitat estacionaria del node:

$$E(x_i) = -\log p_s(x_i) * p_s \quad (5.3)$$

5.3.3 Informació mútua

La informació mútua entre dues variables X i Y es defineix com:

$$I(X; Y) = H(X) - H(X; Y) \quad (5.4)$$

$$= \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (5.5)$$

$$= \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log \frac{p(y|x)}{p(y)} \quad (5.6)$$

$$= \sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log \frac{p(x|y)}{p(x)} \quad (5.7)$$

A on X és la probabilitat d'estar en un node en temps t i Y en el temps $t+1$. A partir d'aquesta mesura podem proposar 4 descomposicions:

$$I_1(y_i; X) = \sum_{x \in X} p(x|y_i) \log\left(\frac{p(x|y_i)}{p(x)}\right) \quad (5.8)$$

$$I_2(y_i; X) = p(x) \sum_{x \in X} p(x|y_i) \log\left(\frac{p(x|y_i)}{p(x)}\right) \quad (5.9)$$

$$I_3(y_i; X) = \sum_{x \in X} p(x|y_i) \log(p(x|y_i)) \quad (5.10)$$

$$I_4(y_i; X) = p(x) \sum_{x \in X} p(x|y_i) \log(p(x|y_i)) \quad (5.11)$$

A on $p(x|y)$ és la matriu d'adjacències normalitzada i $p_s(x_i)$ és la probabilitat estacionària.

5.4 Regressió logística

Un dels objectius del nostre projecte, és crear un model de regressió logística que ens permeti predir quines parelles pacient-donant acabaran essent escollides en la solució.

La regressió logística és l'anàlisi de regressió adequat per dur a terme quan la variable dependent és binària. Com totes les anàlisis de regressió, la regressió logística és un anàlisi predictiu. La regressió logística s'utilitza per descriure dades i explicar la relació entre una variable binària dependent i una o més variables independents. Els models de regressió logística pertanyen al conjunt de models lineals generalitzats (GLM) que utilitzen la funció *logit*.

La regressió logística pot ser utilitzada per diferents motius:

- Previsió dels efectes o de l'impacte de canvis específics. Per exemple, si una empresa vol pronosticar quantes unitats d'un producte concret han de produir per satisfer la demanda actual.
- Previsió de tendències i valors futurs. Per exemple, quin serà el preu de les accions d'una empresa d'aquí a sis mesos?
- Determinar la importància de diferents mesures, o, en altres paraules, avaluació de l'impacte que tenen les variables independents en una variable dependent. Per exemple si una empresa de refrescos patrocina un partit de futbol, podrà determinar si els anuncis que es mostren durant el partit han suposat un augment de les vendes.

En el nostre cas, utilitzem aquest model per determinar quines de les mesures són més importants. La variable categòrica a predir en el nostre model és si la parella pacient-donant pertany a la solució (conjunt de trasplantaments que es realitzaran).

5.5 Visualització de les dades

Un dels objectius del projecte és la realització d'una aplicació web a on es pot visualitzar els possibles trasplantaments en forma de graf i alguna de les mesures explicades a l'apartat anterior.

Per fer-ho s'ha utilitzat la llibreria de JavaScript D3.js, que permet visualitzar grafs i el framework Django que permet la creació d'una aplicació web amb usuaris on aquest poden guardar diferents fitxers són amb les dades dels pacients.

Requisits del sistema

El sistema que s'ha implementat en aquest projecte té, com a finalitats principals:

- Calcular diferents mesures de teoria de grafs sobre el graf de compatibilitats entre donants i pacients
- Utilitzar les mesures calculades anteriorment per fer un model de regressió logística per predir si una parella pacient-donant serà escollida per algun trasplantament.
- Desenvolupament d'una aplicació web que permeti visualitzar les dades de diferents pools de pacients

La informació sobre els pacients i donants que hi ha al pool i els possibles intercanvis que es poden realitzar, es troba en 4 fitxers JSON:

- Un fitxer de pacients a on hi ha la informació detallada dels pacients
- Un fitxer de donants a on hi ha la informació detallada dels donants i a quin pacient estan vinculats
- Un fitxer de compatibilitats a on hi ha tots els possibles trasplantaments i el seu score.
- Un fitxer de cicles a on hi ha un conjunt de trasplantaments organitzats en cicles de 3, cicles de 2 o cadenes. Aquest fitxer de cicles pot ser de tots els cicles possibles del pool o pot ser un fitxer de solució a on es trobaria només els trasplantaments escollits per realitzar.

Es desglossen els requisits del sistema en dos grups: requisits funcionals (F) i requisits no funcional (NF): els requisits funcionals són aquelles funcionalitats que el programa ha de tenir i els requisits no funcionals fan referència a com ha de ser el programa.

| Num. | Titol |
|------|------------|
| | Descripcio |

6.1 Requisits

| | |
|----|--|
| F1 | Importar fitxer JSON |
| | Permet importar un fitxer JSON, guardar-lo organitzadament segons les dades que conté i l'usuari que l'ha penjat |

| | |
|----|--|
| F2 | Llistar els fitxers JSON |
| | Permet als usuaris llistar els fitxer JSON que han penjat anteriorment |

| | |
|----|-------------------------------|
| F3 | Registrar usuaris |
| | Permet registrar usuaris nous |

| | |
|----|---|
| F8 | Canviar la contrasenya |
| | Permet als usuaris canviar la contrasenya |

| | |
|-----|---|
| NF1 | Menú |
| | Hi ha d'haver un menú que permeti escollir que es vol fer (visualitzar, accions amb fitxers, etc.). |

| | |
|-----|--|
| NF2 | Menú d'usuari |
| | Hi ha d'haver un menú per escollir opcions de l'usuari (logout, canvi de contrasenya). |

| | |
|----|--------------------------------|
| F4 | Identificar usuaris |
| | Permet identificar els usuaris |

| | |
|----|-------------------------------------|
| F5 | Tancar sessió |
| | Permet als usuaris tancar al sessió |

| | |
|----|---|
| F6 | Visualitzar les dades |
| | Permet als usuaris escollir i visualitzar les dades |

| | |
|----|--|
| F7 | Eliminar fitxer JSON |
| | Permet als usuaris eliminar fitxers JSON |

Estudi i decisions

7.1 Framework

Per crear l'aplicació web, necessitàvem un framework que permetés tenir un sistema d'identificació d'usuaris senzilla a on aquest poguessin guardar fitxers.

Hi havia diferents frameworks que podríem haver fet servir, els 3 principals que vam estudiar si utilitzar van ser Django, React i Ember. Django és un framework en Python d'alt nivell que fomenta un ràpid desenvolupament i un disseny net. React en canvi, és una llibreria en JavaScript que permet renderitzar visualitzacions fàcilment. I per últim Ember.js és un framework en JavaScript que permet crear aplicacions web. D'aquests 3, vam acabar escollint Django perquè era l'únic que hi havia exemples de com guardar fitxers sencers a la base de dades de forma senzilla.

Django [Django 021], a diferència de les altres opcions, està desenvolupat en Python, fet que també ens ajudava a poder incorporar fàcilment les mesures de graf calculades amb la llibreria NetworkX de Python.

7.1.1 Django

Al crear un projecte a Django, automàticament es crea un conjunt de fitxer que et permet crear fàcilment l'aplicació web. Aquest projecte s'organitza en diferents carpetes segons els diferents objectes que hi ha. A cada projecte hi ha inicialment els següents fitxers:

I es pot afegir diferents directoris que realitzaran diferents funcionalitats d'un objecte, com per exemple directori "polls": **Manage.py**

El fitxer Manage.py és l'encarregat de gestionar diferents aspectes de l'aplicació web. Les principals funcionalitats són les següents:

- `manage.py runserver [port]`: Aquesta comanda executa el servidor en el port especificat.
- `manage.py makemigrations`: Aquesta comanda analitza el projecte i avalua quins canvis s'haurien de fer a les configuracions i a la base de dades.
- `manage.py migrate`: Realitza els canvis necessaris en el projecte perquè funcioni l'aplicació web (creats amb `makemigrations`)

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    asgi.py  
    wsgi.py
```

Figura 7.1: Estructura de fitxers d'un projecte de Django.

```
polls/  
  __init__.py  
  admin.py  
  apps.py  
  migrations/  
    __init__.py  
  models.py  
  tests.py  
  views.py
```

Figura 7.2: Estructura de fitxers d'un objecte de Django.

Urls.py

Aquest és al fitxer a on s'especifica les diferents URL de l'aplicació/directori i quina funció s'ha de cridar. Aquest fitxer també fa d'enllaç entre els diferents directoris del projecte. Per exemple:

```
path('accounts/', include('accounts.urls'))
```

Aquesta línia especifica que dins de l'aplicació, quan es fa un GET del path "/accounts", s'encarrega l'objecte accounts.

Views.py

Dins d'aquest fitxer es troba les diferents funcions a cridar des de Urls.py. Aquestes funcions retornen una resposta en format HTTP.

Models.py

Dins d'aquest fitxer es troba l'objecte amb els seus paràmetres i relacions amb altres objectes, per exemple en el cas de tenir el directori d'usuaris, dins

del fitxer `models.py` podríem tenir els paràmetres; nom, cognom, email, etc. I les característiques dels paràmetres (si és obligatori, si ha de tenir un format, etc.). També podríem tenir relacions com per exemple a quin grup formen part (a on grup és un objecte del nostre projecte).

7.2 Visualització del graf

Per realitzar la visualització del graf, s'ha utilitzat la llibreria de JavaScript D3.js [D3.js 020]. D3.js és una llibreria de codi obert i permet visualitzar dades en diferents formats, entre ells grafs i histogrames (utilitzats en aquest projecte). Hem escollit D3.js perquè permet personalitzar la visualització del graf i fer-la interactiva. En els darrers anys, aquesta llibreria ha esdevingut un estàndard en el desenvolupament de visualitzacions interactives a la web.

Aquesta llibreria permet:

- Visualitzar les dades en diferents formats (entre ells grafs i histogrames)
- Canviar les visualitzacions (canviar color, disseny, lletra, etc.)
- Afegir esdeveniments a realitzar per fer la visualització interactiva.

Per fer-ho, aquesta llibreria et permet seleccionar i modificar fàcilment elements del document html.

7.3 Mesures del graf

Per l'anàlisi del graf, necessitàvem una llibreria per facilitar el càlcul de mesures. Primer vam buscar llibreries en JavaScript per poder-ho incorporar fàcilment a la pàgina web, però ens vam trobar que hi havia poques opcions i aquestes no implementaven càlculs amb grafs dirigits. Buscant llibreries vam trobar referències a una llibreria en Python que tenia implementada el càlcul d'una gran quantitat de mesures de grafs per a grafs dirigits. Per aquest motiu, vam decidir utilitzar-la tot i estar en Python i posteriorment, quan vam escollir el framework a utilitzar pel desenvolupament de l'aplicació web, vam utilitzar un framework en Python per poder-ho incorporar.

La llibreria `networkX` [NetworkX 021], és una llibreria en Python que et permet crear objectes graf als quals es poden calcular diferents mesures. A l'hora de crear un graf, es pot escollir el tipus de graf (no dirigit, dirigit, multigraf dirigit, etc.). En el nostre cas escollim un graf dirigit:

$$G = nx.DiGraph()$$

i afegim els diferents nodes:

```
G.add_nodes_from(nodes)
```

i les arestes amb els seus pesos (en format de llista de tuples de 3 elements):

```
G.add_weighted_edges_from(arestes)
```

Un cop tenim el graf definit, es pot visualitzar el graf i es poden calcular diferents mesures fàcilment.

7.3.1 Regressió logística

Per fer la implementació del model, necessitàvem una eina que permetés crear un model de regressió logística ràpidament i poguéssim comparar diferents models. Per aquest motiu vam utilitzar R, ja que és el llenguatge utilitzat al grau per l'anàlisi de dades i vam veure que era possible realitzar-ho.

Posteriorment vam tenir la necessitat de fer prediccions amb el model definitiu en Python per poder-ho incorporar fàcilment a la pàgina web. Per fer-ho vam utilitzar la llibreria rpy2 [rpy2 021]. Aquesta llibreria et permet executar des de Python codi en R.

Anàlisi i disseny del sistema

8.1 Model de dades

Les dades necessàries per representar el graf es troben en 3 fitxers JSON diferents: un fitxer de pacients, un fitxer de donants i un fitxer de compatibilitats (a on hi ha els possibles trasplantaments). A més a més, per representar el graf de cicles, és necessari un altre fitxer JSON amb la informació d'aquests. Es pot trobar un exemple de cada fitxer en els annexos, a la carpeta *JSONExemple*.

8.1.1 Fitxer de pacients

El fitxer de pacients emmagatzema un llista de pacients amb tota la seva informació. A la capçalera del fitxer hi ha la data de creació, la versió i el número de pacients. Per cada pacient tenim:

- **INCORPORATION_DATE**: data d'incorporació del pacient al pool en format dd/mm/aaaa.
- **PATIENT_STATE**: indica si el pacient està actiu. En els fitxers que disposem de prova tots els pacients estan actius.
- **GENDER**: Indica el gènere del pacient: *Varon* o *Mujer*.
- **BIRTH_DATE**: data de naixement en format dd/mm/aaaa.
- **BLOOD_TYPE**: grup sanguini del pacient (0, A, B, AB).
- **PRA**: panel-reactive antibody (PRA). És una mesura mèdica donada en percentatge.
- **PRA_DATE**: data de la realització del test de PRA.
- **DIALYSIS_START**: data d'inici de diàlisi, en format dd/mm/aaaa. En cas de no estar en diàlisi el valor estarà buit.
- **MONTHS_DIALYSIS_STOP**: nombre de mesos entre la data d'inici de diàlisi i la data en què es va crear el fitxer que el pacient no ha necessitat diàlisi.

A part de la informació detallada anteriorment, per cada pacient tenim un conjunt de mesures mèdiques que no utilitzem en el projecte, però que s'utilitzen a l'hora de generar el fitxer de compatibilitats i el fitxer de solucions.

8.1.2 Fitxer de donants

El fitxer de donants emmagatzema una llista de pacients amb tota la seva informació. A la capçalera del fitxer hi ha la data de creació, la versió i el nombre de donants. Per cada donant tenim:

- **RELATED_PATIENT**: identificació del pacient al qual està vinculat.
- **INCORPORATION_DATE**: data d'incorporació del donant al pool en format dd/mm/aaaa.
- **GENDER**: indica el gènere del donant: *Varon* o *Mujer*.
- **BIRTH_DATE**: data de naixement en format dd/mm/aaaa.
- **BLOOD_TYPE**: grup sanguini del donant (O, A, B, AB).
- **ALTRUISTIC**: indica si el donant es altruista i per tant, no està vinculat a cap pacient.

A part de la informació detallada anteriorment, per cada donant tenim un conjunt de mesures mèdiques que no utilitzem en el projecte, però que s'utilitzen a l'hora de generar el fitxer de compatibilitats i el fitxer de solucions.

8.1.3 Fitxer de compatibilitats

El fitxer de compatibilitats, és un fitxer a on hi ha tots els possibles trasplantaments a realitzar amb aquest conjunt de pacients i donants. A la capçalera del fitxer hi ha indicat quins són els paràmetres que s'han tingut en compte a l'hora de calcular la puntuació de cada possible trasplantament. Aquest fitxer està organitzat en donants i per cada donant hi ha una llista dels possibles trasplantaments. Per cada possible trasplantament tenim la següent informació:

- **ID_PATIENT**: identificador del pacient que rebria el ronyó.
- **SCORE**: puntuació donada a aquest trasplantament.
- **SELFCOMPATIBLE**: indica si el pacient que rebria el ronyó, és compatible amb el seu propi donant.
- **REVERSIBLE**: indica si existeix un possible trasplantament entre el pacient vinculant d'aquest donant i el donant del pacient.

8.1.4 Fitxer de cicles

Hi ha dos tipus de fitxers de cicles, un primer que conté tots els possibles cicles dins d'un conjunt de pacients i donants (hi ha nodes que pertanyen a més d'un cicle) i un segon que conté una possible solució (no hi ha cap node que pertanyi a més d'un cicle).

Aquests fitxers estan organitzats en 3 diferents llistes de cicles:

- **twoSizedExchange**: Cicles de 2 trasplantaments.
- **threeSizedExchange**: Cicles de 3 trasplantaments.
- **chains**: Cadenes de trasplantaments (contenen un donant altruista).

Dins de cada una d'aquestes llistes hi ha la informació per cada possible cicle:

- **id**: identificador del cicle.
- **donors**: llista de donants.
- **recipients**: llista de pacients a rebre un ronyó.
- **score**: suma de les puntuacions dels trasplantaments a realitzar.
- **alternativeDonors**: indica si algun dels trasplantaments a realitzar disposa d'un altre possible donant en cas que no es pogués realitzar amb el donant actual.
- **selfCompatible**: indica si alguna de les parelles ja podia realitzar el trasplantament sense necessitat de buscar una altra parella.
- **paediatric**: indica si algun dels pacients es pediàtric.
- **hardPatient**: indica si algun dels pacients estava categoritzat com un pacient difícil.
- **backArc**: en el cas dels cicles de 3 o en les cadenes, indica si un dels possibles trasplantaments podria ser bidireccional.
- **freeDonor**: en el cas de ser una cadena, indica quin és el donant altruista.

8.2 Interfície d'usuari

La interfície de l'aplicació web és molt senzilla.

Consta de dos menús, un de lateral desplegable a on hi ha les principals funcionalitats i un desplegable sobre el nom de l'usuari per funcionalitats pròpies de l'usuari.

Cada usuari té el seu conjunt de fitxers i pot visualitzar-los dins de l'apartat *Graph*, penjar-ne de nous dins de l'apartat *Upload* i llistar els fitxers i eliminar-ne dins de l'apartat *Files*.

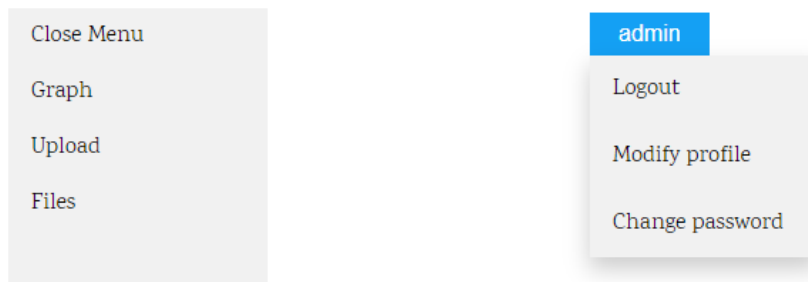


Figura 8.1: Menús de l'aplicació web

8.3 Diagrama i fitxes de casos d'ús

A la Figura 8.2, es mostra el diagrama de casos d'ús de l'aplicació web implementada. A continuació es detalla cada un dels casos d'ús:



Figura 8.2: Diagrama de casos d'ús

| | |
|----------------|---|
| Cas d'ús | Importar fitxers JSON |
| Descripció | L'usuari importa un fitxer JSON que conté totes les dades dels pacients |
| Actor | Metge de l'ONT |
| Precondició | El fitxer importat està en el format correcte |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari desplega el menú lateral 2. L'usuari clica sobre l'opció "Upload" 3. L'usuari clica sobre el botó "Tria un fitxer" 4. L'usuari va a buscar el fitxer que vol importar 5. L'usuari tria el tipus del fitxer (donants, pacients, compatibilitats o cicles) 6. L'usuari clica el boto d' "upload" 7. Si l'extensió del fitxer importat no és JSON <ol style="list-style-type: none"> 7.1 Mostra error 8. Guardar les dades a memòria 9. Es mostra un missatge conforme s'han llegit les dades correctament |
| Postcondició | S'han guardat les dades a memòria |

| | |
|----------------|---|
| Cas d'ús | Llistar fitxers JSON |
| Descripció | Es llisten tots els fitxers d'un usuari |
| Actor | Metge de l'ONT |
| Precondició | Cap |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari desplega el menú lateral 2. L'usuari clica sobre l'opció "Fitxers" 3. Es mostren els 5 primers fitxers d'aquell usuari 4. Si l'usuari escull un tipus de fitxer: <ol style="list-style-type: none"> 4.1 Es mostren els 5 primers fitxers del tipus seleccionat. 5. Si l'usuari vol veure més fitxers pot canviar de pàgina a la barra inferior. |
| Postcondició | S'han llistat els fitxers |

| | |
|----------------|--|
| Cas d'ús | Eliminar fitxers JSON |
| Descripció | Eliminar un fitxer JSON |
| Actor | Metge de l'ONT |
| Precondició | Cap |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari desplega el menú lateral 2. L'usuari clica sobre la opció "Fitxers" 3. Es mostren els 5 primers fitxers d'aquell usuari 4. Si l'usuari escull un tipus de fitxer: <ol style="list-style-type: none"> 4.1 Es mostren els 5 primers fitxers del tipus seleccionat. 5. Si l'usuari vol veure més fitxers es pot canviar de pàgina a la barra inferior. 6. L'usuari escull quin fitxer vol eliminar 7. Es pregunta si es vol eliminar el fitxer seleccionat. 8. S'elimina el fitxer |
| Postcondició | Fitxer eliminat |

| | |
|----------------|---|
| Cas d'ús | Modificar perfil |
| Descripció | Modificar el perfil de l'usuari |
| Actor | Metge de l'ONT |
| Precondició | Cap |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari desplega el menú d'usuari 2. L'usuari clica sobre l'opció "Modify profile" 3. L'usuari modifica els camps que desitja modificar |
| Postcondició | S'ha actualitzat el perfil de l'usuari |

| | |
|----------------|---|
| Cas d'ús | Canviar contrasenya |
| Descripció | Modificar la contrasenya de l'usuari |
| Actor | Metge de l'ONT |
| Precondició | Cap |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari desplega el menú d'usuari 2. L'usuari clica sobre la opció "Change password" 3. L'usuari introdueix la contrasenya actual i la nova contrasenya |
| Postcondició | Contrasenya de l'usuari modificada |

| | |
|----------------|---|
| Cas d'ús | Visualitzar graf |
| Descripció | Es visualitza el graf d'un conjunt de fitxers |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats anteriorment |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari desplega el menú lateral 2. L'usuari clica sobre l'opció "Graph" 3. L'usuari escull un fitxer per cada tipus 4. Si els fitxers son correctes: <ol style="list-style-type: none"> 4.1 Es mostra el graf |
| Postcondició | Visualització del graf. |

| | |
|----------------|--|
| Cas d'ús | Visualitzar cicles |
| Descripció | Es visualitza els cicles del fitxer de cicles |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats i escollits anteriorment |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari escull la opció "cycles" del menú superior |
| Postcondició | Visualització dels cicles del graf. |

| | |
|----------------|---|
| Cas d'ús | Visualitzar cicles |
| Descripció | Es visualitza les estadístiques del fitxer de cicles |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats i escollits anteriorment |
| Flux principal | <ol style="list-style-type: none"> 1. L'usuari escull la opció "stadistics" del menú superior 2. L'usuari escull quina mesura vol visualitzar |
| Postcondició | Visualització de les estadístiques d'una mesura. |

| | |
|----------------|---|
| Cas d'ús | Visualitzar possibles trasplantaments d'un node |
| Descripció | Visualitzar possibles trasplantaments als quals participa el pacient o donant d'un node |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats i escollits anteriorment |
| Flux principal | 1. L'usuari escull el node (clicant sobre el node o utilitzant el cercador) |
| Postcondició | Visualització dels possibles trasplantaments d'un node a la pestanya lateral |

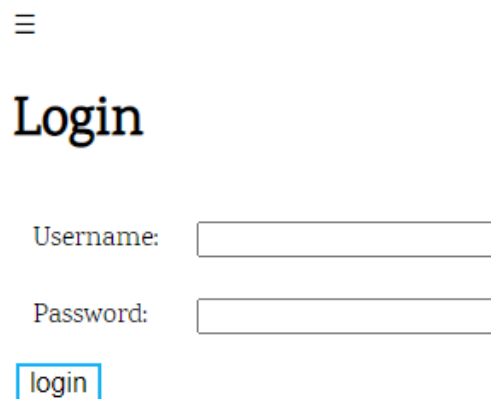
| | |
|----------------|--|
| Cas d'ús | Visualitzar possibles cicles d'un node |
| Descripció | Visualitzar possibles cicles als quals participa el pacient o donant d'un node |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats i escollits anteriorment |
| Flux principal | 1. L'usuari escull el node (clicant sobre el node o utilitzant el cercador) |
| Postcondició | Visualització dels possibles cicles d'un node a la pestanya lateral. |

| | |
|----------------|---|
| Cas d'ús | Visualitzar informació d'un cicle |
| Descripció | Visualitzar la informació d'un cicle |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats i escollits anteriorment i un node escollit anteriorment |
| Flux principal | 1. L'usuari selecciona una de les arestes que representen un cicle de la pestanya lateral o clicant l'aresta. |
| Postcondició | Visualització de la informació del cicle |

| | |
|----------------|--|
| Cas d'ús | Visualitzar la informació d'un trasplantament |
| Descripció | Visualitzar la informació d'un trasplantament |
| Actor | Metge de l'ONT |
| Precondició | Fitxers carregats i escollits anteriorment i un node escollit anteriorment |
| Flux principal | 1. L'usuari selecciona una de les arestes que representen un trasplantament de la pestanya lateral o clicant l'aresta. |
| Postcondició | Visualització de la informació del trasplantament. |

Implementació i proves

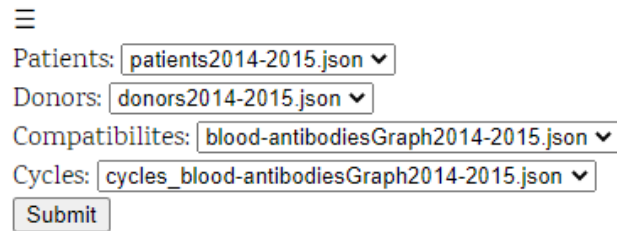
9.1 Aplicació web



The image shows a web application login form. At the top left, there is a hamburger menu icon (three horizontal lines). Below it, the word "Login" is displayed in a large, bold, serif font. Underneath, there are two input fields: the first is labeled "Username:" and the second is labeled "Password:". Below the password field is a blue button with the text "login" inside it.

Figura 9.1: Inici de sessió de l'aplicació web.

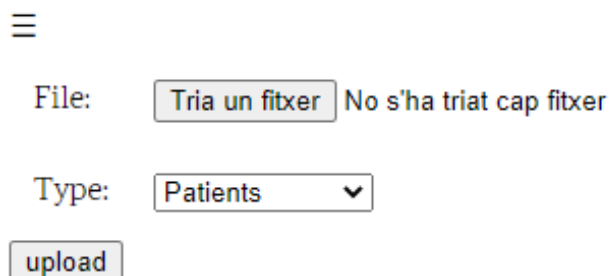
Per implementar l'aplicació web, primer es va crear un projecte de *Django* que permetis tenir un sistema d'autenticació d'usuaris. Un cop aconseguit, es va intentar incorporar la visualització del graf. Seguidament vam crear la classe *fitxer*, de manera que cada usuari pogués guardar els seus fitxers. Vam tenir dificultats per trobar com fer-ho possible perquè guardar els fitxers al mateix servidor no és molt eficient i per tant no estava pensat per fer-ho. Vam plantejar-nos canviar-ho, però en ser una aplicació pensada per ser utilitzada per poca gent i emmagatzemar relativament pocs fitxers (es realitzen 3 programes KEP l'any que equivaldria a 12 fitxers/usuari) no semblava necessari fer servir algun servei d'emmagatzematge de fitxers extern.



A web form with a hamburger menu icon at the top left. Below it are four dropdown menus and a submit button. The dropdowns are labeled 'Patients', 'Donors', 'Compatibilitats', and 'Cycles'. The selected values are 'patients2014-2015.json', 'donors2014-2015.json', 'blood-antibodiesGraph2014-2015.json', and 'cycles_blood-antibodiesGraph2014-2015.json' respectively.

Figura 9.2: Selecció dels fitxers a visualitzar en forma de graf.

Un cop creada la classe *fitxer* i aconseguir guardar els fitxers al mateix servidor, es va anar creant les diverses funcionalitats que permetien als usuaris guardar, llistar i eliminar els fitxers. També vam modificar la visualització del graf perquè es mostrés desplegable amb tots els fitxers disponibles de cada tipus per poder escollir quins fitxers componien el graf a visualitzar.



A web form with a hamburger menu icon at the top left. Below it are two fields: 'File:' with a button 'Tria un fitxer' and the text 'No s'ha triat cap fitxer', and 'Type:' with a dropdown menu showing 'Patients'. At the bottom is an 'upload' button.

Figura 9.3: Pàgina web per penjar un fitxer JSON.

Files

Type of file:

| ID | Name | Type | Date | |
|----|--|-----------------|---------------------------|------------------------|
| 22 | blood-antibodiesGraph2014-2015.json | compatibilities | Feb. 23, 2021, 10:44 a.m. | Delete |
| 23 | cycles_blood-antibodiesGraph2014-2015.json | cycles | Feb. 23, 2021, 10:44 a.m. | Delete |
| 24 | cycles_blood-antibodiesGraph2014-2015.json | cycles | Feb. 27, 2021, 6:19 p.m. | Delete |
| 25 | cycles_blood-antibodiesGraph2014-2015.json | cycles | Feb. 27, 2021, 6:19 p.m. | Delete |

« 1 2 **3** »

Figura 9.4: Llista dels fitxers d'un usuari.

Seguidament es van implementar algunes funcionalitats per permetre als usuaris canviar informació sobre el seu perfil:

Modify Profile

Username:

Email:

First name:

Last name:

Figura 9.5: Formulari per modificar el perfil d'un usuari.

Change Password

Old password:

New password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation:

Figura 9.6: Formulari per canviar la contrasenya d'un usuari.

Per últim, es va afegir el càlcul de mesures realitzades en Python. Per fer-ho es va crear una petició html que permetés donar els identificadors de 4 fitxers de tipus diferents, calcular les mesures i enviar-les en format JSON.

Per provar totes les funcionalitats, es van crear diferents usuaris i es van provar diferents situacions per comprovar que totes les funcionalitats funcionessin correctament.

La implementació de l'aplicació web es pot trobar dins la carpeta *KEPweb* dels annexos.

9.2 Visualització del graf

9.2.1 Graf de compatibilitats

La visualització del graf s'ha implementat en JavaScript utilitzant la llibreria de D3.js (com s'ha explicat anteriorment) i es pot trobar en el fitxer *graph.js* en els annexos. Per fer-ho primer s'ha de reestructurar la informació dels pacients, donants i possibles trasplantaments per poder-hi accedir més fàcilment. Un cop convertits els fitxers JSON a un diccionari de JavaScript, necessitàvem tenir per cada pacient, la llista dels donants vinculats a ells i en cas de trobar un donant altruista, crear un pacient "fictici" per facilitar la visualització. També necessitàvem tenir una matriu de booleans amb totes les arestes del graf i una llista de les arestes.

Un cop reorganitzada la informació es va crear el graf utilitzant la comanda de D3.js *forceSimulation* que té com a paràmetres una llista d'índex (nodes) i una llista de tuples amb els elements *source* i *target* (arestes). Aquesta comanda

permet determinar una força cap al centre del graf i un altre entre les arestes que pot ser igual per totes o dependre de la puntuació d'aquesta. En el nostre cas vam decidir no tenir en compte la puntuació perquè pràcticament no es veia diferència i era costós de calcular.

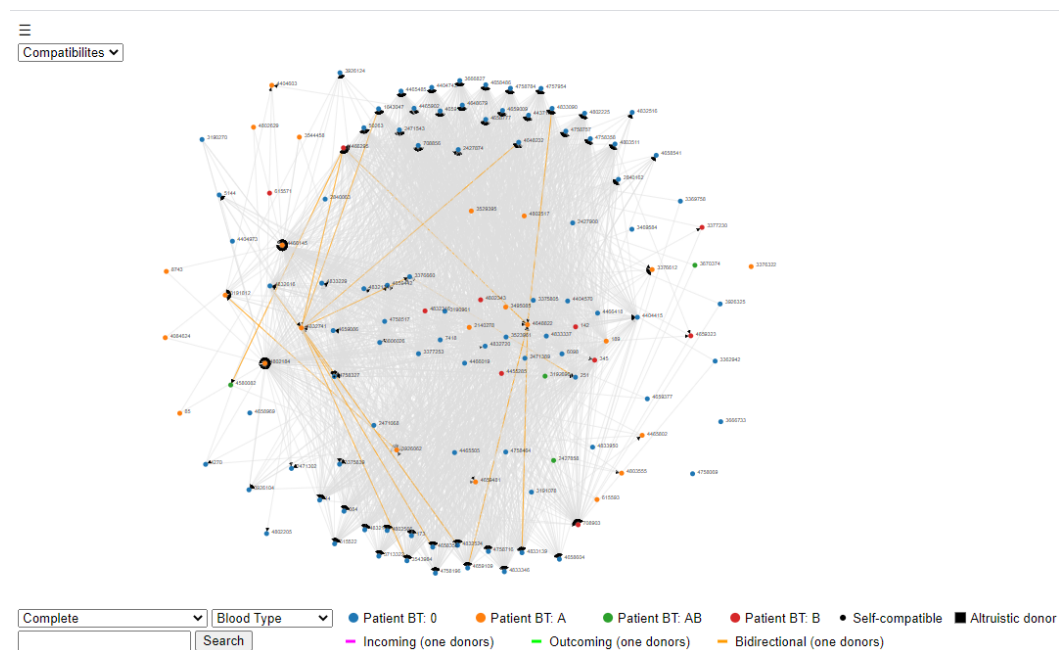


Figura 9.7: Graf de compatibilitats.

Al principi vam assignar els colors dels nodes del graf segons el seu grup sanguini, i posteriorment es va crear un desplegable que en canviar de valor, canvia el color dels nodes per representar una diferents mesures.

Un cop representat el graf, es va crear interaccions amb ell per poder profunditzar la visualització. Es va crear un esdeveniment que seleccionés un node en ser clicat i permetés només visualitzar els nodes del graf que tinguessin alguna aresta amb el node escollit.

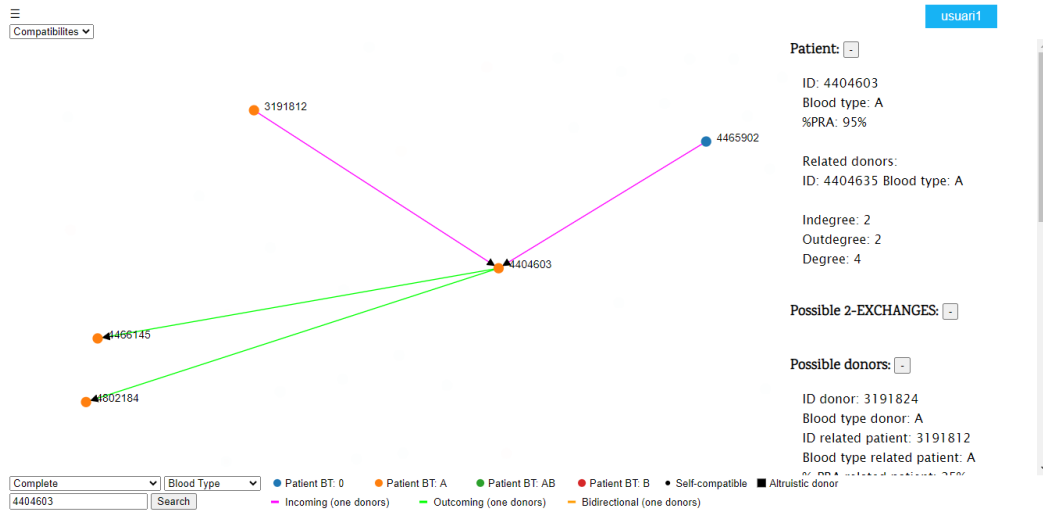


Figura 9.8: Visualització en detall d'un node.

També es va crear una pestanya lateral a on es mostrava la informació d'aquell pacient i els possibles trasplantaments.

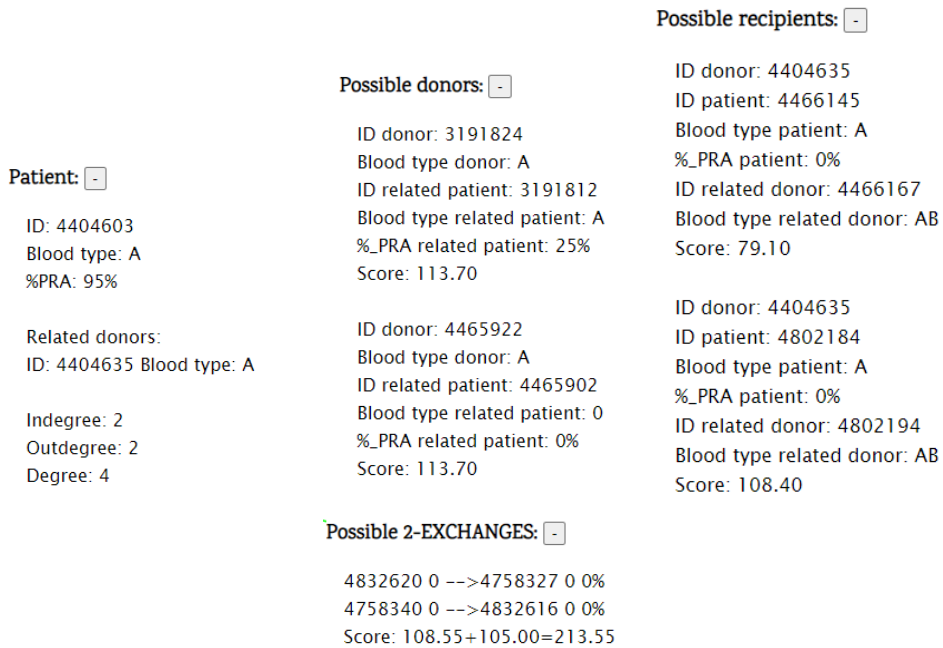


Figura 9.9: Informació d'un node.

Per facilitar la visualització es van assignar colors a les arestes segons la direcció respecte al node seleccionat: verd per arestes que surten, rosa per arestes que entren i taronja per arestes bidireccionals. Posteriorment es van afegir 3 co-

lors més per representar quan una aresta era múltiple (possible quan hi ha més d'un donant vinculat a un pacient). Els colors afegits eren els mateixos però una tonalitat més fosca. També es va crear un desplegable per poder filtrar el tipus d'arestes a visualitzar. Posteriorment es va crear un cercador amb la mateixa funcionalitat per facilitar cercar un pacient en el graf.

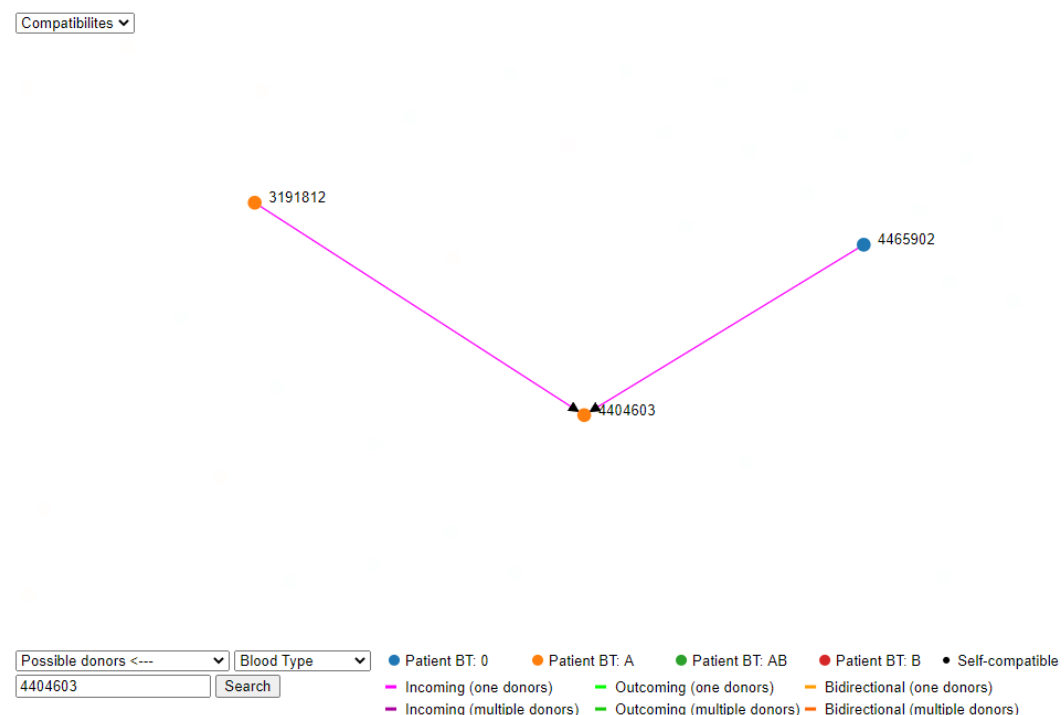


Figura 9.10: Visualització dels possibles donants d'un node.

Per visualitzar el pes de les arestes (puntuació del trasplantament) vam provar diverses maneres. Al principi mostràvem el pes de totes les arestes d'un node seleccionat, però ens vam adonar que era difícil d'entendre quan hi havia moltes arestes, arestes bidireccionals o pacients amb múltiples donants.

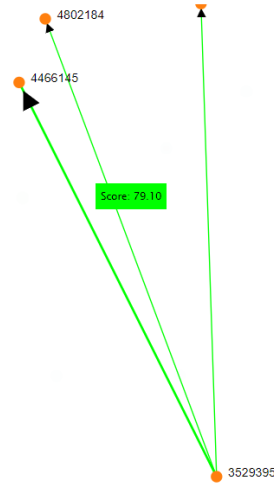


Figura 9.11: Selecció d'una aresta.

Per solucionar-ho vam decidir crear una descripció emergent quan es passava el punter per sobre una aresta a on es podia veure la puntuació de l'aresta i en cas de tenir-ne múltiples (arestes bidireccionals o amb múltiples donants) s'incorporava també una petita descripció per tenir clar a quin possible trasplantament feia referència cada puntuació.

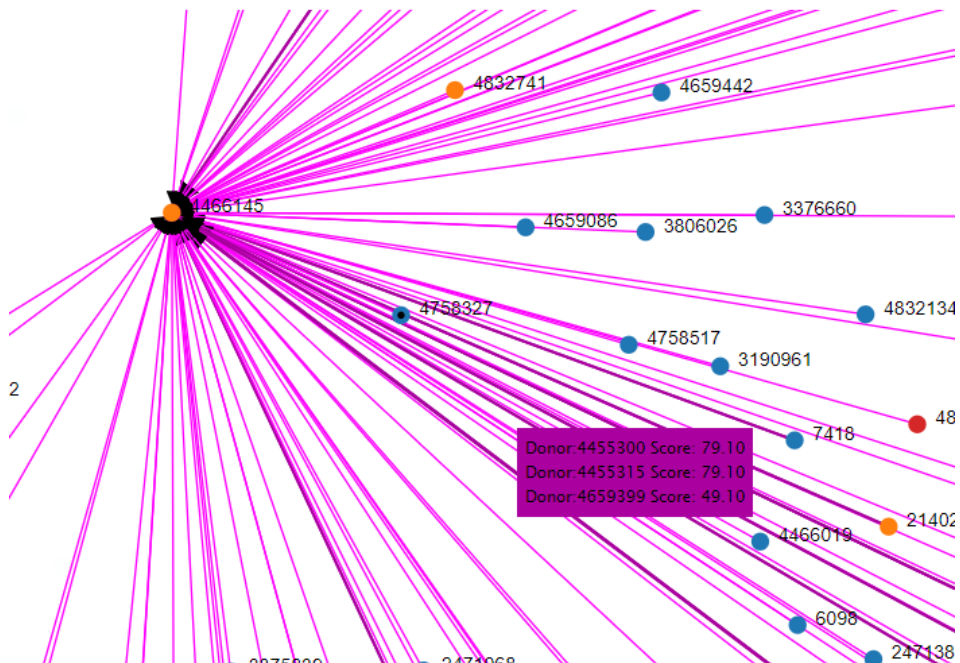


Figura 9.12: Selecció d'una aresta múltiple.

Finalment, es va crear una interacció amb les arestes que només està dis-

ponible quan hi ha un node seleccionat. Aquesta interacció permet visualitzar la informació referent a aquella aresta (o arestes en cas de ser múltiple) en la pestanya lateral.

Possible(s) transplantment(s):

8746 A -->4466145 A 0%

Score: 84.10

Figura 9.13: Informació d'un trasplantament.

Per provar que tot funcionava correctament, es va provar els diferents conjunts de dades que disposàvem i vam afegir algun node i arestes per simular casos que no teníem (com és el cas del donant altruista).

9.2.2 Graf de cicles

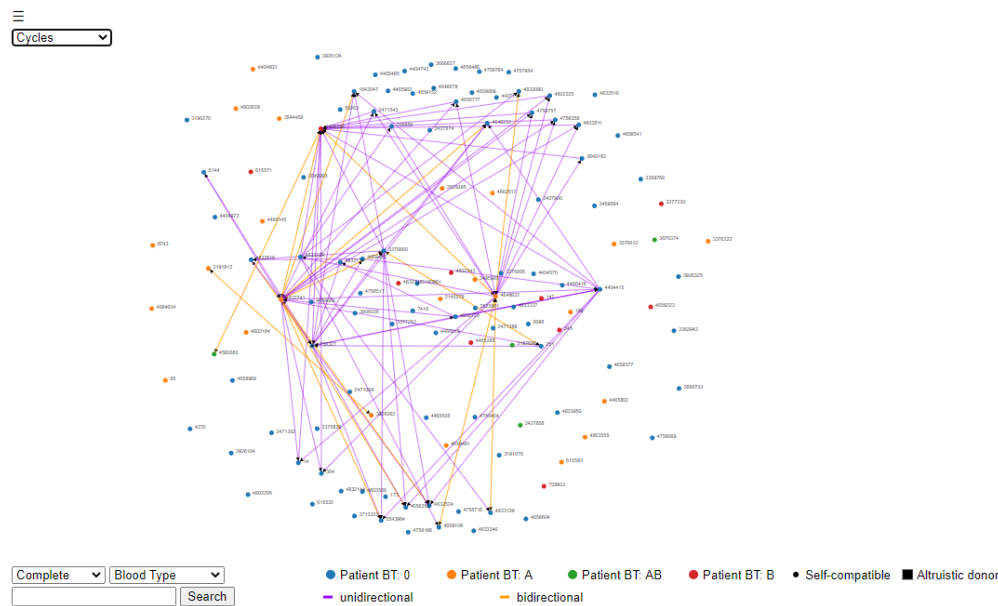


Figura 9.14: Visualització del graf de cicles.

Per visualitzar els cicles, el que es fa és modificar el graf de compatibilitats perquè només es visualitzin les arestes que formen part d'un cicle. També s'ha modificat les interaccions perquè quan se seleccioni un node, es visualitzin els cicles sencers que inclouen aquell node. Els cicles es pinten de color taronja en cas de

ser cicles de 2 i de color lila els cicles de 3. També es modifica el desplegable del menú inferior per poder filtrar segons les categories anteriors.

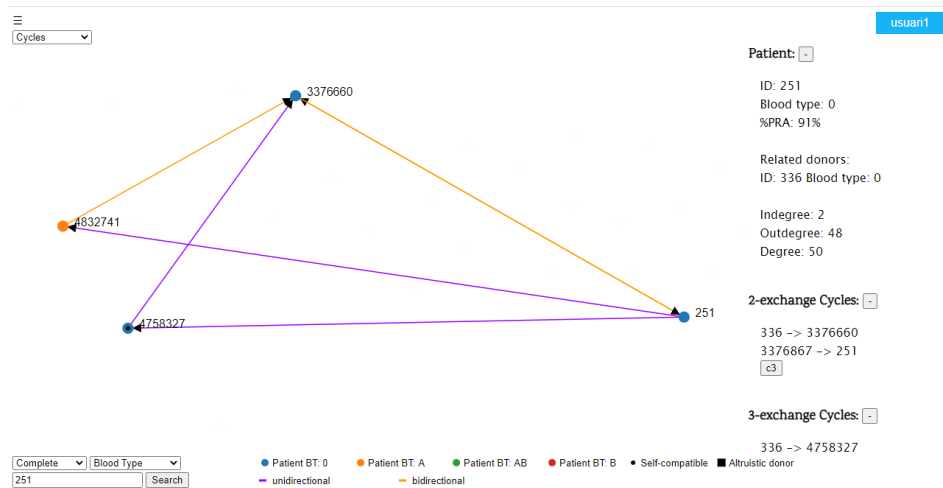


Figura 9.15: Visualització dels possibles cicles d'un node.

En escollir una aresta, es canvia la informació de la pestanya lateral, però no es canvia la visualització, ja que la informació important en el graf de cicles és el conjunt de trasplantaments que formen els cicles. Es pot visualitzar un sol cicle amb l'opció que apareix dins la llista de cicles d'un node a la pestanya lateral en escollir un node en concret.

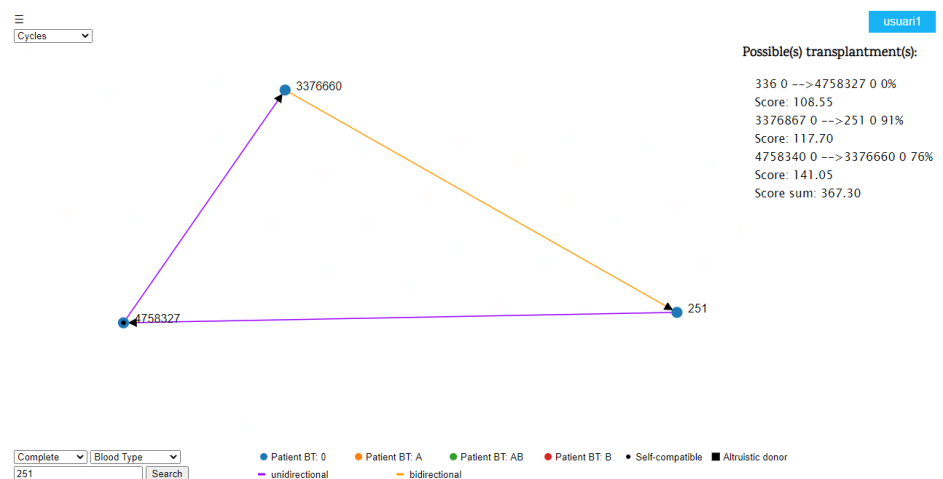


Figura 9.16: Visualització d'un cicle.

Per comprovar que tot funcionava correctament, es va provar amb totes les possibles combinacions de fitxer que teníem i també es va crear algun fitxer

de solució extra per simular la incorporació d'un donant altruista i per tant la visualització de cadenes.

9.2.3 Estadístiques

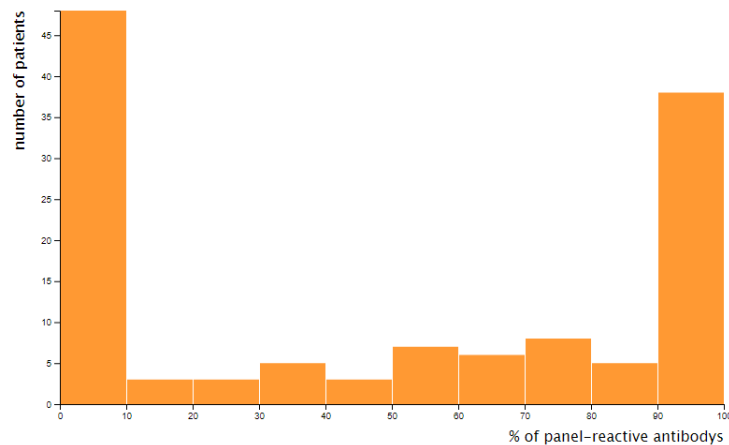


Figura 9.17: Histograma del PRA dels pacients.

L'apartat d'estadístiques va ser l'últim a implementar-se. L'objectiu era mostrar de forma gràfica la importància de diferents mesures. Per fer-ho es va fer servir la funció de D3.js *d3.histogram* que permet donat un conjunt de dades, representar-les en forma d'histograma. Al principi es va fer un sol histograma per mesura, però posteriorment es va implementar un altre histograma doble a on es pogués veure la diferència entre els nodes que acaben escollits en el fitxer de cicles i aquells que no.

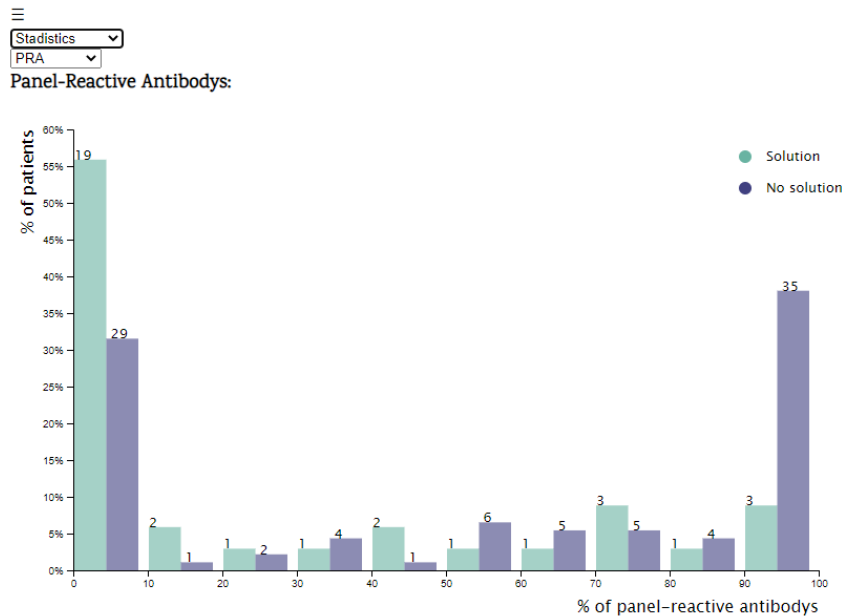


Figura 9.18: Histograma del PRA dels pacients, depenent de si pertanyen a la solució o no.

Igual que als apartats anteriors, vam provar que es visualitzés correctament amb els diferents fitxers que teníem disponibles.

9.3 Mesures del graf

Per calcular les diverses mesures, primer es va realitzar un petit codi en Python a on es carregaven diferents fitxers i es calculaven les diverses mesures i posteriorment es va incorporar a la pàgina web.


```
#Creació graf:
G = nx.DiGraph()
for p in patients['patients']:
    G.add_node(p) #Afehim nodes
donants=donors['donors']
arestes2=arestes['donor_patients']
arestesNoRepetides={}
#Preparem les arestes del graf en format llista de tuples (origen, desti, pes)
for a in arestes2:
    related_patient=donants[a]['RELATED_PATIENT']
    conjuntArestes=arestes2[a]
    for p in conjuntArestes:
        patient=p['ID_PATIENT']
        score=p['SCORE']
        transplant=(related_patient,patient)
        if not ( transplant in arestesNoRepetides):
            arestesNoRepetides[transplant]=score
        else:
            if arestesNoRepetides[transplant]<score:
                arestesNoRepetides[transplant]=score
list = [(d,p, v) for (d,p), v in arestesNoRepetides.items()]
G.add_weighted_edges_from(list) # Afehim arestes
```

Figura 9.19: Codi Python de la creació del graf utilitzant la llibreria de NetworkX.

Primer es va implementar les mesures de teoria de graf, utilitzant la llibreria NetworkX de Python.

```

#Grau
degree =G.degree
#Grau amb pes
degreeW =G.degree(weight='weight')
#Grau d'entrada
in_degree =G.in_degree
#Grau d'entrada amb pes
in_degreeW =G.in_degree(weight='weight')
#Grau de sortida
out_degree=G.out_degree
#Grau de sortida amb pes
out_degreeW=G.out_degree(weight='weight')
#Closeness Centrality
closeness_centrality=nx.closeness_centrality(G)
#Betweenness Centrality
betweenness_centrality=nx.betweenness_centrality(G, weight='weight')
#Eigenvector Centrality
eigenvector_centrality=nx.eigenvector_centrality(G, weight='weight')
#Trophic Levels
trophic_levels=nx.trophic_levels(G,weight='weight')

```

Figura 9.20: Codi Python del càlcul de les mesures de Teoria de Grafs.

Seguidament, es va anar incorporant les diverses mesures de teoria de la informació descrites a l'apartat de conceptes previs. En calcular el vector propi de la matriu d'adjacències, s'havia d'eliminar les files i columnes que no tinguessin cap valor diferent de 0 per aconseguir un vector propi real (si no sortien valors imaginaris). En fer-ho es va reduir molt el nombre de parelles pacient-donant que podien aspirar a formar part de la solució.

```

def matTran_vecPropi(adj):
    """
    Calcula la matriu de probabilitats de transició i el seu vector propi
    (probabilitat estacionaria), la entropia del graf i la sorpresa
    entròpica dels nodes (dos variacions de la mateixa mesura)

    """
    matTran=np.apply_along_axis(lambda x: x/np.sum(x), axis=1,arr=adj.todense())
    mtt = np.transpose(matTran)
    Eigenvalues, Eigenvectors = np.linalg.eig(mtt)
    norm1 =np.array( Eigenvectors[:, 0] / np.sum(Eigenvectors[:, 0])).flatten()
    norm1=np.real(norm1)
    entropia=-sum(map(lambda x: x*np.log2(x),norm1 ))
    sorpresa_entropica=lambda x: -np.log2(x)*x
    sorpresa_entropica2=lambda x: -np.log2(x)
    return norm1, entropia,sorpresa_entropica(norm1),sorpresa_entropica2(norm1)

```

```
def informacioMutua(matriu2, vector, entropia):
    """
    Donada la matriu d'adjecencies, la probabilitat estacionaria i l'entropia
    d'un graf, calcula les 4 variants de la informació mutua dels diferents
    nodes del graf
    """
    matriu = np.apply_along_axis(lambda x: x / np.sum(x), axis=1,
                                 arr=matriu2.todense())
    infMutua=[]
    infMutua2=[]
    infMutua3=[]
    infMutua4=[]
    for i in range(len(vector)):
        act=0
        act2=0
        act3=0
        for j in range(len(vector)):
            act3+=matriu[i,j]
            if vector[j]!=0 and matriu[i,j]!=0:
                act+=matriu[i,j]*np.log2(matriu[i,j]/vector[j])
                act2+=matriu[i,j]*np.log2(matriu[i,j])
        infMutua.append(act)
        infMutua2.append(act*vector[i])
        infMutua3.append(entropia+act2)
        infMutua4.append((entropia+act2)*vector[i])
    return infMutua, infMutua2, infMutua3, infMutua4
```

Figura 9.21: Codi Python del càlcul de les mesures de Teoria de la Informació.

Per comprovar que les mesures estiguessin ben implementades, es van calcular per un graf petit a on es podia comprovar els càlculs manualment.

9.4 Regressió logística

En la creació del model de regressió logística, es van utilitzar diferents mètodes. Al principi, per buscar el millor model, començàvem amb un model amb totes les mesures (mèdiques, de teoria de la informació i de teoria de graf) i anàvem traient del model la mesura menys significativa fins a aconseguir el model amb el valor AIC més petit. Ràpidament ens vam adonar que no era una bona solució perquè en tenir moltes mesures semblants, el model no convergia, ja que un dels requisits de la regressió logística és que les variables han de ser independents entre elles. Per solucionar-ho vam decidir canviar la manera de buscar el millor model. En lloc de començar amb totes les mesures, començàvem amb una sola mesura i buscàvem el model amb millor AIC possible. Seguidament buscàvem quina mesura quan la incorporàvem millorava el model. Per fer-ho vam crear un petit script amb R per automatitzar-ho. Aquesta funció es pot trobar en els annexos en el fitxer *CreacioModel.R*.

Un cop vam obtenir quin era el millor model, el guardem en un fitxer *.rds* i guardem les dependències del nostre model en un fitxer *.dep*. Aquest dos fitxers ens permeten mitjançant la llibreria *rpy2* fer prediccions en el nostre model.

Per poder fer les prediccions des de Python, vam crear una classe anomenada *Model* que permet carregar un model i fer prediccions. Aquest fitxer es pot trobar als annexos en nom de *Model.py*.

```

creaciomodel<-function(data) {
  c <-colnames(data)
  llargada=length(c)
  ## Creem una llista per guardar quines mesures ja hem afegit
  x=data.frame(row.names=c, val=replicate(llargada,0))
  millorAIC<-300
  mida<- 1:length(x[,])
  aics<-vector()
  resultats<-matrix(0,nrow=(length(x[,])),ncol=36)
  volta<-1
  while(volta<length(x[,])){
    mAICtext<-" "
    prvalues <-matrix()
    trobat=FALSE
    for (i in mida){
      ## Si no ha esta escollit anteriorment (o es una solucio)
      if (x[c[i],]==0){
        textAc<-c[i]
        remove(names)
        names<-vector()
        for (columna in c){
          if (x[columna,]==1){
            ## Creem llista amb les mesures a posar a la formula
            names<-append(names,columna)
          }
        }
        names<-append(names,textAc)
        ## Creem la formula
        form <- as.formula(paste("SOLUCIO_1~",paste(names,collapse=" + ")))
        ## Modelem el model
        model <- glm(as.formula(form),family="binomial", data=data)
        resultats[volta,i]<-AIC(model) ## Ens guardem l'AIC del model
        if(AIC(model)<millorAIC){
          millorAIC<-AIC(model)
          millorFormula<-form
          mAICtext<-textAc
          aics<-append(aics,AIC(model))
          millorModel<-model
          trobat=TRUE
        }
      }
    }
    x[mAICtext,]<-1
    volta<-volta+1
    if(!trobat){
      break ## Si cap model es millor que el de la iteracio anterior acabem.
    }
  }
}

```

Figura 9.22: Codi en R de la creació del model.

Implantació i resultats

10.1 Regressió logística

En el desenvolupament del nostre model, tal i com s'ha explicat anteriorment, hem començat creant models amb una sola variable i comparant els diferents valors AIC obtinguts:

Mesures Mediques:

| | |
|---|-------------|
| Grup Sanguini: | 98,55700833 |
| Genere: | 95,44166769 |
| Edat: | 95,58255303 |
| PRA: | 94,81217655 |
| Temps en el pool: | 92,45223859 |
| Mesos en diàlisis: | 95,94680715 |
| Nombre de donants amb grup sanguini A: | 93,27882821 |
| Nombre de donants amb grup sanguini B: | 95,19003828 |
| Nombre de donants amb grup sanguini AB: | 93,05884681 |
| Nombre de donants amb grup sanguini O: | 92,54603999 |
| Nombre de donants: | 93,61274105 |
| Parella pacient-donant compatible amb ella mateixa: | 91,66912475 |

Mesures de Teoria de Grafes:

| | |
|-------------------------------------|-------------|
| Grau del node: | 86,40200132 |
| Grau de sortida del node: | 87,54799221 |
| Grau d'entrada del node: | 95,83781955 |
| Grau del node amb pesos: | 86,40200132 |
| Grau de sortida del node amb pesos: | 87,62682301 |
| Grau d'entrada del node amb pesos: | 95,88904209 |
| Clossness Centrality: | 90,07829053 |
| Betweenness Centrality: | 69,33618415 |
| Trophic Levels: | 80,68577523 |
| Eigenvector Centrality: | 94,76450387 |

Mesures de Teoria de la Informació:

| | |
|-----------------------------------|-------------|
| Sorpresa Entròpica (fórmula 5.2): | 85,13682453 |
|-----------------------------------|-------------|

| | |
|-----------------------------------|-------------|
| Sorpresa Entròpica (fórmula 5.3): | 67,33163311 |
| Informació Mútua (fórmula 5.8): | 78,10939072 |
| Informació Mútua (fórmula 5.9): | 68,56632675 |
| Informació Mútua (fórmula 5.10): | 95,92126756 |
| Informació Mútua (fórmula 5.11): | 84,98078835 |

Tal com es pot veure, la mesura amb un model millor seria la sorpresa entròpica calculada segons la segona fórmula 5.3 (es multiplica per la probabilitat estacionària d'aquell node). Les altres dues mesures que també mostren models amb AIC semblants serien la Informació mútua (fórmula 5.9) i la Betweenness Centrality.

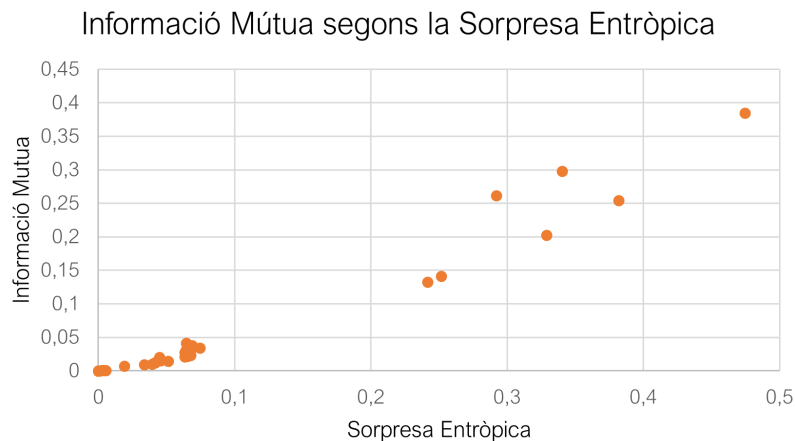


Figura 10.1: Informació Mútua segons la Sorpresa Entròpica.

Tal com es pot veure en el gràfic de dispersió, la informació mútua i la sorpresa entròpica són dues mesures de Teoria de la Informació que van molt lligades, fet que pot provocar que si una entra al model, l'altre en quedi fora. En canvi si comparem la Sorpresa Entròpica amb la Betweenness Centrality, podem veure que estan menys relacionades, per tan probablement la mesura acabarà dins del model.

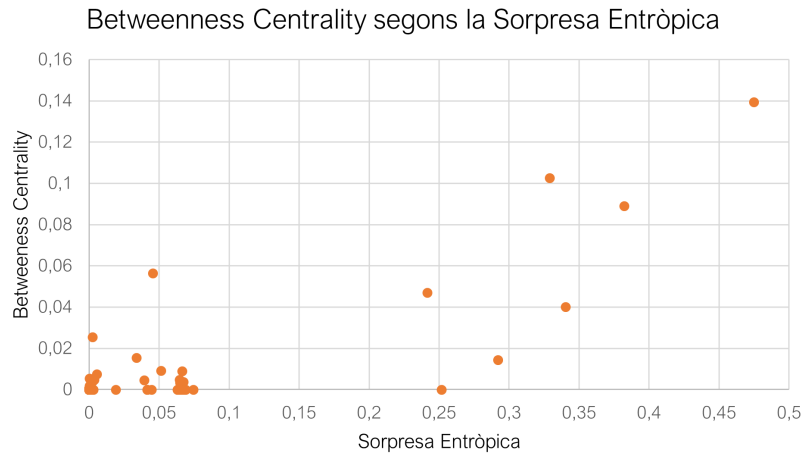


Figura 10.2: Betweenness Centrality segons la Sorpresa Entròpica.

Seguidament vam comparar tots els models creats a partir d'afegir una mesura al model guanyador:

Mesures Mediques:

| | |
|---|-------------|
| Grup Sanguini: | 68,25279364 |
| Genere: | 69,21374834 |
| Edat: | 69,09257572 |
| PRA: | 69,22378466 |
| Temps en el pool: | 66,91856552 |
| Mesos en diàlisis: | 68,97549027 |
| Nombre de donants amb grup sanguini A: | 66,17660083 |
| Nombre de donants amb grup sanguini B: | 61,81034941 |
| Nombre de donants amb grup sanguini AB: | 68,10173071 |
| Nombre de donants amb grup sanguini 0: | 61,93738013 |
| Nombre de donants: | 68,12213497 |
| Parella pacient-donant compatible amb ella mateixa: | 64,79321348 |

Mesures de Teoria de Grafs:

| | |
|-------------------------------------|-------------|
| Grau del node: | 56,92926306 |
| Grau de sortida del node: | 52,60845321 |
| Grau d'entrada del node: | 69,05281014 |
| Grau del node amb pesos: | 56,92926306 |
| Grau de sortida del node amb pesos: | 52,87451616 |
| Grau d'entrada del node amb pesos: | 69,18459903 |
| Clossness Centrality: | 68,52611809 |
| Betweenness Centrality: | 65,45528821 |

| | |
|-------------------------|-------------|
| Trophic Levels: | 66,12317862 |
| Eigenvector Centrality: | 69,0158566 |

Mesures de Teoria de la Informació:

| | |
|-----------------------------------|-------------|
| Sorpresa Entròpica (fórmula 5.3): | 57,83214028 |
| Informació Mútua (fórmula 5.8): | 64,29511399 |
| Informació Mútua (fórmula 5.9): | 69,28394659 |
| Informació Mútua (fórmula 5.10): | 60,01330506 |
| Informació Mútua (fórmula 5.12): | 62,3795539 |

Com es pot veure, la segona mesura més significativa és el grau de sortida d'un node sense tenir en compte la puntuació (52,61), que guanya per molt poca diferència al grau de sortida tenint en compte la puntuació (52,87). Seguidament vam comparar tots els models creats a partir d'afegir una mesura al model guanyador:

Mesures Mediques:

| | |
|---|-------------|
| Grup Sanguini: | 42,23761393 |
| Genere: | 54,52169521 |
| Edat: | 54,48207847 |
| PRA: | 53,16069989 |
| Temps en el pool: | 51,34605875 |
| Mesos en diàlisis: | 54,39795466 |
| Nombre de donants amb grup sanguini A: | 54,27809917 |
| Nombre de donants amb grup sanguini B: | 39,86246687 |
| Nombre de donants amb grup sanguini AB: | 54,48644676 |
| Nombre de donants amb grup sanguini 0: | 52,42540901 |
| Nombre de donants: | 51,10021013 |
| Parella pacient-donant compatible amb ella mateixa: | 53,38466282 |

Mesures de Teoria de Grafs:

| | |
|-------------------------------------|-------------|
| Grau del node: | 48,5629781 |
| Grau d'entrada del node: | 48,5629781 |
| Grau del node amb pesos: | 48,5629781 |
| Grau de sortida del node amb pesos: | 53,45354627 |
| Grau d'entrada del node amb pesos: | 47,10770452 |
| Clossness Centrality: | 44,73878596 |
| Betweenness Centrality: | 54,56728843 |
| Trophic Levels: | 45,83866482 |
| Eigenvector Centrality: | 52,42934046 |

| | |
|-------------------------------------|-------------|
| Mesures de Teoria de la Informació: | |
| Sorpresa Entròpica (fórmula 5.3): | 46,44891665 |
| Informació Mútua (fórmula 5.8): | 40,42696361 |
| Informació Mútua (fórmula 5.9): | 53,73558123 |
| Informació Mútua (fórmula 5.10): | 54,21383318 |
| Informació Mútua (fórmula 5.11): | 45,3658072 |

Com es pot veure, la mesura guanyadora d'aquesta iteració és el nombre de donants amb grup sanguini B (39,86). La mesura en segona posició (40,43) és la informació mútua i en tercera posició hi ha el grup sanguini del pacient (42,24). Aquestes dues mesures són molt semblants, fet que provocarà que el grau del node tenint en compte el pes no entrarà en el model. Seguidament vam comparar tots els models creats a partir d'afegir una mesura al model guanyador:

| | |
|---|-------------|
| Mesures Mediques: | |
| Grup Sanguini: | 42,65759113 |
| Genere: | 41,85776616 |
| Edat: | 41,58320733 |
| PRA: | 41,83507505 |
| Temps en el pool: | 40,87251491 |
| Mesos en diàlisis: | 40,55228278 |
| Nombre de donants amb grup sanguini A: | 36,5504851 |
| Nombre de donants amb grup sanguini AB: | 41,86243865 |
| Nombre de donants amb grup sanguini O: | 40,63720494 |
| Nombre de donants: | 39,03990903 |
| Parella pacient-donant compatible amb ella mateixa: | 41,85929923 |

| | |
|-------------------------------------|-------------|
| Mesures de Teoria de Grafs: | |
| Grau del node: | 39,504891 |
| Grau d'entrada del node: | 39,504891 |
| Grau del node amb pesos: | 39,504891 |
| Grau de sortida del node amb pesos: | 39,30300938 |
| Grau d'entrada del node amb pesos | 39,34611421 |
| Clossness Centrality: | 35,36501542 |
| Betweenness Centrality: | 41,31628898 |
| Trophic Levels: | 37,71646191 |
| Eigenvector Centrality: | 40,71421486 |

Mesures de Teoria de la Informació:

| | |
|-----------------------------------|-------------|
| Sorpresa Entròpica (fórmula 5.3): | 37,7155688 |
| Informació Mútua (fórmula 5.8): | 37,63023679 |
| Informació Mútua (fórmula 5.9): | 41,8443585 |
| Informació Mútua (fórmula 5.10): | 40,77815822 |
| Informació Mútua (fórmula 5.11): | 41,81367436 |

En aquesta interacció la mesura més significativa és la mesura de teoria de grafs *closness centrality* (35,37). Seguida del nombre de donants amb grup sanguini AB (36,55) i la informació mútua (37,63). Seguidament vam comparar tots els models creats a partir d'afegir una mesura al model guanyador:

| | |
|---|-------------|
| Grup Sanguini: | 37,45428392 |
| Genere: | 37,36477048 |
| Edat: | 37,36050799 |
| PRA: | 37,22100026 |
| Temps en el pool: | 37,14130743 |
| Mesos en diàlisis: | 37,26299977 |
| Nombre de donants amb grup sanguini A: | 35,54189902 |
| Nombre de donants amb grup sanguini AB: | 34,65923135 |
| Nombre de donants amb grup sanguini 0: | 35,69566134 |
| Nombre de donants: | 33,57153311 |
| Parella pacient-donant compatible amb ella mateixa: | 37,36447688 |

| | |
|-------------------------------------|-------------|
| Mesures de Teoria de Grafs: | |
| Grau del node: | 36,63744247 |
| Grau d'entrada del node: | 36,63744247 |
| Grau del node amb pesos: | 36,63744247 |
| Grau de sortida del node amb pesos: | 35,9223048 |
| Grau d'entrada del node amb pesos: | 36,98573002 |
| Betweenness Centrality: | 32,88912735 |
| Trophic Levels: | 37,34685007 |
| Eigenvector Centrality: | 32,96811833 |

| | |
|-------------------------------------|-------------|
| Mesures de Teoria de la Informació: | |
| Sorpresa Entròpica (fórmula 5.3): | 37,1004351 |
| Informació Mútua (fórmula 5.8): | 37,07654204 |
| Informació Mútua (fórmula 5.9): | 37,32139045 |
| Informació Mútua (fórmula 5.10): | 37,20033683 |
| Informació Mútua (fórmula 5.11): | 36,81967739 |

En aquesta iteració la mesura guanyadora és la mesura de teoria de grafs *Betweenness centrality* (32,89). Seguida de molt a prop per una altra mesura de centralitat de grafs *Eigenvector centrality*.

| | |
|---|-------------|
| Grup Sanguini: | 34,84802451 |
| Genere: | 34,88683485 |
| Edat: | 34,66406994 |
| PRA: | 34,67382129 |
| Temps en el pool: | 34,58907568 |
| Mesos en diàlisis: | 33,94315626 |
| Nombre de donants amb grup sanguini A: | 34,43833494 |
| Nombre de donants amb grup sanguini AB: | 28,4448117 |
| Nombre de donants amb grup sanguini O: | 34,85745222 |
| Nombre de donants: | 34,02827697 |
| Parella pacient-donant compatible amb ella mateixa: | 34,69574711 |
| Mesures de Teoria de Grafs: | |
| Grau del node: | 34,56602706 |
| Grau d'entrada del node: | 34,56602706 |
| Grau del node amb pesos: | 34,56602706 |
| Grau de sortida del node amb pesos: | 31,31591399 |
| Grau d'entrada del node amb pesos: | 34,67265918 |
| Trophic Levels: | 34,85567132 |
| Eigenvector Centrality: | 28,57698492 |
| Mesures de Teoria de la Informació: | |
| Sorpresa Entròpica (fórmula 5.3): | 34,85075473 |
| Informació Mútua (fórmula 5.8): | 34,07767541 |
| Informació Mútua (fórmula 5.9): | 34,7255627 |
| Informació Mútua (fórmula 5.10): | 34,76105596 |
| Informació Mútua (fórmula 5.11): | 34,88903808 |

Com es pot veure, ja només queden 3 mesures que a l'afegir-les milloren el model (Nombre de donants AB, Eigenvector Centrality i Grau de sortida amb pesos). En aquesta iteració la mesura més significativa és el nombre de donants que té un pacient amb grup sanguini AB (28,58).

| | |
|----------------|-------------|
| Grup Sanguini: | 34,40432935 |
| Genere: | 29,74464477 |
| Edat: | 30,43313651 |
| PRA: | 30,44433484 |

| | |
|---|-------------|
| Temps en el pool: | 29,27590047 |
| Mesos en diàlisis: | 30,15503026 |
| Nombre de donants amb grup sanguini A: | 28,69077044 |
| Nombre de donants amb grup sanguini 0: | 30,40977266 |
| Nombre de donants: | 29,90105256 |
| Parella pacient-donant compatible amb ella mateixa: | 30,04558452 |
| Mesures de Teoria de Grafs: | |
| Grau del node: | 30,44219065 |
| Grau d'entrada del node: | 30,44219065 |
| Grau del node amb pesos: | 30,44219065 |
| Grau de sortida del node amb pesos: | 29,88634857 |
| Grau d'entrada del node amb pesos: | 30,43380821 |
| Trophic Levels: | 30,01380868 |
| Eigenvector Centrality: | 30,3621977 |
| Mesures de Teoria de la Informació: | |
| Sorpresa Entròpica (fórmula 5.3): | 29,10890583 |
| Informació Mútua (fórmula 5.8): | 29,85705039 |
| Informació Mútua (fórmula 5.9): | 30,003527 |
| Informació Mútua (fórmula 5.10): | 30,42648642 |
| Informació Mútua (fórmula 5.11): | 30,44150098 |

Com es pot veure, cap de les mesures ha millorat el model en ser afegides.
El model resultant és el següent:

SOLUCIO ~ Grau de sortida + Closness Centrality
+ Betweeness Centrality + Número de donants B + Sorpresa
Entropica (fórmula 5.3) + Número de donants AB

```

Call:
glm(formula = SOLUCIO_1 ~ OUTDEGREE + CLOSENESS_CENTRALITY +
     BETWEENNESS_CENTRALITY + DONORS_B + SORPRESA_ENTROPICA_1 +
     DONORS_AB, family = "binomial", data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.04982 -0.01055  0.00000  0.00000  2.07479

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -68.8528    34.6210  -1.989  0.0467 *
OUTDEGREE         0.8219     0.4015   2.047  0.0406 *
CLOSENESS_CENTRALITY 210.6137  112.3876   1.874  0.0609 .
BETWEENNESS_CENTRALITY -471.7155  260.0199  -1.814  0.0697 .
DONORS_B          -6.3449    23.4841  -0.270  0.7870
SORPRESA_ENTROPICA_1  115.2352    72.3521   1.593  0.1112
DONORS_AB        -117.8347  4607.0907  -0.026  0.9796
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 91.986  on 125  degrees of freedom
Residual deviance: 14.445  on 119  degrees of freedom
AIC: 28.445

Number of Fisher scoring iterations: 20

```

Figura 10.3: Resum del model obtingut amb R.

Com es pot veure, podem validar la nostra hipòtesi inicial que la centralitat d'un node influeix a determinar si una parella pacient-donant acabarà sent escollida per la solució perquè les mesures de centralitat Grau de sortida, Closness Centrality i Betweenness Centrality han estat escollides per formar part del model.

Un cop obtingut el model, podem predir si una parella pacient-donant acabarà a la solució. Aquesta predicció ha estat incorporada a la aplicació web a on es pot visualitzar amb el color dels nodes:

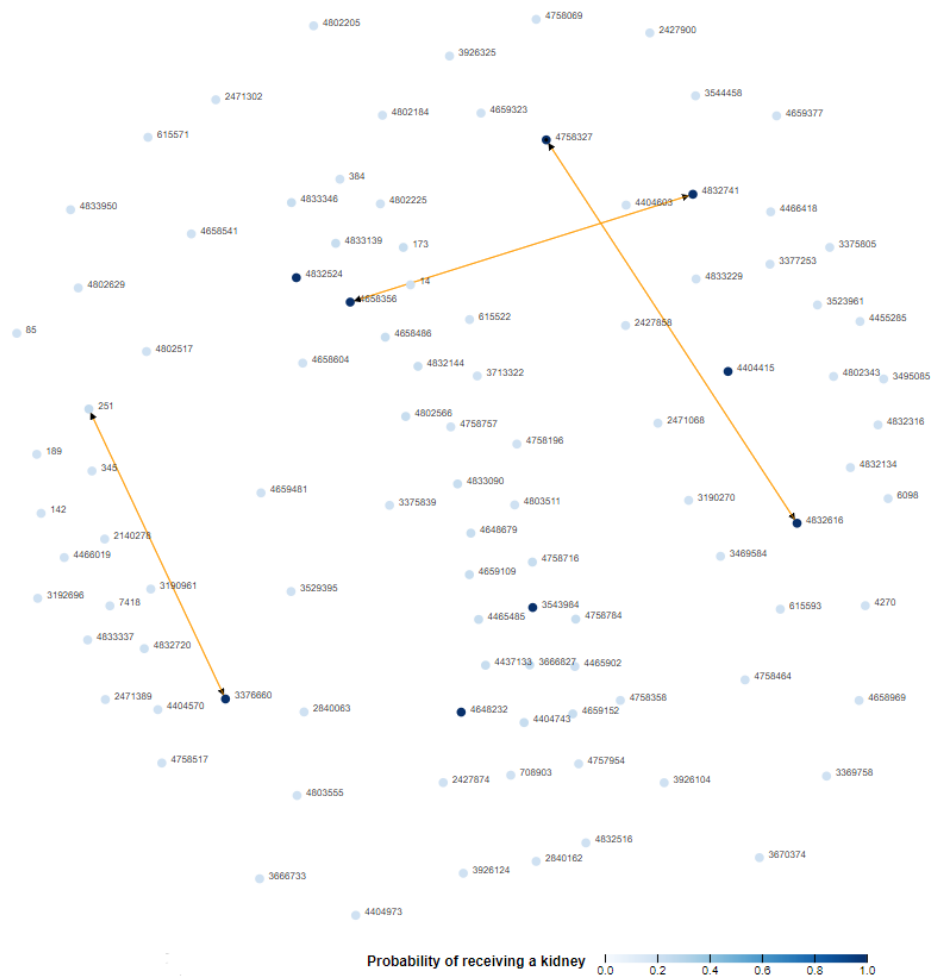


Figura 10.4: Predicció del model (color dels nodes) en un graf de cicles que representa una solució.

Com es pot veure, en aquest cas la predicció del model ha sigut correcte en 96 dels 101 nodes (95% d'encert). Ens els conjunts de dades que tenim disponibles han sortit uns resultats al voltant del 95% d'encerts, tot i que la mostra que tenim és petita i faria falta més dades per poder validar el model.

10.2 Aplicació web

Un dels objectius d'aquest projecte era el desenvolupament d'una aplicació web per visualitzar les dades. L'aplicació resultant al usuaris guardar diferents fitxers JSON amb les dades i visualitzar-los:

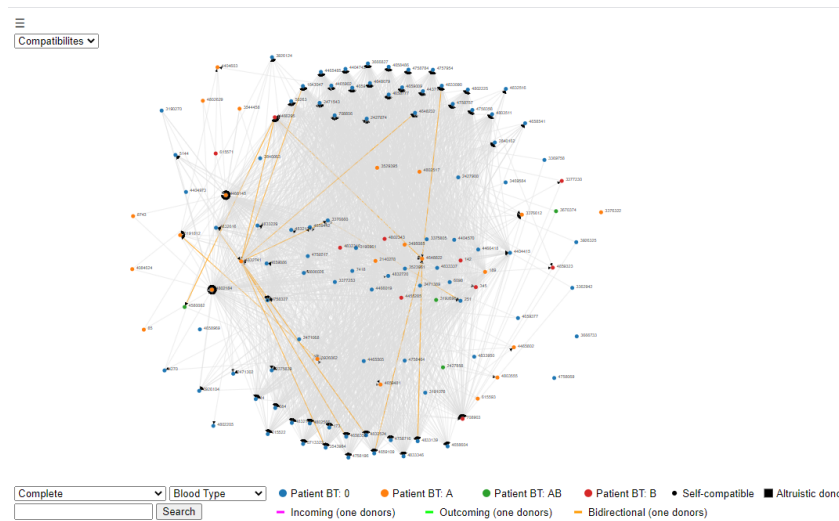


Figura 10.5: Visualització del graf de compatibilitats.

També permet veure en detall parelles de pacient-donant i possibles transplantaments.

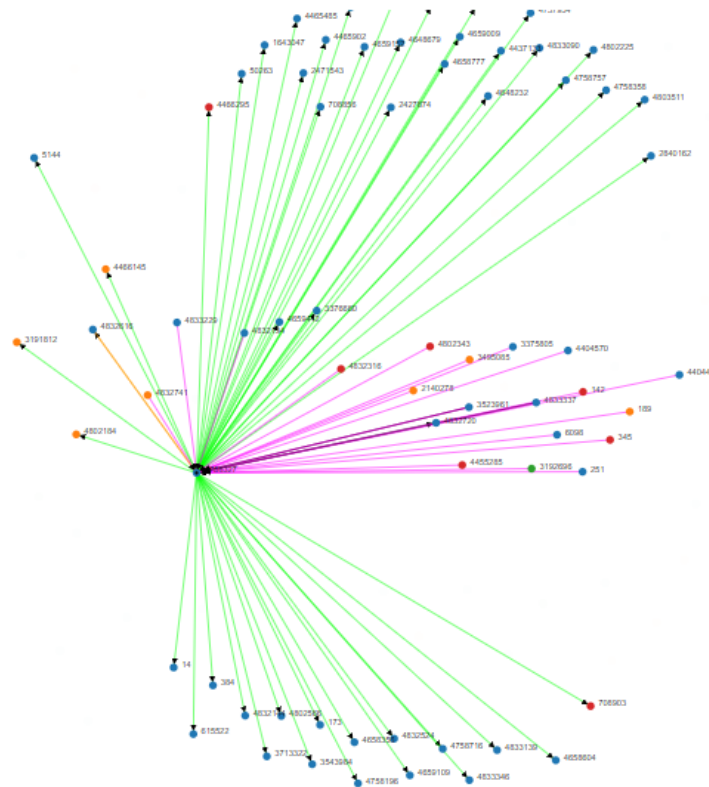


Figura 10.6: Visualització de la selecció d'un node.

A l'aplicació es pot visualitzar els possibles cicles de trasplantaments que es poden realitzar amb el grup de pacients i donants.

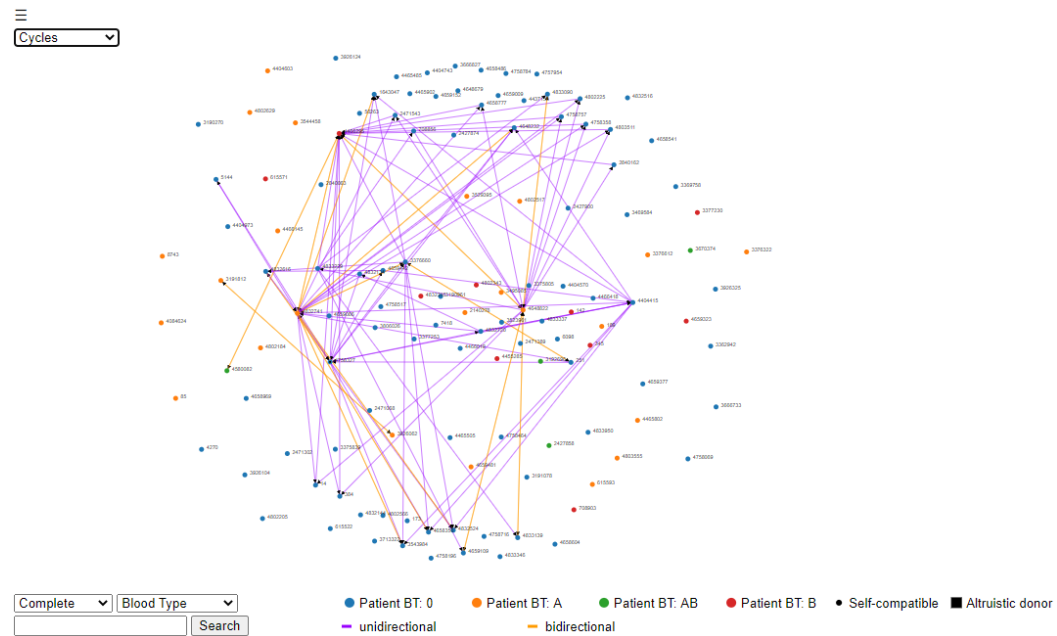


Figura 10.7: Visualització del graf de cicles.

També es pot visualitzar en forma d'histogrames les diferents mesures més significatives a l'apartat d'estadístiques de l'aplicació.

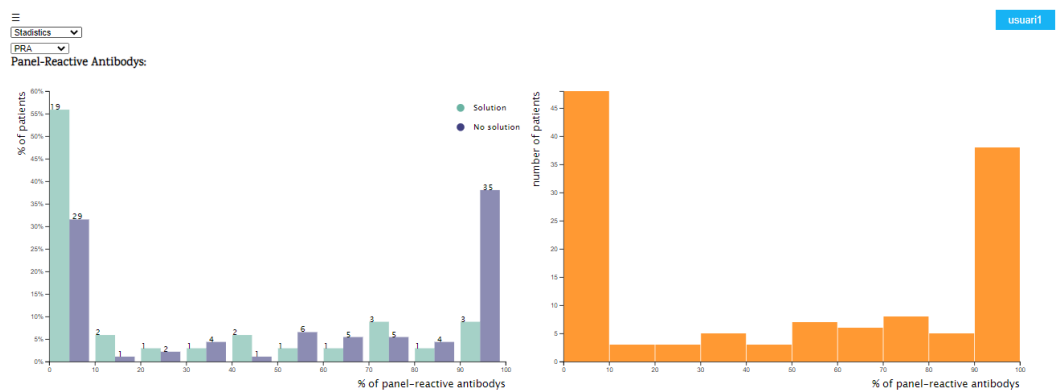


Figura 10.8: Visualització de l'apartat d'estadístiques.

CAPÍTOL 11

Conclusions

L'objectiu inicial d'aquest projecte ha estat el de desenvolupar una aplicació web per a la visualització en forma de graf de les parelles pacient-donant i els possibles trasplantaments d'un programa d'intercanvi de ronyons amb donant viu (KEP).

Es pretenia, d'una banda, permetre als usuaris (principalment personal mèdic) la visualització de dades de receptors i donants, així com les seves compatibilitats i els potencials intercanvis, a partir de dades en format JSON. D'altra banda, un segon objectiu era el d'analitzar les dades disponibles de les parelles donant-receptor, incloent-hi les compatibilitats a priori i els trasplantaments potencialment realitzables, de cares a intentar predir la probabilitat que té un determinat receptor de ser seleccionat en un intercanvi per a trasplantament.

Després d'haver desenvolupat el projecte i d'acord amb els resultats obtinguts, pot afirmar-se que s'ha complert amb els requisits definits inicialment.

En l'àmbit de l'aplicació web s'ha realitzat una aplicació que permet visualitzar el graf de compatibilitats entre les diferents parelles pacient-donant, el graf dels possibles cicles a realitzar i un apartat d'estadístiques per visualitzar diferents mesures en forma d'histogrames.

En l'àmbit del model de regressió logística, s'ha creat un model que permet predir si una parella pacient-donant formarà part del conjunt d'intercanvis seleccionats per realitzar. La probabilitat d'una parella de pertànyer a la solució donada per aquest model és il·lustrada de forma gràfica per l'aplicació amb una escala de colors, a partir dels càlculs obtinguts amb el propi model.

Treball futur

Tot seguit es descriuran algunes possibles futures línies de continuació de la feina desenvolupada al llarg d'aquest projecte.

En primer lloc, hi hauria la possibilitat d'ampliar l'aplicació web perquè permetés escollir diferents nodes i trasplantaments de forma interactiva per crear solucions en format de fitxer de cicles.

Una segona possible millora consistiria a permetre, a través de la pàgina web, la modificació de les dades de donants i receptors. Això permetria afegir noves parelles donant-receptor, eliminar-ne altres ja presents i també modificar les dades de parelles ja presents.

També es podria crear grups d'usuaris que compartissin fitxers de manera que no caldria que cada usuari tingues els seus.

Pel que fa als aspectes més teòrics, es podrien explorar noves mesures de Teoria de Grafs i/o de Teoria de la Informació per tal d'afegir el seu càlcul i, d'aquesta manera, fer més precís el model de regressió logística desenvolupat.

Bibliografia

- [D3.js 020] D3.js. *Llibreria de JavaScript D3.js*, (Consultat: Setembre 2020). Disponible a <https://github.com/d3/d3/wiki>. (Cited on page 25.)
- [Django 021] Django. *Framework de Python Django*, (Consultat: Setembre 2021). Disponible a <https://docs.djangoproject.com/en/3.2/>. (Cited on page 23.)
- [NetworkX 021] NetworkX. *Llibreria de Python NetworkX*, (Consultat: Gener 2021). Disponible a <https://networkx.org/documentation/stable/index.html>. (Cited on page 25.)
- [rpy2 021] rpy2. *Llibreria de Python rpy2*, (Consultat: Març 2021). Disponible a <https://rpy2.github.io/doc/latest/html/introduction.html>. (Cited on page 26.)