

# **INFERÈNCIA CAUSAL GEOMÈTRICA:**

## causalitat com a estructura geomètrica de l'espai-temps

Mètode de conversió de pathways a un model causal  
geomètric 3D i validació empírica multiescala del graf  
resultant

david graupere villà

2025-09-25

1. <INFERÈNCIA CAUSAL GEOMÈTRICA: causalitat com a estructura geomètrica de l'espai-temps >

<Mètode de conversió de pathways a models geomètrics 3D i validació empírica multiescala del graf resultant>

## 2. L'avanç del projecte

2.1 Abans de reportar el grau de compliment, hom en reescriu de forma aclaridora les fases del model fetes fins ara.

### PROJECCIÓ I TRANSFORMACIÓ GEOMÈTRICA (TOPOLOGICA I EMPÍRICA) VEGEU QUADRE SÍNTESI pàgs. 4-6

#### ⇒ Projecció d'un espai empíric d'R^11 a un espai geomètric causal d'R^3

Cada variable empírica és una dimensió pròpria, això vol dir que les dades viuen a dins d'un espai de dimensió R^11 on cada mostra (observació) és un punt  $x=(x_1, x_2, \dots, x_{11})$ , però cap coordenada d'aquest espai és espacial ni causal, és un espai empíric cru. Tanmateix el model geomètric introduceix una estructura que redueix les 11 dimensions a 3, formalment és f:  $R^{11} \rightarrow R^3$  per mitjà d'una immersió topològica o immersió geomètrica causal a fi de fer visible les relacions estructurals (hom entén per topològica les coordenades aspcionals derivades de les tres dimensions que extreuen tota la informació causal del DAG 2D). Dins d'un model topològic, els nodes s'organitzen per nivells i graus; en canvi, en el registre empíric hi circula un volum d'informació crua que recorre la mateixa estructura geomètrica, on no hi ha directament les observacions empíriques, sinó un espai de propietats derivades; cada punt a 3D és ara un node (una variable empírica) i la seves coordenades (X,Y,Z) provenen de la seva estructura causal (grau d'entrada/dependència, grau de sortida/efecte i nivell/jerarquia de flux causal).

#### ⇒ CADENA DE TRANSFORMACIONS GEOMÈTRIQUES A PARTIR DE LES DADES:

Hom afaiçona les coordenades topològiques (les que provenen del pathway sense cicles) segons les dades, és a dir, hom reubica els nodes a partir d'estadístics que resumeixen la informació continguda en les dades: d'aquí que hom aprofiti mesures de tendència central, de variabilitat, de correlació...

#### → Normalització de la distribució espacial causal topològica (escalatge de coordenades)

Perquè els tres eixos (X, Y, Z) representin magnituds comparables de variació causal; és a dir, l'objectiu és que el rang de valors de cada dimensió axial sigui comparable en termes de magnitud amb la resta. Posar X, Y i Z en la mateixa escala numèrica a fi d'evitar biaixos d'unitat de mesura, totes tres dimensions aporten un pes geomètric comparable:

$$X' = \frac{X - \bar{X}}{s_X}; \quad Y' = \frac{Y - \bar{Y}}{s_Y}; \quad Z' = \frac{Z - \bar{Z}}{s_Z} \quad \text{on } s_i \text{ és desviació estàndard o rangs}$$

Per exemple, si el nivell jeràrquic Z domina, el model deixa d'esser un espai isòtrop de causalitat. Cal fer que les dependències entre variables-node siguin comparables en magnitud geomètrica, si les coordenades Z no són escalades respecte de X i Y, llavors les variacions verticals entre nivells tenen molt més de pes geomètric que no pas les desviacions retrolaterals dins d'un mateix nivell i el flux resultant tendeixi sempre més cap a Z encara que les dades no ho justifiquin.

Se'n diu anisotropia geomètrica: un espai on una direcció (Z) té un pes més gran que les altres; **d'aquí que calgui un espai isòtrop de causalitat**, un espai geomètric on totes les direccions causals, siguin verticals o retrolaterals són comparables en magnitud i contribueixen de manera equilibrada al flux resultant. Hom entén <un espai causal isòtrop> com aquell en què cap eix (X, Y o Z) domina el comportament geomètric del model. És evident que a nivells intermigts hi ha molta més presència de connexions retrolaterals que no pas a d'altres nivells, per exemple, al nivell de nodes-arrel on hi ha sempre grau\_IN zero.

Sense normalització, una diferència d'1 unitat en Z pot equivaldre a 100 en X o Y (si les escales són molt diferents) i això fa que qualsevol distància sigui del tot sotmesa per Z, cal que totes tres tinguin el mateix ordre de magnitud (comparteixin la mateixa escala). Per tant, així, les distàncies euclidianes siguin comesurables, és a dir, que reflecteixen relacions reals sense distorsions d'escala. Ara el centre geomètric (0,0,0) és el punt de referència del flux causal global (després el centre de masses serà els valors mitjans de cada eix a fi de treballar sols amb escales positives) i les variacions de cada eix tenen la mateixa unitat de mesura.

#### → normalització de la distribució espacial causal empírica: estadístics per tal de recollir el màxim d'informació empírica de cada variable de les dades i transmetre-la a dins de l'espai geomètric

Cada coordenada dels nodes del model geomètric 3D és una transformació geomètrica d'un subespai empíric multidimensional (no una única variable), en la qual hom pot mesurar direcció, angles, fluxos... Cal traduir les

mesures empíriques de cada variable en termes geomètrics, cosa que implica sintetitzar les dades crues a informació estructural. Per exemple, és el mateix procediment que fan els físics: traduir dades multidimensionals a una representació geomètrica 3D (+temps), cosa que implica una metrització, reduir un espai de moltes de dimensions a un espai 3D amb propietats mètriques que mantenen les relacions causals i dinàmiques.

Amb les dades empíriques no cal fer l'escalatge (centrar i tipificar) per a cada columna-variable tal com és fet abans amb les coordenades topològiques, perquè el rang de valors és molt semblant entre totes onze variables. Havent fet l'escalatge dels eixos del model topològic pur, la normalització de la distribució espacial causal empírica dels nodes és definida segons una cadena d'operacions de transformació geomètrica. Hom aplica aquesta cadena (hom la podria formalitzar com un producte matricial, on cada matriu és una transformació espacial de les coordenades prèvies).

Aquesta cadena d'operacions pot esser tan llarga i elaborada com hom vulgui: aquí hom opta per dues eines bàsiques de l'estadística per a dur a terme dues transformacions a fi d'associar les mesures estadístiques de mitjana i desviació tipus a la posició geomètrica relativa dels nodes i associar la correlació de les variables a la llargària de les arestes del model geomètric.

### **==>NORMALITZACIÓ RADIAL: PRIMERA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI.**

#### **Mitjana/desviació tipus global**

El producte de l'estadístic (definit com a quotient de la mitjana de cada columna, mu\_columna, per desviació tipus de totes les dades, sigma\_global) per les coordenades topològiques escalades perquè hom tradueix en termes geomètrics la mitjana relativa de cada columna com la mitjana de distàncies de totes les arestes pertanyents a un mateix node (vector de mitjanes de les noves coordenades, ara ja topoempíriques, producte de la matriu topològica escalada per la matriu diagonal empírica de mitjanes ponderades), cosa que resitua els nodes a l'espai de forma radial, és a dir, intervé en cada grup radial d'arestes (en resituar el node totes les arestes pertanyents a aquest node també es resituen). Aquesta primera transformació és una normalització radial de les coordenades dels nodes pel quotient mitjana column i desviació típica global aplicat a les coordenades del model topològic escalat. La primera és una reubicació radial de cada node informada per les dades registrades, que reajusta les coordenades topològiques de cada node segons l'evidència empírica, cosa que afecta la posició del node i també totes les seves arestes en bloc, tant entrants com sortints.

La transformació empírica a dins de l'espai causal hom la duu a terme primer per mitjà de la susdita metrització diagonal de les coordenades topològiques, la qual consisteix en multiplicar cada columna de la matriu de coordenades topològiques T (que conté les posicions x, y i z dels nodes del DAG) pel valor mitjà de cada variable-columna, cadascuna dividida per la desviació típica global. Geomètricament, equival a una deformació radial del camp causal: els nodes s'allunyen o s'acosten respecte del centre de masses segons la intensitat empírica, però sense rotacions ni barreja d'eixos.

#### **matriu coords3Dn\_diag= matriu coords\_3Dn · matriu diagonal dades\_diag**

El producte de cada fila de la matriu pel valor corresponent del vector de valors mu/sigma\_global que és equivalent a multiplicar per una matriu diagonal

Per exemple, pel node Mek:

$$\text{coords3Dn} = (0.519, 3.255, 1.864)$$

$$\text{escala} = 6.864877$$

$$\text{coords3Dn\_diag} = (0.519 \cdot 6.864, 3.255 \cdot 6.864, 1.864 \cdot 6.864) = (3.565, 22.347, 12.796)$$

$$M_{\text{coords\_topoem}} = M_{\text{coords\_topo}} \cdot M_{\text{emp\_diagonal}}$$

### **==>NORMALITZACIÓ INDIVIDUAL: SEGONA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI. Correlació**

(per mitjà de l'ajustament de la llargària d'aresta per aresta)

A partir de la Matriu de correlació d'arestes, hom treballa amb dataframes perquè cal identificar cada aresta entre nodes (a més a més de les coordenades de cada node) i això és més fàcil de fer directament (l'objectiu ara no és automatitzar el procés sinó calcular les noves i difinitives coordenades un cop feta la segona transformació geomètrica)

$$M_{\text{corr\_arestes}} = M_{\text{correlacions}} \cdot M_{\text{adjacent}}$$

$$DF_{\text{refòs}} = DF_{\text{corr\_arestes}} + DF_{\text{coords\_topoemp1}}$$

La correlació entre variables, un refinament més de caràcter local, és associada per a cada aresta individual, perquè hom tradueix en termes geomètrics la correlació com a longitud del mòdul (mòdul més llarg, més de força o d'associació), cosa que intervé en cada aresta de forma individual. Aquesta segona associació de la correlació amb les arestes sols és possible per als nodes que són lligats per una aresta; la correlació és com un factor d'allargament o escurçament de les arestes individuals. Per exemple, la longitud de les arestes (d'on es deriven els mòduls, atès que el mòdul és una expressió de força causal efectiva) pugui esser entesa segons la informació empírica a fi que el flux causal reflecteixi la força real de transmissió observada i no sols la distància topològica empírica a priori.

La segona metrització diagonal reescala, doncs, localment i de manera independenta cada aresta, sense alterar la jerarquia causal ni la connexió estructural entre nodes. En canvi, una metrització plena (no diagonal) pot descriure dependències creuades entre nodes sense aresta, donant lloc a un espai causal anisotòpic. Tanmateix el model aquí presentat empra una versió filtrada, en què els valors empírics sols s'apliquen a les arestes existents al model geomètric topològic, cosa que garanteix que les correccions empíriques deformen localment el camp causal.

Així, la primera transformació implica una deformació en la radialitat de cada node, és a dir, totes les arestes (no importa si l'aresta és in o out) en són afectades; en canvi, cada valor de la segona transformació afecta una única aresta entre dos nodes adjacents (l'origen és fix, s'allarga o s'escurça en el sentit de l'aresta).

Així, totes dues transformacions geomètriques defineixen la variabilitat local geomètrica topològica com la suma de la variabilitat radial d'un node i la variabilitat internodal de cada aresta; i la variabilitat total com la suma de totes les radials i les seves respectives internodals.

Totes dues transformacions consecutives vers les coordenades topològiques escalades tenen com a punt de referència un centre de masses (vector de mitjanes de les coordenades calculat després d'haver fetes les transformacions) que mostra com la xarxa de punts s'expandeix o es contreu respecte d'aquest centre. En aquest estudi, una gran part dels nodes s'expandeix, també hi han acostaments al centre de masses però són molt tímids, és clar que hi ha una expansió, un allunyament del centre de masses i els nodes guanyen pes i visibilitat.

Cal afegir que hi han moltes de possibilitats perquè la cadena d'operacions pugui esprémer de forma òptima tota la informació de les dades i traduir-la en termes geomètrics. Hom pot trobar la traducció en termes geomètrics de les dades empíriques a angles, a superfícies... El disseny de la metrització podria aprofitar els signes de cada transformació a fi de fer una lectura més sofisticada de la configuració final de les coordenades topoempíriques finals. També, si hi han dades d'altres assaigs que fossin compatibles, hom podria aprofitar-les per seguir amb la cadena d'operacions, aprofitar tota la informació numèrica que hi ha a l'abast.

- ⇒ **Normalització vectorial** transforma cada vector en un vector unitari, cosa que és del tot incoherent, sols justificable si els angles són a prop de  $0^\circ$  o de  $90^\circ$  (segons una geometria non-euclidiana, vegeu Annex1), els quals poden esser problemàtics per al càlcul d'angles, cosa que hom pot resoldre fent una anàlisi prèvia dels angles continguts al model i establir un valor determinat si n'hi ha cap, angle de risc. La normalització unitària ---tot i que no distorsiona els angles entre vectors--- no sembla adient perquè elimina tota la informació de força causal (el mòdul real). Un vector unitari no conté ni profunditat jeràrquica ni intensitat causal, sols és en essència direccional, sols dóna orientació.

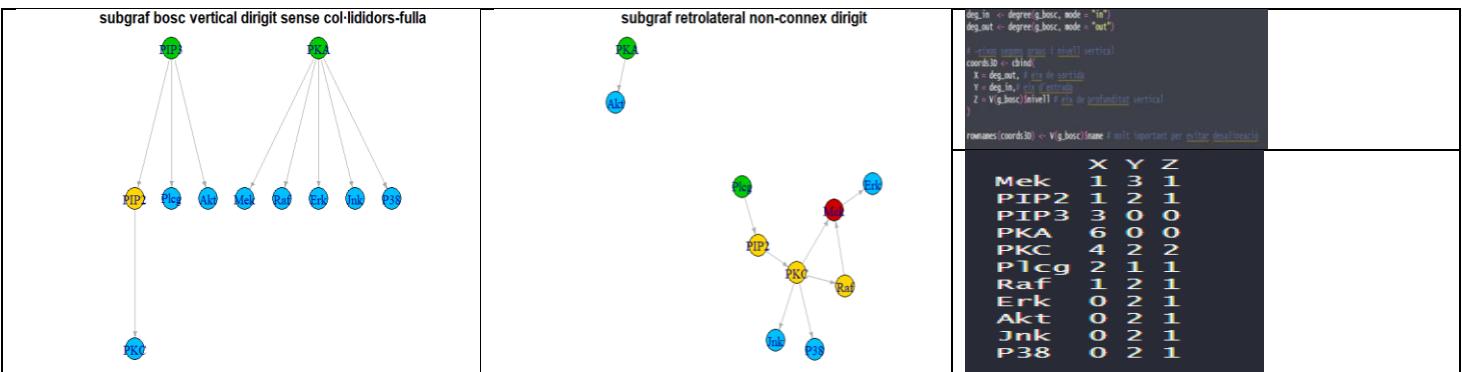
**La metàfora bayesiana:** la informació topològica juga el mateix paper que una prèvia (prior) en inferència bayesiana, és a dir, el coneixement causal previ, allò que sabem del sistema abans de registrar les dades específiques; és empíric (prové d'assaigs previs o de la literatura biològica), però no és pas numèric (no té magnituds o intensitats assignades), és una estructura empírica non-quantitativa. Les dades de l'estudi són l'evidència observada (l'equivalent a la versemblança o likelihood) perquè contenen magnituds reals (nivells d'expressió) que poden refermar, refinar o contradir parcialment la distribució espacial topològica prèvia. A la fi, el procés geomètric, aquí anomenat geomètric causal o espai topològic-empíric, és una posteriorització. El model estima una distribució espacial a posteriori de coherència geomètrica entre la topològica i l'empírica.

→ **EMPÍRICA ESTRUCTURAL** (de natura qualitativa o relacional): aquesta és la que va del pathway al DAG 2D o al model topològic 3D. La informació a priori és el que Pearl anomena "història causal" ("background causal knowledge"), és una informació estructural prèvia, que no és numèrica però oísí empírica, atès que prové de l'experiència científica acumulada, de l'observació en diversos assaigs i de coneixement del mecanisme biològic. Hom parteix d'un coneixement empíric de base, és informació empírica non-numèrica però sí factual.

→ **EMPÍRICA QUANTITATIVA** (de natura numèrica o observacional): les dades afaïconen les coordenades 3D topològiques en termes d'estructura, de força de flux (la longitud de les arestes), etc.

D'aquí que el núvol de nodes connectats per arestes dirigides és en si mateix significatiu; les mesures de cada angle, mòdul, sentit... són per si mateixes explicatives perquè el flux d'informació correspon no sols a l'eix vertical Z (flux general vertical extret en essència del subgraf vertical), sinó també a la connectivitat topològica d'un node (graus\_in i graus\_out), la qual és traduïda a dins d'un espai angular local definit pels vectors de flux incident i emergent; aquesta estructura angular també condiciona la divergència o convergència del flux causal vertical i, per tant, la seva distribució a la coherència geomètrica global del sistema.

1.PROJECCIÓ D'UN ESPAI EMPÍRIC D'R^11 A UN ESPAI GEOMÈTRIC CAUSAL D'R^3		
Pathway article	DAG 2D amb cicles	DAG 2D sense cicles
	<p>pathway o network original</p>	<p>pathway sense cycles</p>
Descomposició del DAG 2D en els subgrafs vertical i retrolateral (SÓN COMPLEMENTÀRIAMENT EXCLOENTS)		
<p>→les arestes verticals són triades segons:</p> <ul style="list-style-type: none"> <li>*la definició d'arbre/bosc de la teoria de grafs</li> <li>**l'aresta vertical més informativa segons localTest() quan hi han col·lididors-fulla verticals i cal triar quina aresta roman al vertical i quina va al retrolateral (quin model amb l'aresta absent del col·lididiro-fulla obté menys de tests negatius) vegeu Annex1 Codi</li> </ul> <p>→tota la resta d'arestes que no pertany al vertical va al retrolateral</p> <p>→hi ha un espai de transició entre tots dos subgrafs (els nodes-frontera)</p>		
subgraf bosc vertical dirigit amb col·lididors-fulla que contravenen la definició d'arbre/bosc	subgraf retrolateral non-connex dirigit sense les arestes dels col·lididors fulls	Espai de transició entre els subgrafs vertical i retrolateral: els nodes-frontera
		Els nodes-fulla del subgraf vertical són també els nodes-frontera de l'espai de transició que uneixen els dos subgrafs i són tots col·lididors retroverticals; els col·lididors que surten al subgraf retrolateral són o bé retrolaterals o bé retrolateroverticals, tenen sempre com a mínim una aresta lateral, però tota la resta són col·lididors sense lateralitat i sols són visibles si hom té en compte aquest espai de transició entre els dos subgrafs de la descomposició, és a dir, els nodes comuns a tots dos subgrafs (d'aquí que hom en dedueix que el mínim de nodes que pot tenir el subgraf retrolateral és el nombre de fulles del subgraf vertical). D'aquí que la descomposició hagi trencat tots els col·lididors retroverticals i sols hagin romàs els que tenen com a mínim una aresta lateral.
<pre> g_bosc_vertical1 &lt;- subgraph, from: edge(g_bosc_vertical, id= c(1:4,6:10)) arrels &lt;- V(g_bosc_vertical1)\$edges(degree(g_bosc_vertical1, mode="in") == 0 fulles &lt;- V(g_bosc_vertical1)\$edges(degree(g_bosc_vertical1, mode="out") == 0 noms_nodes1 &lt;- V(g_bosc_vertical1\$name) V(g_bosc_vertical1)\$role &lt;- lapply(lis.namatch(noms_nodes1, noms1), "arell",                                     lisef(lis.namatch(noms_nodes1, fulles1)), "fulla", "mijancer") V(g_bosc_vertical1)\$color &lt;- ifelse(V(g_bosc_vertical1)\$role == "arell", "#green3",                                     ifelse(V(g_bosc_vertical1)\$role == "fulla", "#blue",   "#gold")) ))  par(mar=c(0,0,1,0)) plot(g_bosc_vertical1,       layout=layout_treemap(g_bosc_vertical1, root = arrels),       edge.arrow.size=0.4,       vertex.size=15,       main ="subgraf bosc vertical dirigit sense col·lididors-fulla") </pre>	<pre> set.seed(777) par(mar=c(0,0,1,0)) arrels &lt;- V(g_retrolateral1)\$edges(degree(g_retrolateral1, mode="in") == 0 fulles &lt;- V(g_retrolateral1)\$edges(degree(g_retrolateral1, mode="out") == 0 noms_nodes &lt;- V(g_retrolateral1)\$name V(g_retrolateral1)\$role &lt;- lapply(lis.namatch(noms_nodes, arrels), "arell",                                     lisef(lis.namatch(noms_nodes, fulles)), "fulla", "mijancer")) V(g_retrolateral1)\$color &lt;- ifelse(V(g_retrolateral1)\$role == "col·lididor", "red3",                                     ifelse(V(g_retrolateral1)\$role == "arell", "#green3",   ifelse(V(g_retrolateral1)\$role == "fulla", "#blue",   "#gold"))) ))  plot(g_retrolateral1,       layout=layout_treemap(g_retrolateral1),       edge.arrow.size=0.4,       vertex.size=15,       main ="subgraf retrolateral non-connex dirigt") </pre>	Coords Matriu Topològica (hom entén topològica com les dimensions extrems de la descomposició del DAG 2D: vertical, lateral i retroflux ) Al codi R: coords3D



## 2.PRIMERA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI

(per mitjà de l'ajustament de les coordenades del pathway de cada node pel valor de l'estadístic mu/sigma\_global calculat en les dades)

**matriu coords3Dn\_diag = matriu coords\_3Dn - matriu diagonal dades\_diag**

El producte de cada fila de la matriu pel valor corresponent del vector de valors mu/sigma\_global que és equivalent a multiplicar per una matriu diagonal

Per exemple, pel node Mek:

coords3Dn = (0.519, 3.255, 1.864)

escala = 6.864877

$$\text{coords3Dn\_diag} = (0.519 \cdot 6.864, 3.255 \cdot 6.864, 1.864 \cdot 6.864) = (3.565, 22.347, 12.796)$$

$$M_{\text{coords\_topo}} = M_{\text{coords\_topo}} \cdot M_{\text{emp\_diagonal}}$$

	Normalitzats: eixos de rang comparable	Vector (de mitjanes/desviació tipus global) com a matriu diagonal
al codi R: coords3Dn_diag	al codi R: coords_3Dn	<pre>Mek 6.864877 6.462049 5.916116 6.911364 6.642788 6.843741 7.099163 6.096218 6.337969 6.490635 6.457</pre>
<pre>m_means &lt;- colMeans(dades, na.rm = TRUE) sigma_global &lt;- sd(unlist(dades), na.rm = TRUE) dades_diag &lt;- dades_diag[rownames(coords3Dn)] coords3Dn_diag &lt;- coords3Dn[dades_diag]</pre>	<pre>#AQUÍ HOM CENTRA I TIPIFICA ELS EIXOS TOPOLOGICS coords3Dn &lt;- scale(coords3Dn) #coords3Dn &lt;- apply(coords3Dn, 2, function(col) col/max(abs(col))) #coords3Dn &lt;- apply(coords3Dn, 2, function(col) col-min(col)-1e-2)</pre>	<pre>m_means &lt;- colMeans(dades, na.rm = TRUE) sigma_global &lt;- sd(unlist(dades), na.rm = TRUE) dades_diag &lt;- m_means/sigma_global dades_diag &lt;- dades_diag[rownames(coords3Dn)]</pre>
<pre>X Y Z [1,] 3.56525080 22.34717517 12.79647227 [2,] 3.35604330 14.04544137 12.04558036 [3,] 9.09921518 0.05916116 0.05916116 [4,] 21.19079368 0.06911364 0.06911364 [5,] 13.60035355 14.43828175 24.69854441 [6,] 7.04011000 7.47174833 12.75707306 [7,] 3.68692626 15.43022601 13.23319236 [8,] 0.06096218 13.25029828 11.36365377 [9,] 0.06337796 13.77575030 11.81428927 [10,] 0.06490635 14.10757318 12.09886554 [11,] 0.06457962 14.03655694 12.03796095</pre>	<pre>X Y Z Mek 0.5193466 3.255291 1.864050 PIP2 0.5193466 2.173527 1.864050 PIP3 0.51380398 0.010000 0.010000 PKA 0.36660796 0.010000 0.010000 PKC 0.20473864 2.173527 3.718099 Plcg 0.10286932 1.091764 1.864050 Raf 0.5193466 2.173527 1.864050 Erk 0.0100000 2.173527 1.864050 Akt 0.0100000 2.173527 1.864050 Jnk 0.0100000 2.173527 1.864050 P38 0.0100000 2.173527 1.864050</pre>	

## 3.SEGONA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI

(per mitjà de l'ajustament de la llargària d'aresta per aresta)

$$M_{\text{corr\_arestes}} = M_{\text{correlacions}} \cdot M_{\text{adjacent}}$$

$$DF_{\text{refós}} = DF_{\text{corr\_arestes}} + DF$$

Matriu de les correlacions on hi ha arestes R_filt	Matriu de correlacions entre nodes A	Matriu d'adjacències (nodes units per aresta) R
<pre># filtre per estructura: sols on hi ha aresta (element a element) R_filt &lt;- R * A + zeros on no hi ha aresta</pre>	<pre>nodes &lt;- rownames(coords3Dn) A &lt;- as.matrix(as_adjacency_matrix(g_bosc, type = "both", attr = NULL))</pre>	<pre>#matriu de correlacions empiriques entre variables I &lt;- cor(dades[, nodes], drop = FALSE, use = "pairwise.complete.obs", method = "pearson")</pre>

Matriu de les correlacions on hi ha arestes R_filt	Matriu de correlacions entre nodes A	Matriu d'adjacències (nodes units per aresta) R
<pre>Mek PIP2 PIP3 PKA PKC Plcg Raf Erk Akt Jnk P38 Mek 0 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 PIP2 0 0 -0.07702225 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 PIP3 0 0 0.0000000 0.2295123 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 PKA 0.4933645 0.0000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 PKC 0.66869487 0.0000000 0 0.0000000 0.0238052 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 Plcg 0.0000000 0.1221024 0 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 Raf 0.7375410 0.0000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 Erk 0.0000000 0.2000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 Akt 0.0000000 0.0000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 Jnk 0.0000000 0.0000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 P38 0.0000000 0.0000000 0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000</pre>	<pre>Mek PIP2 PIP3 PKA PKC Plcg Raf Erk Akt Jnk P38 Mek 1.0000000 0.04214902 -0.06659628 0.04836464 0.06869870 -0.0598962 0.73275499 0.08364466 0.07149276 0.02516495 0.12 PIP2 0 0 0.0000000 0.25912424 0.01974636 0.6427788 0.02203000 0.0000000 0.0000000 0.0000000 0.0000000 0.00 PIP3 0 0 1.0000000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.00 PKA 0 1 0 0 0 0 0 0 1 0 0 0 0 PKC 0 1 0 0 0 0 0 0 0 1 1 1 Plcg 0 0 0 0 0 0 0 0 0 0 1 0 0 Raf 0 1 0 0 0 0 0 0 0 0 0 0 0 Erk 0 0 0 0 0 0 0 0 0 0 0 0 0 Akt 0 0 0 0 0 0 0 0 0 0 0 0 0 Jnk 0 0 0 0 0 0 0 0 0 0 0 0 0 P38 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>	<pre>Mek PIP2 PIP3 PKA PKC Plcg Raf Erk Akt Jnk P38 Mek 1.0000000 0.04214902 -0.06659628 0.04836464 0.06869870 -0.0598962 0.73275499 0.08364466 0.07149276 0.02516495 0.12 PIP2 0 0 0.0000000 0.25912424 0.01974636 0.6427788 0.02203000 0.0000000 0.0000000 0.0000000 0.0000000 0.00 PIP3 0 0 1.0000000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.0100000 0.00 PKA 0 1 0 0 0 0 0 0 1 0 0 0 PKC 0 1 0 0 0 0 0 0 0 1 1 1 Plcg 0 0 1 0 0 0 1 0 0 0 0 0 Raf 0 1 0 0 0 0 0 0 0 0 0 0 Erk 0 0 0 0 0 0 0 0 0 0 0 0 Akt 0 0 0 0 0 0 0 0 0 0 0 0 Jnk 0 0 0 0 0 0 0 0 0 0 0 0 P38 0 0 0 0 0 0 0 0 0 0 0 0</pre>

A partir de la Matriu de correlació d'arestes, hom treballa amb dataframes perquè cal identificar cada aresta entre nodes (a més a més de les coordenades de cada node) i això és més fàcil de fer directament (l'objectiu ara no és automatitzar el procés sinó calcular les noves i difinitives coordenades un cop feta la segona transformació geomètrica)

Dataframe final refosa

```
df_fos <- R_filt %>% left_join(edges_df, by=c("des", "fins"))
df_fos$llarg_nova <- (df_fos$corr*df_fos$length) + df_fos$length
df_fos$prop <- df_fos$llarg_nova/df_fos$length
(df_fos <- df_fos[order(df_fos$llarg_nova, decreasing = TRUE),])
```

Dataframe

Hom calcula la llargària de cada aresta segons la darrera matriu de coordenades <coords3Dn\_diag>  
edges <- as\_edgelist(g\_bosc, names = TRUE)  
# funció d'utilitat per a longituds  
edge\_lengths <- function(coords3Dn\_diag, edges){  
 des <- edges[,1]

Dataframe corresponent a la matriu de les correlacions on hi ha arestes R\_filt

```
R_filt_df <- as.data.frame(as.table(R_filt))
colnames(R_filt_df) <- c("des", "fins", "corr")
R_filt_df <- R_filt_df[R_filt_df$corr != 0]
R_filt_df
```

	<pre>fins &lt;- edges[,2] p1 &lt;- coords3Dn_diag[des , , drop = FALSE] p2 &lt;- coords3Dn_diag[fins , , drop = FALSE]  vec &lt;- p2 - p1 lon &lt;- sqrt(rowSums(vec^2))  data.frame(des = des, fins = fins, dx = vec[,1], dy = vec[,2],            dz = vec[,3], lon = lon, row.names = NULL))</pre>	
> df_fos	<pre>des fins corr dx dy dz length llarg_nova prop 12 PKA Erk 0.259596487 -27.23234085 15.540608 14.455625 34.52645 43.48940 1.2595965 14 PKA Akt 0.276368066 -25.88097377 16.774069 14.3736461 34.07640 41.72889 1.2765817 18 PKC P38 0.488494531 -18.4809443 15.507105 -19.2568142 26.71118 39.75944 1.4884945 1 PKA Nek 0.048366457 -19.23437971 29.305466 11.4877461 36.88821 38.67236 1.0483665 15 PKA Jnk 0.087830114 -26.27883784 15.461841 10.650759 32.25265 35.08921 1.0878301 9 PKA Raf 0.036694411 -19.21457309 16.695559 14.0411859 29.07054 31.50286 1.0836694 4 PIP3 PIP2 0.225912342 -7.0147212 18.211423 14.7273656 24.62794 30.19169 1.2259123 16 PKC Jnk 0.210998302 -18.62499795 1.004483 14.0154671 23.33094 28.25373 1.2109983 17 PKA P38 -0.059613508 -26.0847842 12.950253 5.4394038 29.62621 27.86008 0.9403865 13 PIP2 Akt -0.002099523 -13.7840108 16.784027 14.3835986 26.05262 25.99792 0.9979005 2 PKC Mek 0.068699870 -11.6305398 14.848108 -13.1784719 23.00887 24.58957 1.0686999 3 Raf Mek 0.732754099 -0.01906062 12.609907 -2.5534398 12.86585 22.29336 1.7327541 8 PIP3 Plcg 0.135894703 -2.33883394 8.419911 14.4234909 16.86424 19.1568 1.1358947 11 Mek Erk 0.08364460 -7.99796115 -13.764851 2.9678764 16.19404 17.54842 1.0836446 10 PKC Raf 0.021805821 -11.11073320 2.238201 -10.6250221 15.89685 16.42350 1.0218058 6 PIP2 PKC -0.00770225 12.10245403 -3.744112 9.948808 16.10796 15.98390 0.9929789 7 Plcg PKC 0.011182880 6.047392 10.2526796 13.72194 13.87540 1.0111828 5 PIP3 PIP2 0.122102384 -5.27578788 9.791505 0.3038747 11.12654 12.48511 1.1221024</pre>	<pre>edges_df &lt;- edge_lengths(coords3Dn_diag, edges)  &gt; edges_df    des fins corr dx dy dz length 4  PKA  Mek 0.048366457 5  PKC  Mek 0.068699870 7  Raf  Mek 0.732754099 14 PIP3  PIP2 0.225912342 17 Plcg  PIP2 0.122102384 46 PIP2  PKC -0.007702225 50 Plcg  PKC 0.011182880 58 PIP3  Plcg 0.135894703 70  PKA  Raf 0.083669411 71  PKC  Raf 0.021805821 78  Mek  Erk 0.083634460 81  PKA  Erk 0.259596487 91  PIP3  Akt -0.002099523 92  PKA  Akt 0.226368066 103 PKA  Jnk 0.087830114 104 PKC  Jnk 0.210998302 114 PKA  P38 -0.059613508 115 PKC  P38 0.488494531</pre>

### Finalment les coordenades finals

coords3Dn_diag_corr2	coords3Dn_diag_corr1	coords3Dn_diag (ja vist abans )
coords3Dn_diag_corr3		
coords3Dn_diag_corr4		
coords3Dn_diag_corr5		

(coords3Dn_diag_corr2 <- sweep(coords3Dn_diag_corr1, 2, offset2, "-")) (coords3Dn_diag_corr3 <- sweep(coords3Dn_diag_corr1, 2, offset3, "-")) (coords3Dn_diag_corr4 <- sweep(coords3Dn_diag_corr1, 2, offset4, "-")) (coords3Dn_diag_corr5 <- sweep(coords3Dn_diag_corr1, 2, offset5, "-"))	<pre># reajust de la llargària de cada aresta segons la correlació (columna 'prop')  # hom itera per cada aresta for (i in seq_len(nrow(df_fos))) {   des &lt;- df_fos\$des[i]   fins &lt;- df_fos\$fins[i]    # indexs dels nodes   v1 &lt;- which(rownames(coords3Dn_diag) == des)   v2 &lt;- which(rownames(coords3Dn_diag) == fins)    # posicions originals   p1 &lt;- coords3Dn_diag[v1, ]   p2 &lt;- coords3Dn_diag[v2, ]    # vector direccional (de &lt;des&gt; cap a &lt;fins&gt;)   v &lt;- p2 - p1    # Factor d'ajust   f &lt;- df_fos\$prop[i]    # Nova posició del node d'arribada, allarga/escurça segons el signe del factor   p2_new &lt;- p1 + v * f    # Hom actualitzat coordenades del node de destinació ==&gt;   # SOLS EL NODE DESTINACIÓ   coords3Dn_diag[v2, ] &lt;- p2_new }  min &lt;- apply(coords3Dn_diag, 2, min)</pre>	<pre># hom pesa amb dades empíriques (quotient de la mitjana de cada columna per la desviació típica global) m_means &lt;- colMeans(dades, na.rm = TRUE) sigma_global &lt;- sd(unlist(dades), na.rm = TRUE) dades_diag &lt;- m_means/sigma_global dades_diag &lt;- dades_diag/rownames(coords3Dn) coords3Dn_diag &lt;- coords3Dn\$dades_diag # coincideix que el producte per files és correcte: (N)^{-1} * N coords3Dn_diag &lt;- diag(dades_diag) %*% coords3Dn # el producte de cada fila de la matriu de coordenades per la matriu diagonal cal formalitzar-la així rownames(coords3Dn_diag) &lt;- rownames(coords3Dn)</pre>
---	--	--

És la mateixa taula de coordenades coords3Dn_diag_corr1 desplaçada a l'espai	<pre>X Y Z Mek 14.448295 41.866928013 12.083315 PIP2 13.174897 23.883814098 21.965677 PIP3 23.325302 0.000000000 3.707198 PKA 35.416874 0.009952487 3.717150 PKC 27.785274 14.583345512 28.424531 Plcg 20.668520 9.564140838 20.090765 Raf 14.308713 18.180365864 18.728196 Erk 0.000000 17.721297718 22.748539 Akt 3.718526 20.537924397 21.307501 Jnk 2.468439 17.328291665 12.543498 P38 2.397436 11.495871330 0.000000</pre>	<pre>X Y Z Mek 3.56525080 22.34717517 12.79647227 PIP2 3.35604330 14.04544137 12.04558036 PIP3 9.09922158 0.05916116 0.05916116 PKA 21.19079368 0.06911364 0.06911364 PKC 13.60035355 14.43828175 24.69854441 Plcg 7.04011000 7.47174833 12.75707306 Raf 3.68692626 15.43022601 13.23319236 Erk 0.06096218 13.25029828 11.36365377 Akt 0.06337969 13.77575030 11.81428927 Jnk 0.06490635 14.10757318 12.09886554 P38 0.06457962 14.03655694 12.03796095</pre>
--	--	---

## 2.2 Grau de compliment dels objectius i resultats previstos en el pla de treball:

La validació del model geomètric causal 3D amb acàrament dels resultats d'aquest model o bé amb el DAG 2D o bé amb els resultats de l'article (en quina mesura convergeixen) ha estat de caràcter local i limitada a casos concrets

## VALIDACIÓ LOCAL DEL MODEL CAUSAL GEOMÈTRIC

Atès que hom parteix de l'evidència empírica de localTest() del DAG en 2D i dels resultats de l'article de Sachs et al., hom assaja estratègies de càlcul o d'aproximació del flux causal i del seu subsegüent anàleg geomètric localGeomTest() i comparar els resultats de localTest() amb els resultats del comportament del model geomètric (fins i tot s'assaja amb la creació d'una funció localGeomTest() ).

És per això que hom pot calcular:

--> o bé l'angle del flux causal holístic sols respecte de Z (un únic angle respecte l'eix Z) i comparar-lo amb l'angle de flux local a fi de comprovar que que aquest darrer convergeix al flux causal general

--> o bé els angles del flux causal holístic respecte de Z i YX (dos angles a l'esfera de direccions, un respecte l'eix Z i el segon respecte del pla YX) i comparar-los amb els angles del flux local en estudi a fi de comprovar que que aquest darrer convergeix al flux causal general

Havent determinat l'angle o angles (segons si el flux és calculat o bé sols respecte Z o bé segons Z i YX) segons el comportament angular del model geomètric, hom els compara els resultats amb localTest() i/o els resultats de l'article de Sachs et al.; és a dir, a partir de la funció localTest() del paquet daggity, el qual analitza la independència condicional de nodes non-adjacents, hom obté un resultat del test i hom malda per analitzar si el resultat és coherent amb el comportament del model geomètric.

La funció impliedConditionalIndependencies() és molt útil perquè llista per a un model de DAG 2D totes les independència condicionada; en el cas d'estudi són 59 tests, és moltíssima de feina perquè hi han estructures de nodes no adjacents que inclouen fins a 3 nodes 3 nodes.

## VALIDACIÓ DE L'ARESTA ABSENT DE L'ARTICLE DE SACHS ET AL.

També a la fase d'anàlisi de resultats del model causal geomètric amb relació a la tesi de l'article de Sachs et al. (el descobriment d'una aresta latent al DAG 2D)

L'anàlisi en què hom centra l'anàlisi és la via Akt $\parallel$ Erk | PKA perquè no sols pertany a la llista de independències condicionals implicades sinó que també és de vital importància per a l'anàlisi de la tesi de l'article de Sachs et al. en què assereix l'existència d'una aresta absent entre Akt i Erk.

Per a tal fi, caldrà definir una regió conoïdal de convergència geomètrica entre el flux causal holístic i els fluxos causals locals definits pel vector director de la cadena Raf->Mek->Erk i la forca Akt<-PKA->Erk (vegeu punt 2.2)

Analitzant l'estructura geomètrica del model causal topoempíric hom en descriu els trets que contextualitzen l'aresta absent reportada per Sachs et al.:

--> Hi han dues vies de flux causal encapçalades pels dos nodes arrels dels dos arbres del bosc: PIP3 i PKA

--> PIP3 no té cap camí de flux cap a una fulla pròpria final (PKC acaba essent un node medial), sols un node col·lididor-fulla vertical amb PKA

--> PKA té un únic camí de flux cap a una fulla pròpria final però retro lateralitzada per la cadena Raf->Mek->Erk, la resta de nodes finals són col·lididors retroverticals (cosa que debilita en gran

mesura el flux tant de PKA com de PIP3 però els seus fluxos de sentit contrari xoquen als nodes col·lididors).

Concloent, els dos nodes terminals de flux verticals finals que semblen d'entrada operatius són un col·lididor-fulla (Akt) de PIP3 i PKA i un node-fulla (Erk) però retrolateritzat de PKA. Hom ja veu que el node PKC és el taló d'Aquil·les del model justament perquè és el principal obstacle del flux causal vertical.

**LA QÜESTIÓ CLAU** és si el node fulla vertical retrolateritzat Erk és connectat a l'únic node col·lididor de flux vertical. L'efecte dels nodes arrels convergeix o no a Akt? Caldrà veure si l'angle de flux general respecte de les cadenes Raf->Mek->Erk retrolateral i/o la forca Raf->Mek->Erk lateral convergeixen.

impliedConditionallIndependencies(pathway_net)	localTests(pathway_net, data = dades, type = "cis")
Una anàlisi local completa del cas en estudi segons el model gràfic, aquestes independències condicionals haurien d'esser validades empíricament amb les dades; aquestes són les 59 relacions esperables segons el model	Ara hom veu quines independències condicionades es compleixen i quines no. El primer test de tots rebutja amb molta d'evidència empírica que hi hagi independència entre Akt i Erk, fins i tot quan hom condiciona per PKA.

```

  ...{r}
length(ImpliedConditionalIndependencies(pathway_net))
impliedConditionalIndependencies(pathway_net)
...
[1] 59
Akt _||_ Erk | Mek, PKA
Akt _||_ Erk | PKA, PKC
Akt _||_ Erk | PIP2, PKA, Plcg
Akt _||_ Erk | PIP3, PKA
Akt _||_ Jnk | PKA, PKC
Akt _||_ Jnk | PIP2, PKA, Plcg
Akt _||_ Jnk | PIP3, PKA
Akt _||_ Mek | PKA, PKC
Akt _||_ Mek | PIP2, PKA, Plcg
Akt _||_ Mek | PIP3, PKA
Akt _||_ P38 | PKA, PKC
Akt _||_ P38 | PIP2, PKA, Plcg
Akt _||_ P38 | PIP3, PKA
Akt _||_ PIP2 | PIP3
Akt _||_ PKC | PIP2, Plcg
Akt _||_ PKC | PIP3
Akt _||_ Plcg | PIP3
Akt _||_ Raf | PKA, PKC
Akt _||_ Raf | PIP2, PKA, Plcg
Akt _||_ Raf | PIP3, PKA
Erk _||_ Jnk | PKA, PKC
Erk _||_ Jnk | Mek, PKA
Erk _||_ P38 | PKA, PKC
Erk _||_ P38 | Mek, PKA
Erk _||_ PIP2 | PKC
Erk _||_ PIP2 | Mek, PKA
Erk _||_ PIP3 | PIP2, Plcg
Erk _||_ PIP3 | PKC
Erk _||_ PKC | Mek, PKA
Erk _||_ Plcg | PKC
Erk _||_ Raf | Mek, PKA
Jnk _||_ Mek | PKA, PKC
Jnk _||_ P38 | PKA, PKC
Jnk _||_ PIP3 | PIP2, Plcg
Jnk _||_ PIP3 | PKC
Jnk _||_ Plcg | PKC
Jnk _||_ Raf | PKA, PKC
PIP2 _||_ PKA | PKC
PIP2 _||_ Raf | PKC
PIP3 _||_ PKA | PKC
PIP3 _||_ Raf | PKC
PIP3 _||_ Plcg | PKC
PKA _||_ PKC
PKA _||_ Plcg
Plcg _||_ Raf | PKC

```

```

> localTests(pathway_net, data = dades, type = "cis")
#> #> estimate          p.value      2.5%      97.5%
#> Akt _||_ Erk | Mek, PKA    0.8970851604  0.000000e+00  0.963463790  0.9750846669
#> Akt _||_ Erk | PKA, PKC    0.8976677512  0.000000e+00  0.964366850  0.9760253392
#> Akt _||_ Erk | PIP2, PKA, Plcg 0.8969931314  0.000000e+00  0.96310055  0.9749365408
#> Akt _||_ Erk | PIP3, PKA    0.8972970312  0.000000e+00  0.964875186  0.9764490934
#> Akt _||_ Jnk | PKA, PKC    -0.0571097695  6.609962e-13  -0.075175525  -0.039067031
#> Akt _||_ Jnk | PIP2, PKA, Plcg -0.0571097695  6.609962e-13  -0.075175525  -0.039067031
#> Akt _||_ Jnk | PIP3, PKA    -0.0624895340  1.393854e-11  -0.080541934  -0.0443963911
#> Akt _||_ Mek | PKA, PKC    0.0648072773  2.383684e-12  0.046719287  0.0828541553
#> Akt _||_ Mek | PIP2, PKA, Plcg 0.0603038930  6.970457e-11  0.042205811  0.0783626161
#> Akt _||_ Mek | PIP3, PKA    0.0617998831  2.328828e-11  0.043705410  0.0798540533
#> Akt _||_ P38 | PKA, PKC    0.2137041498  1.508917e-121 0.1963411569  0.2309967871
#> Akt _||_ P38 | PIP2, PKA, Plcg 0.1612337255  1.965115e-69  0.143919205  0.1792682785
#> Akt _||_ P38 | PIP3, PKA    0.1655014964  9.050500e-73  0.147805595  0.1831090696
#> Akt _||_ PIP2 | PIP3     0.0136674536  1.407006e-01  -0.031772228  0.00491063921
#> Akt _||_ PKC | PIP2, Plcg  -0.0034278992  7.111872e-01  -0.021570048  0.0147165062
#> Akt _||_ PKC | PIP3     -0.0039815169  6.671376e-01  -0.022122632  0.0141622189
#> Akt _||_ Plcg | PIP3     -0.0638592012  4.937006e-12  -0.081907264  -0.0457695295
#> Akt _||_ Raf | PIP2, PKA, Plcg 0.0191680005  3.351180e-02  -0.037810778  -0.0013362876
#> Akt _||_ Raf | PIP3, PKA    0.0233162749  1.177094e-02  -0.041442244  -0.0051749741
#> Erk _||_ Jnk | PKA, PKC    0.0190370631  3.973284e-02  -0.037167720  -0.0008938819
#> Erk _||_ Jnk | Mek, PKA    0.0236080825  1.078186e-02  -0.041726456  -0.0054596728
#> Erk _||_ P38 | PKA, PKC    0.2352229982  8.164557e-148 0.218043306  0.2523539457
#> Erk _||_ P38 | Mek, PKA    0.1873918261  2.286637e-93  0.169962079  0.2049884262
#> Erk _||_ PIP2 | PKC     -0.0106795997  2.486485e-01  -0.028816730  0.0746445593
#> Erk _||_ PIP2 | Mek, PKA  -0.0173498184  6.090157e-02  -0.035482143  0.0007939216
#> Erk _||_ PIP3 | PIP2, Plcg 0.0430115261  3.340170e-06  0.024887458  0.0611073861
#> Erk _||_ PIP3 | PKC     0.0310118228  8.054193e-04  -0.02876355  0.0491269139
#> Erk _||_ PIP3 | Mek, PKA  0.0007349409  9.367273e-01  -0.01408781  0.0188781794
#> Erk _||_ PKC | Mek, PKA  -0.0212918748  1.243877e-02  -0.039420153  -0.0031496114
#> Erk _||_ Plcg | PKC     -0.0655465654  1.338120e-12  -0.083588237  -0.0747593949
#> Erk _||_ Plcg | Mek, PKA -0.0691058037  7.644257e-14  -0.087140372  -0.0510263145
#> Erk _||_ Raf | Mek, PKA  -0.1036104943  2.880355e-29  -0.12151879  -0.0856197056
#> Jnk _||_ Mek | PKA, PKC  0.0087265133  3.458812e-01  -0.099418668  0.0268657507
#> Jnk _||_ P38 | PKA, PKC  0.0321854183  5.057859e-04  -0.04501033  0.0502995571
#> Jnk _||_ PIP2 | PKC     0.0131965551  1.539968e-01  -0.004947342  0.0313317689
#> Jnk _||_ PIP3 | PIP2, Plcg 0.0415695651  2.999791e-11  0.043361827  0.0795119993
#> Jnk _||_ PIP3 | PKC     0.0487249416  1.392775e-07  0.030598484  0.0667979167
#> Jnk _||_ Plcg | PKC     -0.0426640538  4.001450e-06  -0.0675579  -0.0245043370
#> Jnk _||_ Raf | PKA, PKC  -0.0679550460  1.9113157e-13  -0.086023242  -0.0499036325
#> Mek _||_ P38 | PKA, PKC  0.1085003780  5.852406e-32  0.093535520  0.1263972121
#> Mek _||_ PIP2 | PKC     0.0431071653  3.173489e-06  0.024983993  0.0612020704
#> Mek _||_ PIP3 | PIP2, Plcg 0.0122087029  1.872444e-01  -0.030435472  0.0059361025
#> Mek _||_ PIP3 | PKC     0.0099681582  2.815780e-01  -0.028105578  0.0081760331
#> Mek _||_ Plcg | PKC     -0.0365465094  7.832827e-05  -0.045633012  -0.0184610031
#> P38 _||_ PIP2 | PKC     0.0370824934  6.138214e-05  -0.051188114  -0.0189525317
#> P38 _||_ PIP3 | PIP2, Plcg 0.0356793835  1.154907e-04  0.017457253  0.0537880840
#> P38 _||_ PIP3 | PKC     -0.0015846976  8.640847e-01  -0.09725843  0.0165584913
#> P38 _||_ Plcg | PKC     -0.0601620129  7.693778e-11  -0.07821954  -0.0420652232
#> P38 _||_ Raf | PKA, PKC  -0.07222846553  5.224399e-15  -0.09031016  -0.0542122957
#> PIP2 _||_ PKA | PKC     0.0109783862  2.356352e-01  -0.007164976  0.0291145243
#> PIP2 _||_ Raf | PKC     0.0470177755  3.725312e-07  0.028899236  0.065310453108
#> PIP3 _||_ PKA | PKC     0.1242992653  1.628804e-41  0.106398171  0.1423246064
#> PIP3 _||_ Raf | PIP2, Plcg 0.0491059982  1.105999e-07  0.030990002  0.0673895980
#> PIP3 _||_ Raf | PIP3, PKA 0.0798126545  1.276274e-03  0.011676257  0.0479294598
#> PIP3 _||_ Raf | PIP2, Plcg 0.0147883923  1.191614e-01  -0.003355998  0.0329230513
#> PKA _||_ PIP2 | PKC     0.1506394654  1.997898e-60  0.132863400  0.1683304918
#> PKA _||_ Plcg | PKC     0.0147671144  1.106438e-01  -0.003375275  0.0329002384
#> Plcg _||_ Raf | PKC     0.0595229433  2.127059e-10  0.041424974  0.0775802625
> |

```

## 5.6.2 SUBGRAF AILLAT: CADENA VERTICAL DE N=3 NODES AMB TEST NEGATIU del model geomètric topoempíric: primera validació introductòria a escala local a partir del tipus causal cadena

Hom rebutja que Pip3 sigui independent de PKC fins i tot condicionant no sols per Plsg sinó també per PIP2

Hom agafa com a mostra una cadena vertical del DAG 2D, dos nodes non\_adjacents; després s'hi aplica el test d'independència condicionada per mitjà de la funció de dagitty d'R localTest() i hom compara el resultat del test amb el comportament del model geomètric.

Condicionant sols per Plcg el test surt negatiu  
cadena3n\_vertical <- dagitty(dag {  
bb="-0.5,-0.5,0.5,0.5"  
Plcg|pos="12.801835,8.419918510, 14.423490868"]  
PIP3|pos="15.140769,0.000000000, 0.000000000"]  
PKC|pos="19.628501,14.467310719, 4.676170477"]  
PIP3 -> Plcg  
Plcg -> PKC  
}  
')

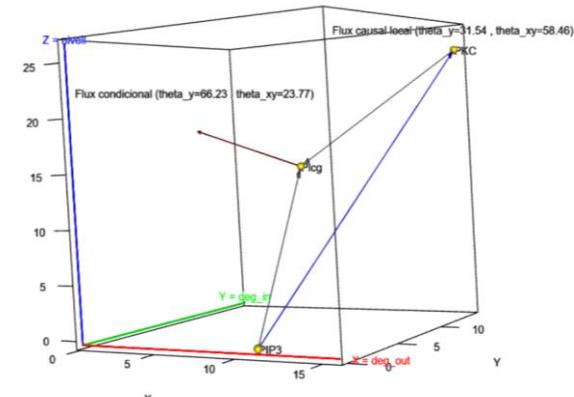
```
(pathway_ind_con <- localTests(x = cadena3n_vertical, data = dades[,c(3,5,9)], type = "cis"))
```

estimate	p.value	2.5%	97.5%
PIP3 _  _ PKC   Plcg 0.04603344	6.491795e-07	0.02791405	0.06412268

1 row

Condicionant pels dos medials el test surt negatiu

```
PIP3 _||_ PKC | PIP2, Plcg 0.049105879 1.105999e-07 0.030990002 0.067189598
```



Pel que fa al test, aquest resol en contra de la independència entre nodes en aquesta cadena si hom condiciona pel medial quan teòricament n'haurien d'esser, d'independents, perquè el flux que els uneix és tallat (de fet el test encara és més categòric perquè condiciona per tots dos nodes medials entre PIP3

i PKC, tant per Plcg com per PIP2). Encara que hom condiciona pels medials, no hi ha independència, situació que es tradueix geomètricament en dos vectors directors (amb o sense condicionament) i els seus respectius angles respecte l'eix vertical Z.

Sense condicionament el sentit de les arestes segueixen dibuixant una cadena, el vector director de la qual té l'origen a PIP3 i la terminació en PKC amb un angle respecte el pla YX de 58.46. Però amb condicionament, el sentit de les arestes canvien, ara totes les arestes van cap al node medial Plcg i dibuixen un col·lididor com a origen del flux, el cap terminal avança més o menys cap a X=0 i Y=15 en direcció Y ascendent amb un angle respecte el pla YX de 23.77 graus.

### 5.6.3 SUBGRAF AÏLLAT: CADENA LATERAL DE N=3 NODES AMB TEST NEGATIU del model geomètric topoempíric

Hom rebutja que Erk sigui independent de Raf fins i tot condicionant no sols per Mek sinó també per PKA

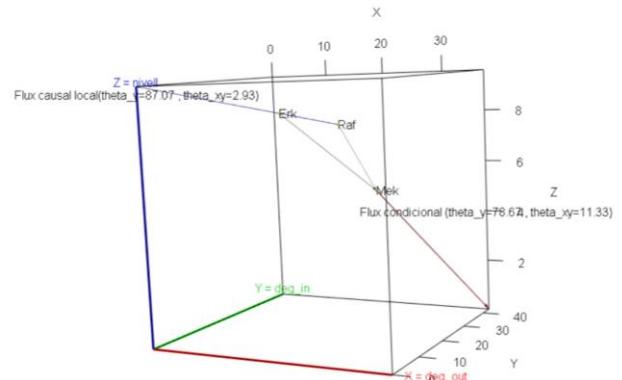
Hom agafa com a mostra una cadena vertical del DAG 2D, dos nodes non\_adjacents; després s'hi aplica el test d'independència condicionada per mitjà de la funció de dagitty d'R localTest() i hom compara el resultat del test amb el comportament del model geomètric.

```
cadena3n_lateral1 <- dagitty('dag {
bb="-0.5,-0.5,0.5,0.5"
Raf[pos="8.017768,16.705511579 , 14.051138412"]
Mek[pos="7.997961,29.315418380 , 11.497698618"]
Erk[pos="0.000000,15.550560292 , 14.465575026"]
Raf -> Mek
Mek -> Erk
}
')
(pathway_ind_con      <-      localTests(x      =
cadena3n_lateral1, data = dades[,c(1,2,6)], type
= "cis"))
```

	estimate	p.value	2.5%	97.5%
Erk   _ Raf   Mek	-0.08165858	9.544188e-19	-0.09965431	-0.06361018

Condicionant pels dos medials el test també surt negatiu

	estimate	p.value	2.5%	97.5%
Erk   _ Raf   Mek, PKA	-0.1036019493	2.880355e-29	-0.121518799	-0.085619



Pel que fa al test, aquest resol en contra de la independència entre nodes en aquesta cadena si hom condiciona pel medial quan teòricament n'haurien d'esser, d'independents, perquè el flux que els uneix és tallat (de fet el test encara és més categòric perquè condiciona per tots dos nodes medials entre Raf i Erk, tant per Mek com per PKA). Encara que hom condiciona pels medials, no hi ha independència, situació que es tradueix geomètricament en dos vectors directors (amb o sense condicionament) i els seus respectius angles respecte l'eix vertical Z.

Sense condicionament el sentit de les arestes segueixen dibuixant una cadena, el vector director de la qual té l'origen a Raf i el cap terminal en Erk amb un angle respecte el pla YX de 2.93. Però amb condicionament, el sentit de les arestes canvien, ara totes les arestes van cap al node medial Mek i dibuixen un col·lididor com a origen del flux, el cap terminal avança més o menys cap a X=40 i Y=40 en direcció Y descendent amb un angle respecte al pla YX de 11.33 graus.

### 5.6.3 BATERIA DE SUBGRAFS AÏLLATS de 3 NODES AMB TEST NEGATIU del model geomètric topoempíric

PIP3   _ PKC   Plcg	theta_z 31.54352 versus theta_z 66.22501 theta_xy 58.45648 versus theta_xy 23.77499	1	1
---------------------	--	---	---

Erk   _ Raf   Mek	theta_z 87.0712 versus theta_z 78.66945 theta_xy 2.928796 versus theta_xy 23.77499	1	-1
-------------------	---	---	----

### 5.6.4 BATERIA DE SUBGRAFS AÏLLATS de 3 NODES AMB TEST POSITIU del model geomètric

topoempíric			
P38 _  _ PIP3   PKC plantejable com a cadena PIP3->...->PKC->P38	theta_z 43.9617 versus theta_z 32.50765 theta_xy 46.0383 versus theta_xy 57.49235	-1	1
Jnk _  _ PIP2   PKC plantejable com a cadena PIP2->PKC->Jnk	theta_z 60.1084 versus theta_z 52.37682 theta_xy 29.8916 versus theta_xy 32.13276	-1	1
Mek _  _ PIP3   PKC plantejable com a cadena PIP3->...->PKC->Mek	theta_z 55.05735 versus theta_z 23.06977 theta_xy 34.94265 versus theta_xy 66.93023	-1	1
NO PLANTEJABLES COM A CADENA PERQUÈ NO TÉ UNA DE LES ARESTES			
Erk _  _ PIP2   PKC plantejable com a cadena PIP2-> PKC [->] Erk	theta_z 51.85653 versus theta_z 57.86724 theta_xy 38.14347 versus theta_xy 37.62318	1	1
Akt _  _ PKC   PIP3 plantejable cadena PIP3 ->...->PKC [->] Akt	theta_z 56.48893 versus theta_z 36.42483 theta_xy 33.51107 versus theta_xy 53.57517	1	1
Akt _  _ PIP2   PIP3 forca			

### CONCLUSIÓ TAULA

Si no hi ha errors de càlcul ni de plantejament, **no és correcte comparar l'angle del flux local del vector director del col·lididor (amb condicionament) versus el flux local del vector director de la cadena (sense condicionament) com a aproximació del flux causal general;** CAL FER LA COMPARANÇA AMB EL FLUX CAUSAL GENERAL o bé respecte l'eix vertical Z com a aproximació o bé respecte dels tres eixos com a mètode robust.

Les diferències d'angles respecte de Z no estableixen un llindar per a destriar un test positiu d'un altre de negatiu.

**Un altra cosa fóra convertir aquests subgrafs com a grafs, és a dir, com a exemples trivials, un model geomètric basat sols en 3 nodes amb un joc de dades de sols 3 columnes. Entendre si l'heurística geomètrica aquí definida és consistent amb la semàntica causal en condicions pures.**

### 5.7 ASSAIG DE VALIDACIÓ LOCAL (l'aresta absent entre Akt i Erk segons reporten SACHS et al.) A PARTIR DEL FLUX CAUSAL GENERAL

Aquí cal tenir en compte una distinció essencial: el càlcul del flux causal geomètric general té dos criteris: O bé hom agafa

Aquestes tres dimensions suposen un altre salt en la construcció del model causal geomètric, perquè ara hom formalitza un flux causal òptim en què no sols treballa l'eix Z sinó tots tres eixos de manera que el flux causal sortint sigui òptim (màxima verticalitat i màxim graus\_IN i mínim graus\_OUT), d'aquí que si hom opta per aquesta definició robusta del flux causal, cal que parteixi dels nodes fulla del subgraf retrolateral. Ara hi ha una única direcció a dins de l'esfera de direccions, cosa que és un salt qualitatius respecte del flux causal basat sols en la verticalitat.

--> Hi han dues vies de flux causal encapçalades pels dos nodes arrels dels dos arbres del bosc: PIP3 i PKA

--> PIP3 no té cap camí de flux cap a una fulla pròpria final (PKC acaba essent un node medial), sols un node col·lididor-fulla vertical amb PKA

--> PKA té un únic camí de flux cap a una fulla pròpria final però retrolateralitzada per la cadena Raf->Mek->Erk, la resta de nodes finals són col·lididors retroverticals (cosa que debilita en gran mesura el flux tant de PKA com de PIP3 però els seus fluxos de sentit contrari xoquen als nodes col·lididors).

Concloent, els dos nodes terminals de flux verticals finals que semblen d'entrada operatius són un col·lididor-fulla (Akt) de PIP3 i PKA i un node-fulla (Erk) però retrolateritzat de PKA. Hom ja veu que el node PKC és el taló d'Aquil·les del model justament perquè és el principal obstacle del flux causal

vertical.

LA QÜESTIÓ CLAU és si el node fulla vertical retrolateritzat Erk és connectat a l'únic node col·lididor de flux vertical. L'efecte dels nodes arrels convergeix o no a Akt? Caldrà veure si l'angle de flux general respecte de les cadenes Raf->Mek->Erk lateral o Raf->Mek->Erk vertical convergeixen.

### Flux local local lateral versus FLUX CAUSAL VERTICAL

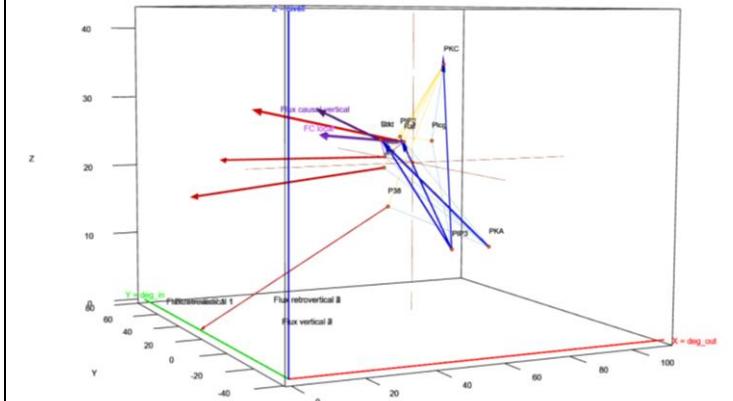
→4 vectors directors verticals blaus  
→4 vectors directors retroverticals que serveixen per corregir el flux global vertical

#### AQUÍ HOM SE SERVEIX DEL SUBGRAF VERTICAL

\*amb lila tant el local com el vertical, hom pot comparar tots dos en sentit ascendent (caldria situar l'origen del vector director del vertical a la coordenada mitjana dels nodes arrels)

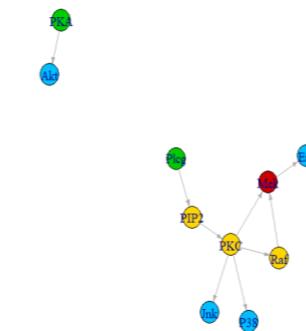
Hom no pot fer servir el con causal per què el flux causal és definit sols per l'eix vertical.

El flux causal vertical és basat en el subgraf bosc vertical (tot tenint en compte els retrofluxos a fi de corregir el vertical)



El flux causal holístics és basat en el subgraf retrolateral (en blau hom pot veure just els 4 col·lididors fulla que s'empren per a calcular el vector director del flux causal holístic)

subgraf retrolateral non-connex dirigit



## 2.3 Reflexió general abans de tancar els objectius prevists

L'estructura del model causal geomètric és fixada en les fases aquí dessota detalles, però no pas el com: hom no imposa per a la fase de descomposició del DAG 3D un graf boscoïde, hom no imposa per a la fase de descomposició del subgraf retrolateral un tipus causals ni el seu nombre mínim de nodes, no imposa per a la fase de transformacions geomètriques un estadístic concret o la correlació entre variables, no imposa a dins del flux general causal la mitjana vetorial global. Totes són propostes, el mètode és obert i flexible, més encara quan el mètode és en procés de construcció gràcies a l'assaig-error amb estudis que són <golden standard> o eines estadístiques com és el localTest() per al DAG 2D.

→ **A la fase de descomposició del DAG 2D**, hom tria el graf arbòri o boscoïde de la teoria de grafs perquè és raonable pensar que és el més vertical i directe en termes de flux causal entre uns nodes arrels i uns nodes fulles finals; tanmateix hom pot fer la descomposició segons un altre graf, per exemple, a causa de la natura de les dades empíriques.

→ **A la fase de descomposició del subgraf retrolateral (fase nova respecte de la PAC2)**, hom tria els tres tipus essencials de la inferència causal (cadena, forca i col·lididor) en la seva expressió comuna mínima que són tres nodes, però hom pot ampliar tant els tipus essencials a partir de la seva composició (per exemple una doble forca unida pels fills) com el nombre de nodes mínims.

→ **A dins de la fase de la cadena de transformacions geomètriques**, hom tria un estadístic senzill com és la mitjana de cadascuna de les 11 columnes variables dividides per la desviació tipus global; un estadístic senzill que és fet amb les dues mesures més essencials de l'estadística, un mesura central i una altra de desviació, a fi d'assajar una transformació geomètrica associativa entre les dades topològiques (les 3 coordenades que emergeixen de la descomposició: verticalitat, graus\_in, graus\_out) i les empíriques; però és lògic pensar que si el model és viable, hom pot construir estadístics més robusts i operatius per a tal fi. El mateix raonament per a la segona transformació geomètrica basada en la correlació estadística entre columnes associada a la llargària entre els nodes existents.

El nombre de transformacions geomètriques per tal que les dades empíriques afaiçoni el model geomètric topològic no és fixat, aquí hom en construeix dues, de transformacions, però no té cap topall i no tenen perquè esser en exclusiva de natura estadística, de fet, en bioestadística estructural moltes de les eines són de natura física.

→ **A dins del flux general causal**, hom pot escollir una mitjana vectorial global (un mètode més robust que no pas fer una bisectriu sistemàtica) entre els vectors directors verticals respecte sols de l'eix Z, però (novetat respecte de la PAC2) l'òptim és entre els vectors directors dels nodes fulla del subgraf retrolateral perquè la mitjana global ara ja és respecte de tots tres eixos Z-Y-X. Per tant, aquí l'eina per explorar geomètricament el mode és a través de la mitjana vectorial global però

hom pot explorar d'altres eines com fóra una regressió lineal per tal de determinar el flux general causal.

## 2.4 Justificació dels canvis

Fins a la PAC2 hom fa sols la mitjana vectorial global de tots els vectors directors provinents tant dels nodes fulla pertanyents del subgraf vertical (cadenes de 3 o 2 nodes SENSE retroflux al model geomètric 3D) com dels nodes fulles pertanyents del subgraf retrolateral (col·lididors AMB retroflux al model geomètric 3D). Així, hom calcula l'angle respecte de z de la mitjana dels vectors directors; però ara hom fa la mitjana vectorial global dels vectors directors directors provinents de tots els nodes fulla del subgraf retrolateral (siguin col·lididors verticals, retroverticals o lateroverticals al model geomètric 3D)

Hom passa de la definició de flux de causalitat que hom lliga sols amb l'eix vertical Z, que és senzilla però amb biaix perquè no és l'òptima, aquí sols treballa l'eix Z vertical. Aquesta no deixa d'esser una bona via exploratòria o simplificada, útil en cas d'ambigüïtat. Però de fet n'hi ha una segona, de manera, més holística i robusta: el flux causal més potent i robust és tenir en compte alhora:

1. Z: Una vertical MÀXIMA
2. Y: uns graus\_IN MÀXIMA
3. X: uns graus\_OUT MÍNIMA

De fet, els nodes qui compleixen amb exactitud aquests requisits d'optimització són els nodes fulla del subgraf retrolateral (que justament són els col·lididors-fulla del model causal geomètric 3D), per tant, el flux òptim en totes tres dimensions és el resultat de calcular la mitjana vectorial global entre tots els vectors directors dels col·lididors-fulla. És la mitjana de les orientacions dels vectors directors a dins de l'esfera de direccions que té com a origen la coordenada mitjana dels nodes arrels.

Ara, la mitjana vectorial global respecte dels tres eixos (Z, Y i X) hom pot llegir el problema causal com a optimització geomètrica, perquè optimitzant els tres eixos hom pot llistar les vies causal prioritàries; fins i tot, hom pot detectar arestes absents al model de treball: si tots dos nodes tenen 0 outs, el mateixos IN però una diferència d'alçada, és probable que hi hagi un node absent entre ells; també la situació en què tots dos tinguin 0 outs, el mateixos IN, sense diferència d'alçada però un IN col·lididor vertical i l'altre un IN col·lididor retrovertical, aquesta asimetria juga a favor d'una aresta absent del col·lididor retrovertical cap al vertical. Però treballant amb els tres eixos amb perquè hom obté un flux causal del model general únic en l'esfera de direccions, perquè és l'únic vector resultant de maximitzar Z i Y tot minimitzant alhora X.

## 3. Relació de les activitats realitzades

### 3.1 Activitats previstes en el pla de treball: intent de validació geomètrica d'una aresta absent entre Akt i Erk per mitjà d'una regió conoïde de convergència entre el flux causal holístic i el flux causal local

### 3.2 Activitats no previstes i realitzades o programes:

- ⇒ **Flux causal vertical corregit (mitjana vectorial vertical global respecte de l'eix vertical Z) ja explicat a la PAC2**

- ⇒ ARA hi ha un segon procediment més reboust: el flux causal vertical corregit (mitjana vectorial vertical global projectat respecte dels 3 eixos), implica el disseny d'un con casual al voltant d'aquest vector de flux causal holístic respecte dels tres eixos.

Ara hi han l'angle de Z i l'angle del pla YX, amb els quals es determina el punt exacte en l'esfera de direccions tot tenint en compte que hom estableix com a origen d'aquest flux causal holístic la coordenada mitjana dels nodes arrels del subgraf vertical.

Hom vol crear una regió en l'espai de direccions causals en forma de con: havent obtingut el vector director de flux causal (expressat en termes d'angles respecte de Z (theta) i respecte del pla YX (phi)) i un origen com a coordenada mitjana dels nodes arrel, hom circumscriu aquest flux amb un con segons la variació angular mitjana dels 4 col·lididors perquè hom pugui determinar quins fluxos causals locals hi convergeixen o no. És una forma del tot geomètrica i visual de determinar la convergència causal d'una via local al flux holístic.

- ⇒ **Pel que fa a la fase de descomposició**, hom s'adona que el subgraf retrolateral no ha estat gaire analitzat, ha servit per copsar les dues dimensionalitats de graus\_in i graus\_out, però pot tenir més utilitat pel que fa al model causal geomètric. D'aquí que cal descompondre el subgraf retrolateral als tres tipus causals essencials de tres nodes (cadena, col·lididor i forca).
- ⇒ **Interval de flux (IF)**  
Sorgeix d'agafar la distància angular mitjana dels 4 col·lididors fulla del subgraf retrolateral, és a dir, la variabilitat dels vectors col·lididors fulla
- ⇒ **Con causal**  
Amb els dos angles del flux causal holístic (és a dir amb el vector director final) hom ja pot construir el con causal. L'amplada del con depèn de la variabilitat real dels vectors directors dels col·lididors fulla, és a dir, de la seva dispersió angular.
- ⇒ **Proximitat entre nodes al model espacial**  
Com més a prop són dos nodes a l'espai causal més probable que hi hagi una aresta no detectada fins al punt que hom pugui agrupar-les com a una única variable atès que en el cas degenerat les dues variables se sobreposarien del tot.
- ⇒ **Proximitat dels nodes amb el centre de masses al model espacial**  
També és significativa la proximitat dels nodes amb el centre de masses, hom podria intuir que a mesura que s'hi acosten més possibiliats de connectivitat entre els qui hi són a prop i no tenen aresta entre ells
- ⇒ **Mitjana vectorial global:**  
La bisectriu simple deixa d'esser representativa i cal obtenir una direcció mitjana que reflecteixi la resultant geomètrica real del camp causal 3D. La versió robusta de la bisectriu sistemàtica és la mitjana vectorial global, quan hi ha més de dues arestes que defineixen el flux (diverses direccions locals verticals, quan hi ha 3 o més direccions locals o angles dispersos) i els angles entre aquests poden anar més enllà dels 45°, altrament dit, és invariant a l'ordre de combinació dels vectors i geomètricament equivalent a una direcció sortint mitjana (reflecteix la direcció resultant real del camp causal 3D).

És la direcció de màxim consens funcional entre totes les arestes verticals del subgraf i, així, obtenir l'angle òptim de flux causal vertical per a un model geomètric concret. Formalment, correspon a la suma vectorial ponderada dels vectors directors  $\vec{v}_i$  de cada aresta, on  $w_i = |\vec{v}_i|$  és el mòdul de cada vector (la seva força en termes de flux causal local) que actua com a pes.

Aquesta ponderació assegura que les arestes amb flux causal més intens tinguin més de pes en la definició de la direcció resultant. Geomètricament, és la mitjana vectorial global és una direcció sortint mitjana.

$$\vec{V}_{global} = \frac{\sum_i w_i \vec{v}_i}{\|\sum_i w_i \vec{v}_i\|}$$

- ⇒ **Test d'independència condicionada: reversió del sentit de l'aresta\_IN o aresta\_OUT (a fi de comparar els dos vectors directors locals <amb> condicionament respecte del flux causal holòstic o global)**

Si hom treballa amb **geometria algebraica lineal** les definicions de condicionament sobre un node tenen restriccions perquè, si bé obereixen a simetries que responen a relacions purament algebraiques ( $AB=B-A$ ,  $BA=-(B-A)$ ,  $|AB|=|BA|$ ,  $\text{angle}(AB,Z)=-\text{angle}(BA,Z)\dots$ ), no responen a relacions causals ( $BA$  no és  $-AB$ , el pendent vertical és causal no pas geomètric, hi han contrafluxos...els vectors directors no són lineals algebraics sinó físics perquè tenen intensitat, verticalitat causal...); llavors això podria esser un problema perquè limita les possibilitats d'expressar geomètricament les susdites relacions causals. La geometria algebraica lineal és euclidiana, la geometria algebraica causal és non-euclidiana (el mateix eix privilegiat de flux causal vertical Z ja trenca Euclides)

Per exemple: si hom condiciona pel node medial d'una forca ( $A <- B -> C$ ) *sota la definició que totes les arestes\_OUT esdeven arestes\_IN* ( $A -> B <- C$ ) i aplica una pura geometria algebraica, llavors tant *<sense>* condicionament ( $A <- B -> C$ ) com *<amb>* condicionament ( $A -> B <- C$ ) l'angle respecte l'eix vertical Z és el mateix (amb canvi de signe) i el mòdul és exactament el mateix. Cosa que en termes causals o físics no és correcte perquè són fenòmens opositius (un és el xoc de fluxos, es frenen, l'energia vectorial es dissipa, per tant el mòdul ha d'escurçar-se); en canvi, a una forca el flux es divideix però no topa, el mòdul ha d'ésser més llarg que el d'una topada de forces). D'aquí que calgui recórrer a la cadena per poder comparar la forca *<sense>* condicionament ( $A <- B -> C$ ) i la mateixa *<amb>* condicionament ( $A -> B -> C$ ), però pagant-ne un preu, la independència dels nodes extrems és parcial, sols un n'és respecte de l'altre, el node A respecte del C o a l'inrevés (hi torna haver simetria) però no tots dos alhora. El mateix problema hi ha amb la figura causal del col·lididor, hom s'és a la situació inversa; cal recórrer una altra volta a la solució parcial de la cadena. Una segona solució és fer la comparança *<amb>* condicionament versus el flux causal vertical general corregit.

Ara bé, si hom treballa amb **geometria física o causal**, no hi ha cap restricció, sigui quina sigui la definició que hom fa de condicionament en cap dels tres tipus causals essencials: cadena, forca i col·lididor. Ara hom pot definir geomètricament el col·lididor com a invers causal de la seva forca i la forca com a invers causal del seu col·lididor (ara hom pot invertir el sentit de les arestes de tots dos vectors), hom no cau en l'artifici algebraic (la simetria) que no descriu el seu significat causal; igual com en física: la forca/divergència allarga el mòdul (té un guany de nivell energètic) i el col·lididor/convergència escurça el mòdul (pèrdua de força) per saturació.

Per exemple, de la forca  $A <- B -> C$ ,  $B -> A$  és molt vertical i  $B -> C$  és molt horitzontal, les dues no poden ésser perfectament opositives perquè són arrelades a la geometria espacial dels nodes: → la direcció sortint no és simètrica (el vectors directors no tenen la mateixa inclinació)

→ de la mateixa manera, quan hom inverteix el sentit d'una aresta, el mòdul/intensitat del flux tampoc no és la mateixa (no hi pot haver simetria inversa quan en canvia el sentit, perquè ara A->B suposa anar en contra la verticalitat i B->C té un pendent diferent; és a dir, la inversió del flux no inverteix la geometria del camí).

Cal afegir tres notes addicionals: hom preveu que la millor manera de desenvolupar el model causal geomètric és assumir la geometria física perquè hom suposa que aquesta dóna millors resultats respecte als resultats estadístics com a punt de referència de validació. I una segona observació, les dades biomèdiques són treballades per mitjà de la geometria física, no pas amb àlgebra lineal. I en darrer lloc, dir que els vectors directors són definits de manera diferent segons sigui geometria algebraica lineal o física. En resum, no sembla que la simetria algebraica existeixi en la geometria real en l'àmbit biomèdic on hom mesura les dades d'éssers vivents.

En condicionar per un node ---resultat d'aplicar un test local geomètric d'independència entre dos nodes non-adjacents--- en un model geomètric 3D vol dir teòricament (les dades poden contradir el model si detecten camins espuris):

#### → EN GEOMETRIA ALGEBRAICA LINEAL: geometria euclidiana

→ *per un cadena (vector director cadena versus vector director col·lididor)*: que una de les arestes\_OUT reverteixi el seu sentit cap al mateix node condicionat; o just a l'inrevés, que una de les arestes\_IN reverteixi el seu sentit cap al node adjacent (la cadena esdevé o bé un col·lididor o bé una forca).

→ *per una forca (vector director forca versus vector director cadena)*: que una de les dues arestes\_OUT reverteixi el seu sentit cap al mateix node condicionat, és a dir, que el node condicionat sols tingui una aresta\_IN i la resta romanguin arestes\_OUT (la forca esdevé cadena). Hom no pot revertir totes les arestes\_OUT a arestes\_IN perquè són geomètricament simètriques; a més, hom vol veure si hi ha un camí espuri entre els nodes fills, cosa que implica un vector director en configuració de cadena, no pas de forca.

→ *per un col·lididor (vector director col·lididor versus vector director cadena)*: que una de les dues arestes\_IN sotmeses a test d'independència condicionada reverteixi el seu sentit cap al mateix node condicionat, és a dir, que el node condicionat sols tingui una aresta\_OUT i la resta romanguin com a arestes\_IN (el col·lididor esdevé també cadena). El mateix fenomen de simetria susdit al cas d'una forca.

#### → EN GEOMETRIA FÍSICA O CAUSAL: geometria vectorial non-euclidiana a dins d'un espai causalment estratificat.

→ *per un cadena (vector director cadena versus vector director col·lididor)*: que una de les arestes\_OUT reverteixi el seu sentit cap al mateix node condicionat (també és plantejable just a l'inrevés, que una de les arestes\_IN reverteixi el seu sentit cap al node adjacent: aquí la cadena esdevé una forca).

→ *per una forca (vector director forca versus vector director col·lididor)*: que les dues arestes\_OUT sotmeses a test d'independència condicionada reverteixin el seu sentit cap al mateix node condicionat.

→ *per un col·lididor (vector director col·lididor versus vector director forca)*: que les dues arestes\_IN sotmeses a test d'independència condicionada reverteixin el seu sentit cap al node condicionat.

- ⇒ **El model toroïdal accepta retrofluxos locals sense comprometre l'aciclicitat total del model**

# Table of Contents

## TESI DE MÀSTER EN BIOESTADÍSTICA I BIOINFORMÀTICA UOC-UB

## INFERÈNCIA CAUSAL GEOMÈTRICA: causalitat com a estructura geomètrica de l'espai-temps

*Mètode de conversió de pathways a un model causal geomètric 3D i validació empírica multiescala del graf resultant*

### 1. Taula de continguts

### 2. Abstract o resum

In this work we introduce a Geometric Causal Model (GCM), a novel extension af standard DAGs thta embeds causal structure into a three-dimensional coordinate system in which axes represent causal dependency, causal effect, and hierarchical causal depth. This geometric representation allows for consistency checks, detection of spurious edges, and a topogeometric validation of causal flow.

### 3. Objectius

3.1 Descripció metodològica de conversió de pathways a un model causal geomètric

3.2 Validació causal geomètrica local estadística: tests d'independència condicionada

3.2 Validació multiescala per machine learning o aprenentatge estadístic  
*Random Forests (variables), SVM (validació local) i ANN/CNN (validació global)*

5.b.1. Validació de rellevància de les variables (nodes) mitjançant Random Forests, per avaluar la influència dels nodes sobre el model causal. 5.b.2. Validació local de les relacions locals d'arestes horizontals amb SVM; cal avaluar una a una i en combinació quines arestes milloren o empitjoren la capacitat explicativa i/o predictiva del model.

5.b.3. Validació global del DAG complet mitjançant ANN/CNN per avaluar la millora o deteriorament

en predicció (també pot explorar-se l'efecte d'una cross edge en termes globals amb la seva inclusió i exclusió). 5.b.4. Integració dels resultats obtinguts a les tres escales (variable, local, global), i extracció de conclusions sobre la validesa i aplicabilitat del mètode.

## 4. Mètode

Per mitjà d'un codi en R, primer hom trasllada un pathway biomèdic com a graf acíclic dirigit 2D (DAG); en segon lloc, en el marc de la teoria de grafs, tot seguit, hom descompon el DAG 2D en dos subgrafts com a subgrafts (vertical o de jerarquia i retrolaterals o de cross edges). Aquesta descomposició permet una anàlisi de les tres dimensions que hi ha a l'estructura subjacent del DAG i fer-ne la conversió a un model causal geomètric.

La descomposició dels subgrafts és un primer estadi, permeten copsar el DAG 2D de forma òptima, per exemple en termes d'estructura, quina mena de nodes uneixen els dos subgrafts (en aquest estudi són col·lididors-fulla, un únic vertical i la resta retroverticals); ara bé, a fi d'aplicar-hi inferència causal, cal arribar fins al model causal geomètric topoempíric.

Tot seguit validar experimentalment el model causal geomètric mitjançant tests locals i la tríada d'aprenentatge estadístic.

## 5. Anàlisi crítica dels resultats

- 5.1
- 5.2 5.3
- 5.4
- 5.5

També l'anàlisi crítica dels resultats i comparança amb els resultats de Sachs et al. (2005),

## 6. Discussió sobre l'aplicabilitat i l'abast del nou mètode

- 6.1

cal tenir en compte que m'he trobat amb un món del tot nou i que cal crear eines per tal de navegar-hi, d'aquí que hi hagin situacions o objectes nous que requereixen la creació de tractaments o l'elaboració de conceptes per tal de manipular-los.

La gran quantitat de pathways (cascada de processos, xarxa bioquímica, via de senyalització...o en d'altres disciplines és anomenat com a diagrama de flux, sistema de control, pathway de transició, cadenes tròfiques, workflow...) que hi han, no sols en biologia o medicina sinó en tants d'altres camps, conté una gran quantitat d'informació sintetitzada després de molts d'anys de recerca, assaigs i articles; aquesta informació té una gran vàlua empírica, en esser la síntesi de l'estat de la qüestió no hi han unes dades registrades concretes que accompanyin el pathway en el seu marc global però també és informació empírica i molt robusta. En el marc de l'assignatura del màster de Biologia Estructural, hi ha una gran quantitat de malalties descrites com a pathway

## 7. Conclusions

possibles línies de continuació de desenvolupament del mètode

El cas d'estudi de Sachs et al. és emprat com a banc de proves (golden standard), tanmateix hom no pretén que la comparança de resultats sigui una reproducció exacta, sinó que hi hagi una convergència en la coherència dels resultats. Tot i que el mètode proposat no té precedents directes i suposa un repte dotar-lo d'eines i de claus d'interpretació, els resultats són prou coherents amb els resultats de l'article original. També cal tenir en compte que és una metodologia en desenvolupament. Dit això, atès que els resultats en inferència causal del mètode semblen convergir amb els resultats de Sachs et al., hom publica aquest mètode a dins dels repositoris GitHub i Zenodo a fi de facilitar-ne la reutilització, del codi i la verificació del mètode per part d'altres recercadors.

## 8. Referències

### 8.1 Llibres constitutius

=> Bosch Rué, Anna et al., Deep learning principios y fundamentos Editorial UOC 2025 => Babette A.Brumback 2022, Fundamentals of Causal Inference: With R (Chapman & Hall/CRC Texts in Statistical Science) => Babette A.Brumback 2022 Odd Exercises Solutions Manual for Fundamentals of Causal Inference (versió gairebé completa 59 pàg.) => Castellví, Jordi combinatòria i teoria de grafs FME 2020 => Chollet, François Deep Learning with R Manning Publications 2017 => Genuer, Robin et al. Random Forests with R Springer 2020 <https://doi.org/10.1007/978-3-030-56485-8> => Lefebvre, François 2022, Study project Brumback: Fundamentals of Causal Inferences With R Tidyverse, <https://franklef.github.io/FundamentalsCausalInference/Index.html> => Lantz, Brett Machine Learning with R Packt Publishing 2015 => Pearl, Judea; Glymour, Madelyn; Jewell, Nicholas P. 2016, Causal Inference in Statistics A Primer => Pearl, Judea; Glymour, Madelyn; Jewell, Nicholas P. (Text Authors) & Judea Pearl, Ang Li, Andrew Forney, and Johannes Textor (Solution Authors) Causal Inference in Statistics: A Primer Solution Manual (versió completa 76 pàg.) => Kowalczyk, Alexandre Support Vector Machines Succinctly Syncfusion 2017 => Rocafort, Alfredo & Amat, Oriol, 2017, Com fer recerca (treball de fi de grau, tesi de màster, tesi doctoral i altres projectes de recerca) => Trias Pairó, Joan Matemàtica discreta. Problemes resolts EDICIONS UPC 2001

### 8.2 Llibres consultius

=>Cobo Erik et al., 2007, Bioestadística Para no Estadísticos. Bases Para Interpretar Artículos Científicos =>Cunningham Scott, 2021, Causal Inference The Mixtape =>Lash Timothy L. et al., 2021, Modern Epidemiology =>Nguyen Mike, 2023, A Guide on Data Analysis =>[https://bookdown.org/mike/data\\_analysis/](https://bookdown.org/mike/data_analysis/) =>Rosenbaum Paul R., 2017, Observation and Experiment An Introduction to Causal Inference =>Shipley Bill, 2016, Cause and Correlation in Biology

### 8.3 Treballs acadèmics: TFG, TFM, Tesis, Seminaris

Bougioukas Konstantinos I., 2023, Practical Statistics in Medicine with R <https://bougioukas-medstats-r.netlify.app/> Macías Piguave, d'Amanda Elizabeth, 2022, TFM Grafs Acíclics Dirigits (DAGs) i reproductibilitat d'estudis observacionals <https://upcommons.upc.edu/handle/2117/362818>

### 8.4 Articles

Suchs, Karen et al. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data Article in Science · May 2005 DOI: 10.1126/science.1105809 · Source: PubMed <https://www.researchgate.net/publication/7891495>

Ankan Ankur Testing Graphical Causal Models Using the R Package “dagitty” Current Protocols, 1, e45. doi: 10.1002/cpz1.45 <https://currentprotocols.onlinelibrary.wiley.com/doi/10.1002/cpz1.45>

Tennant Peter WG et al., 2019, Use of directed acyclic graphs (DAGs) in applied health research: review and recommendations <https://pubmed.ncbi.nlm.nih.gov/33330936/>

Bengio Yoshua et al. Deep learning 8 MAY 2015 doi:10.1038/nature14539  
<https://www.nature.com/articles/nature14539>

## ANNEX1 CODI R COMENTAT

```
knitr:::opts_chunk$set(rgl = TRUE)
# a la consola , no a dins del Rmd
#install.packages("remotes")
#remotes::install_github("jtextor/dagitty/r")
library(dagitty)
library(ggdag)

##
## Attaching package: 'ggdag'

## The following object is masked from 'package:stats':
##      filter

library(graph)

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##      IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##      anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##      colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##      get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##      match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##      Position, rank, rbind, Reduce, rownames, sapply, setdiff, table,
##      tapply, union, unique, unsplit, which.max, which.min

##
## Attaching package: 'graph'

## The following object is masked from 'package:dagitty':
##      edges

library(igraph)

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:graph':
##      degree, edges, intersection, union
```

```
## The following objects are masked from 'package:BiocGenerics':  
##  
##     normalize, path, union  
  
## The following object is masked from 'package:dagitty':  
##  
##     edges  
  
## The following objects are masked from 'package:stats':  
##  
##     decompose, spectrum  
  
## The following object is masked from 'package:base':  
##  
##     union  
  
library(rgl)  
library(ggplot2)  
library(plot3D)  
library(scales)  
library(dplyr)  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:igraph':  
##  
##     as_data_frame, groups, union  
  
## The following object is masked from 'package:graph':  
##  
##     union  
  
## The following objects are masked from 'package:BiocGenerics':  
##  
##     combine, intersect, setdiff, union  
  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union  
  
packageVersion("dagitty")  
## [1] '0.3.4'  
  
packageVersion("ggdag")  
## [1] '0.2.13'  
  
packageVersion("igraph")  
## [1] '2.1.4'  
  
packageVersion("graph")
```

```
## [1] '1.82.0'
```

Hom carrega l'aplec de dades

```
dades <- read.csv("C:/Users/graup/Desktop/TFM/protocol3/protein_signal.csv")
head(dades)

##          Raf      Mek     Plcg      PIP2      PIP3      Erk      Akt      PKA
## 1 3.179665 3.030625 3.222752 2.836624 3.386284 1.910165 2.179628 3.464657
## 2 3.313155 3.127535 3.367189 2.799483 2.527001 2.359419 2.461049 3.394200
## 3 3.531847 3.554489 3.441636 2.582776 2.730851 2.263097 2.461049 3.452962
## 4 3.621384 3.828081 3.640896 2.704508 1.727649 1.855651 2.021082 3.570291
## 5 3.285691 3.206716 2.992451 2.562289 3.011359 2.414187 2.612864 3.331957
## 6 3.032219 2.484086 3.522796 2.918565 2.654335 2.165465 2.359102 3.632987
##          PKC      P38      Jnk
## 1 3.705143 3.921797 3.566766
## 2 3.002324 3.487035 3.753581
## 3 3.531599 3.773342 3.254741
## 4 3.611415 3.725917 3.328318
## 5 3.143080 3.679484 3.874796
## 6 3.611415 3.960633 3.726634
```

## 1. REDUCCIÓ D'UN PATHWAY SENSE CICLES A ARBRE

### 1.1 Representació del pathway o network biològic (xarxa de dependències direccionals)

Aquest pathway o network conté retroalimentacions funcionals o feedbacks biològics a fi de fer-lo coherent amb la bioquímica coneguda, però abans de passar a un arbre hauran d'esser tractades perquè els cicles no són acceptats per definició d'arbre ni tampoc per definició de DAG en 2D (al model geomètric hom veu que sí que podrien esser possibles cicles locals sense comprometre l'acilcicitat general)

```
pathway <- dagitty('dag {

bb="-0.5,-0.5,0.5,0.5"
Akt [pos="-0.419,0.379"]
Erk [pos="0.243,0.289"]
Jnk [pos="0.071,0.320"]
Mek [pos="0.406,0.126"]
P38 [pos="-0.177,0.378"]
PIP2 [pos="-0.325,-0.203"]
PIP3 [pos="-0.425,-0.442"]
PKA [pos="0.257,-0.434"]
PKC [pos="-0.364,0.170"]
Plcg [pos="-0.065,-0.337"]
Raf [pos="0.455,-0.153"]

PIP3 <-> PIP2
PIP3 <-> Plcg

Mek -> Erk
PIP2 -> PKC
PIP3 -> Akt}
```

```

PKA -> Akt
PKA -> Erk
PKA -> Jnk
PKA -> Mek
PKA -> P38
PKA -> Raf
PKC -> Jnk
PKC -> Mek
PKC -> P38
PKC -> Raf
Plcg -> PIP2
Plcg -> PKC
Raf -> Mek
}
')

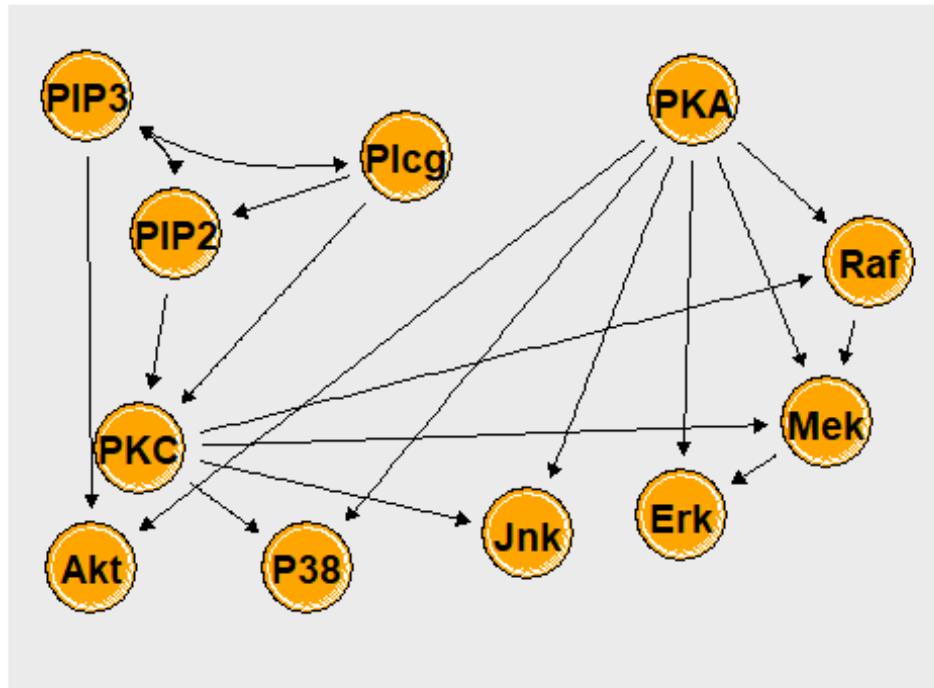
```

```

ggdag(pathway) +
  geom_dag_node(color = "orange", size = 15) +
  geom_dag_text(color = "black", size = 5) +
  coord_cartesian(ylim = rev(c(-0.5,0.5))) +
  ggtitle("pathway o network original") +
  theme ( panel.grid = element_blank(),
          axis.text = element_blank(),
          axis.title = element_blank())

```

pathway o network original



El nombre d'arestes

```

arestes_pathway <- node_status(pathway)$data
table(arestes_pathway[,4])

```

```
## direction
##  -> <-> --
##  16   2   0
```

## 1.2 Garantir l'aciclicitat general abans de reduir el pathway a arbre/bosc

Tal com és reportat el pathway, en aquest cas hi han dues relacions bidireccionals biològiques, PIP3<->PIP2 i PIP2<->Plcg, però cal reduir totes dues a relacions causals unidireccionals. Això és així a fi de fer inferència causal (cal respectar la definició matemàtica d'arbre i DAG 2D), les relacions són unidireccionals. Així que hom tria la direcció que maximitza la probabilitat condicional de l'aplec de dades; altrament dit, amb cicles hom no pot calcular ni la probabilitat condicional  $P(Y|X)$  ni factoritzar una distribució conjunta

$$P(X_1, \dots, X_n) = \prod_i P(X_i | Pa(X_i))$$

A més a més, és normal que les relacions bioquímiques puguin esser recíproques (feedback, cross-talks i cicles) però en un moment determinat del sistema cel·lular hi ha un flux predominant d'informació (d'estímul a resposta), per tant, una direcció causal efectiva que no nega tanmateix la reciprocitat biològica. D'aquí que Sachs i el seu equip parlin d'unidirectionalitat funcional; per exemple, PIP3 és la causa de canvi observat a Plcg, per tant, PIP3->Plcg; també amb PIP3->PIP2, atès que PIP3 és considerat qui genera el senyal (upstream).

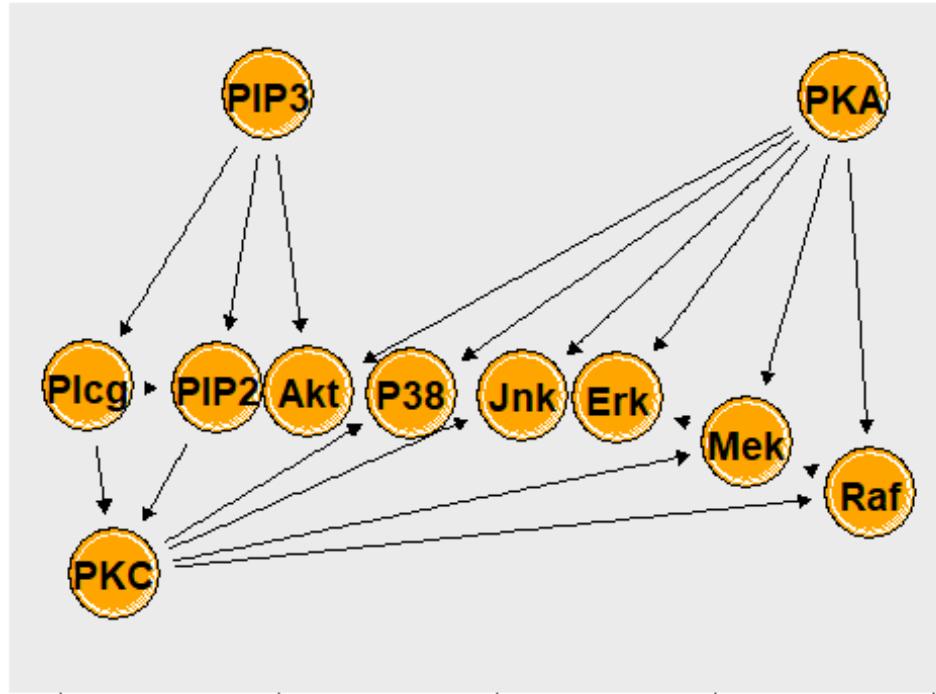
```
pathway_net <- dagitty('dag {
bb="-0.5,-0.5,0.5,0.5"
Akt [pos="-0.214,0.076"]
Erk [pos="0.138,0.088"]
Jnk [pos="0.030,0.082"]
Mek [pos="0.288,0.165"]
P38 [pos="-0.096,0.079"]
PIP2 [pos="-0.320,0.071"]
PIP3 [pos="-0.261,-0.441"]
PKA [pos="0.398,-0.437"]
PKC [pos="-0.438,0.392"]
Plcg [pos="-0.468,0.067"]
Raf [pos="0.429,0.253"]
Mek -> Erk
PIP2 -> PKC
PIP3 -> Akt
PIP3 -> PIP2
PIP3 -> Plcg
PKA -> Akt
PKA -> Erk
PKA -> Jnk
PKA -> Mek
PKA -> P38
PKA -> Raf
PKC -> Jnk
PKC -> Mek
PKC -> P38
PKC -> Raf
Plcg -> PIP2
Plcg -> PKC
Raf -> Mek
}
')
```

```

ggdag(pathway_net, edge_type = "link") +
  geom_dag_node(color = "orange", size = 15) +
  geom_dag_text(color = "black", size = 5) +
  coord_cartesian(ylim = rev(c(-0.5, 0.5))) +
  ggtitle("pathway sense cicles") +
  theme ( panel.grid = element_blank(),
          axis.text = element_blank(),
          axis.title = element_blank())

```

## pathway sense cicles



El nombre d'arestes

```

arestes_pathway_net <- node_status(pathway_net)$data
table(arestes_pathway_net[,4])

## direction
## -> <-> --
## 18   0   0

length(impliedConditionalIndependencies(pathway_net))

## [1] 59

impliedConditionalIndependencies(pathway_net)

## Akt _||_ Erk | Mek, PKA
## Akt _||_ Erk | PKA, PKC
## Akt _||_ Erk | PIP2, PKA, Plcg
## Akt _||_ Erk | PIP3, PKA
## Akt _||_ Jnk | PKA, PKC

```

```

## Akt _||_ Jnk | PIP2, PKA, Plcg
## Akt _||_ Jnk | PIP3, PKA
## Akt _||_ Mek | PKA, PKC
## Akt _||_ Mek | PIP2, PKA, Plcg
## Akt _||_ Mek | PIP3, PKA
## Akt _||_ P38 | PKA, PKC
## Akt _||_ P38 | PIP2, PKA, Plcg
## Akt _||_ P38 | PIP3, PKA
## Akt _||_ PIP2 | PIP3
## Akt _||_ PKC | PIP2, Plcg
## Akt _||_ PKC | PIP3
## Akt _||_ Plcg | PIP3
## Akt _||_ Raf | PKA, PKC
## Akt _||_ Raf | PIP2, PKA, Plcg
## Akt _||_ Raf | PIP3, PKA
## Erk _||_ Jnk | PKA, PKC
## Erk _||_ Jnk | Mek, PKA
## Erk _||_ P38 | PKA, PKC
## Erk _||_ P38 | Mek, PKA
## Erk _||_ PIP2 | PKC
## Erk _||_ PIP2 | Mek, PKA
## Erk _||_ PIP3 | PIP2, Plcg
## Erk _||_ PIP3 | PKC
## Erk _||_ PIP3 | Mek, PKA
## Erk _||_ PKC | Mek, PKA
## Erk _||_ Plcg | PKC
## Erk _||_ Plcg | Mek, PKA
## Erk _||_ Raf | Mek, PKA
## Jnk _||_ Mek | PKA, PKC
## Jnk _||_ P38 | PKA, PKC
## Jnk _||_ PIP2 | PKC
## Jnk _||_ PIP3 | PIP2, Plcg
## Jnk _||_ PIP3 | PKC
## Jnk _||_ Plcg | PKC
## Jnk _||_ Raf | PKA, PKC
## Mek _||_ P38 | PKA, PKC
## Mek _||_ PIP2 | PKC
## Mek _||_ PIP3 | PIP2, Plcg
## Mek _||_ PIP3 | PKC
## Mek _||_ Plcg | PKC
## P38 _||_ PIP2 | PKC
## P38 _||_ PIP3 | PIP2, Plcg
## P38 _||_ PIP3 | PKC
## P38 _||_ Plcg | PKC
## P38 _||_ Raf | PKA, PKC
## PIP2 _||_ PKA
## PIP2 _||_ Raf | PKC
## PIP3 _||_ PKA
## PIP3 _||_ PKC | PIP2, Plcg
## PIP3 _||_ Raf | PKC
## PIP3 _||_ Raf | PIP2, Plcg
## PKA _||_ PKC
## PKA _||_ Plcg
## Plcg _||_ Raf | PKC

```

```
localTests(pathway_net, data = dades, type = "cis")
```

				estimate	p.value	2.5%
##	Akt	Erk	Mek, PKA	0.8970851604	0.000000e+00	0.963463790
##	Akt	Erk	PKA, PKC	0.8976677512	0.000000e+00	0.964436850
##	Akt	Erk	PIP2, PKA, Plcg	0.8969931314	0.000000e+00	0.963310055
##	Akt	Erk	PIP3, PKA	0.8979297031	0.000000e+00	0.964875186
##	Akt	Jnk	PKA, PKC	-0.0571097695	6.600962e-10	-0.075175525
##	Akt	Jnk	PIP2, PKA, Plcg	-0.0667608395	5.140573e-13	-0.084802677
##	Akt	Jnk	PIP3, PKA	-0.0624895340	1.393854e-11	-0.080541934
##	Akt	Mek	PKA, PKC	0.0648078273	2.383684e-12	0.046719287
##	Akt	Mek	PIP2, PKA, Plcg	0.0603038930	6.970457e-11	0.042205815
##	Akt	Mek	PIP3, PKA	0.0617998837	2.328828e-11	0.043705410
##	Akt	P38	PKA, PKC	0.2137041498	1.508917e-121	0.196341569
##	Akt	P38	PIP2, PKA, Plcg	0.1616372551	1.965115e-69	0.143919205
##	Akt	P38	PIP3, PKA	0.1655014964	9.050500e-73	0.147806559
##	Akt	PIP2	PIP3	-0.0136374356	1.407006e-01	-0.031772289
##	Akt	PKC	PIP2, Plcg	-0.0034278992	7.111872e-01	-0.021570048
##	Akt	PKC	PIP3	-0.0039815169	6.671376e-01	-0.022122632
##	Akt	Plcg	PIP3	-0.0638592012	4.937006e-12	-0.081907264
##	Akt	Raf	PKA, PKC	-0.0235946248	1.080260e-02	-0.041720263
##	Akt	Raf	PIP2, PKA, Plcg	-0.0196800065	3.351180e-02	-0.037810778
##	Akt	Raf	PIP3, PKA	-0.0233162749	1.177094e-02	-0.041442241
##	Erk	Jnk	PKA, PKC	-0.0190370631	3.973284e-02	-0.037167720
##	Erk	Jnk	Mek, PKA	-0.0236008250	1.078186e-02	-0.041726456
##	Erk	P38	PKA, PKC	0.2352229982	8.164557e-148	0.218043301
##	Erk	P38	Mek, PKA	0.1875178021	2.286637e-93	0.169962075
##	Erk	PIP2	PKC	-0.0106795997	2.486485e-01	-0.028816730
##	Erk	PIP2	Mek, PKA	-0.0173498184	6.090157e-02	-0.035482143
##	Erk	PIP3	PIP2, Plcg	0.0430115261	3.340170e-06	0.024887458
##	Erk	PIP3	PKC	0.0310118228	8.054193e-04	0.012876355
##	Erk	PIP3	Mek, PKA	0.0007349409	9.367273e-01	-0.017408781
##	Erk	PKC	Mek, PKA	-0.0212918847	2.143877e-02	-0.039420153
##	Erk	Plcg	PKC	-0.0655456540	1.338120e-12	-0.083589237
##	Erk	Plcg	Mek, PKA	-0.0691058037	7.644257e-14	-0.087140372
##	Erk	Raf	Mek, PKA	-0.1036019493	2.880355e-29	-0.121518799
##	Jnk	Mek	PKA, PKC	0.0087265133	3.458812e-01	-0.009418468
##	Jnk	P38	PKA, PKC	0.0321854183	5.057859e-04	0.014050133
##	Jnk	PIP2	PKC	0.0131965556	1.539968e-01	-0.004947342
##	Jnk	PIP3	PIP2, Plcg	0.0614569561	2.999791e-11	0.043361827
##	Jnk	PIP3	PKC	0.0487141516	1.392775e-07	0.030598484
##	Jnk	Plcg	PKC	-0.0426640538	4.001450e-06	-0.060759791
##	Jnk	Raf	PKA, PKC	-0.0679855460	1.913157e-13	-0.086023242
##	Mek	P38	PKA, PKC	0.1085003780	5.852406e-32	0.090535520
##	Mek	PIP2	PKC	0.0431071653	3.173489e-06	0.024983993
##	Mek	PIP3	PIP2, Plcg	-0.0122087025	1.872444e-01	-0.030345472
##	Mek	PIP3	PKC	-0.0099681582	2.815780e-01	-0.028105789
##	Mek	Plcg	PKC	-0.0365465049	7.832827e-05	-0.054653012
##	P38	PIP2	PKC	-0.0370824934	6.138214e-05	-0.055188111
##	P38	PIP3	PIP2, Plcg	0.0356793835	1.154907e-04	0.017547253
##	P38	PIP3	PKC	-0.0015846976	8.640847e-01	-0.019726843
##	P38	Plcg	PKC	-0.0601620129	7.693778e-11	-0.078219545
##	P38	Raf	PKA, PKC	-0.0722846553	5.224399e-15	-0.090310106
##	PIP2	PKA		0.0109783861	2.356352e-01	-0.007164976
##	PIP2	Raf	PKC	0.0470172755	3.725312e-07	0.028899216
##	PIP3	PKA		0.1242992653	1.628804e-41	0.106398171
##	PIP3	PKC	PIP2, Plcg	0.0491058792	1.105999e-07	0.030990002

## PIP3	_  _	Raf	PKC	0.0298126545	1.276224e-03	0.011676257
## PIP3	_  _	Raf	PIP2, Plcg	0.0147883923	1.101614e-01	-0.003355998
## PKA	_  _	PKC		0.1506394654	1.997989e-60	0.132863400
## PKA	_  _	Plcg		0.0147671144	1.106438e-01	-0.003375725
## Plcg	_  _	Raf	PKC	0.0595229433	1.217059e-10	0.041424974
				97.5%		
## Akt	_  _	Erk	Mek, PKA	0.9750846669		
## Akt	_  _	Erk	PKA, PKC	0.9760253392		
## Akt	_  _	Erk	PIP2, PKA, Plcg	0.9749365408		
## Akt	_  _	Erk	PIP3, PKA	0.9764490934		
## Akt	_  _	Jnk	PKA, PKC	-0.0390067031		
## Akt	_  _	Jnk	PIP2, PKA, Plcg	-0.0486755522		
## Akt	_  _	Jnk	PIP3, PKA	-0.0443963911		
## Akt	_  _	Mek	PKA, PKC	0.0828541553		
## Akt	_  _	Mek	PIP2, PKA, Plcg	0.0783626161		
## Akt	_  _	Mek	PIP3, PKA	0.0798540533		
## Akt	_  _	P38	PKA, PKC	0.2309967871		
## Akt	_  _	P38	PIP2, PKA, Plcg	0.1792682785		
## Akt	_  _	P38	PIP3, PKA	0.1831090696		
## Akt	_  _	PIP2	PIP3	0.0045063921		
## Akt	_  _	PKC	PIP2, Plcg	0.0147165062		
## Akt	_  _	PKC	PIP3	0.0141622189		
## Akt	_  _	Plcg	PIP3	-0.0457695295		
## Akt	_  _	Raf	PKA, PKC	-0.0054534693		
## Akt	_  _	Raf	PIP2, PKA, Plcg	-0.0015362876		
## Akt	_  _	Raf	PIP3, PKA	-0.0051749741		
## Erk	_  _	Jnk	PKA, PKC	-0.0008938819		
## Erk	_  _	Jnk	Mek, PKA	-0.0054596728		
## Erk	_  _	P38	PKA, PKC	0.2523594587		
## Erk	_  _	P38	Mek, PKA	0.2049884262		
## Erk	_  _	PIP2	PKC	0.0074645593		
## Erk	_  _	PIP2	Mek, PKA	0.0007939216		
## Erk	_  _	PIP3	PIP2, Plcg	0.0611073861		
## Erk	_  _	PIP3	PKC	0.0491269139		
## Erk	_  _	PIP3	Mek, PKA	0.0188781794		
## Erk	_  _	PKC	Mek, PKA	-0.0031496114		
## Erk	_  _	Plcg	PKC	-0.0474593949		
## Erk	_  _	Plcg	Mek, PKA	-0.0510263145		
## Erk	_  _	Raf	Mek, PKA	-0.0856197056		
## Jnk	_  _	Mek	PKA, PKC	0.0268657507		
## Jnk	_  _	P38	PKA, PKC	0.0502995571		
## Jnk	_  _	PIP2	PKC	0.0313317689		
## Jnk	_  _	PIP3	PIP2, Plcg	0.0795119993		
## Jnk	_  _	PIP3	PKC	0.0667979167		
## Jnk	_  _	Plcg	PKC	-0.0245403370		
## Jnk	_  _	Raf	PKA, PKC	-0.0499036325		
## Mek	_  _	P38	PKA, PKC	0.1263972121		
## Mek	_  _	PIP2	PKC	0.0612020704		
## Mek	_  _	PIP3	PIP2, Plcg	0.0059361025		
## Mek	_  _	PIP3	PKC	0.0081760331		
## Mek	_  _	Plcg	PKC	-0.0184160031		
## P38	_  _	PIP2	PKC	-0.0189525317		
## P38	_  _	PIP3	PIP2, Plcg	0.0537880840		
## P38	_  _	PIP3	PKC	0.0165584913		
## P38	_  _	Plcg	PKC	-0.0420652232		

```

## P38 _||_ Raf | PKA, PKC      -0.0542122957
## PIP2 _||_ PKA                 0.0291145243
## PIP2 _||_ Raf | PKC          0.0651045308
## PIP3 _||_ PKA                 0.1421246064
## PIP3 _||_ PKC | PIP2, Plcg   0.0671895980
## PIP3 _||_ Raf | PKC          0.0479294598
## PIP3 _||_ Raf | PIP2, Plcg   0.0329230513
## PKA _||_ PKC                 0.1683304918
## PKA _||_ Plcg                0.0329002384
## Plcg _||_ Raf | PKC          0.0775820625

```

## 2. SUBGRAF VERTICAL

### 2.1 Generació de l'arbre o bosc (segons sigui un únic node arrel o més d'un)

En aquest cas, atès que hi han dos nodes arrels, hom genera un bosc dirigit

```

llista_arestes <- dagitty::edges(pathway_net)
df_arestes <- data.frame(
  from = llista_arestes$v,
  to = llista_arestes$w )

(g_bosc <- graph_from_data_frame(df_arestes, directed = TRUE))

## IGRAPH 2310e11 DN-- 11 18 --
## + attr: name (v/c)
## + edges from 2310e11 (vertex names):
## [1] Mek ->Erk  PIP2->PKC  PIP3->Akt  PIP3->PIP2  PIP3->Plcg  PKA ->Akt
## [7] PKA ->Erk  PKA ->Jnk  PKA ->Mek  PKA ->P38  PKA ->Raf  PKC ->Jnk
## [13] PKC ->Mek  PKC ->P38  PKC ->Raf  Plcg->PIP2  Plcg->PKC  Raf ->Mek

```

Però hom vol sols les connexions que defineixen el flux causal principal, és a dir un bosc vertical pur, sense relacions laterals

```

#nodes arrel (sense pares)
arrels <- V(g_bosc)$name[degree(g_bosc, mode="in") == 0]
distancies <- sapply(V(g_bosc)$name, function (n) {
  min(distances(g_bosc, v=arrels, to=n, mode="out")), na.rm=TRUE)
})

V(g_bosc)$nivell <- distancies

E(g_bosc)$type <- apply(as.data.frame(as_edgelist(g_bosc)), 1, function(e){
  nivell1 <- V(g_bosc)$nivell[V(g_bosc)$name == e[1]]
  nivell2 <- V(g_bosc)$nivell[V(g_bosc)$name == e[2]]
  if (is.na(nivell2) | is.na(nivell1)) {return("indefinit")}
  } else if (nivell2>nivell1) {return("vertical")}
  } else if (nivell2<nivell1) {return("lateral")}
  } else {return("col·lididor")})
} )

table(E(g_bosc)$type)

```

```

##  

## col·lididor      lateral      vertical  

##            3           4           11  

##  

## FALSE  TRUE  

##    14     4  

##  

## FALSE  TRUE  

##    15     3  

distancies  

##  Mek PIP2 PIP3  PKA   PKC Plcg   Raf   Erk   Akt   Jnk   P38  

##  1     1     0     0     2     1     1     1     1     1     1     1  

(g_bosc_vertical <- igraph::subgraph_from_edges(g_bosc, E(g_bosc)[E(g_bosc)$type ==  

"vertical"], delete.vertices = TRUE))  

## IGRAPH 2331d53 DN-- 11 11 --  

## + attr: name (v/c), nivell (v/n), type (e/c)  

## + edges from 2331d53 (vertex names):  

## [1] PIP2->PKC  PIP3->Akt  PIP3->PIP2  PIP3->Plcg  PKA ->Akt  PKA ->Erk  

## [7] PKA ->Jnk  PKA ->Mek  PKA ->P38  PKA ->Raf  Plcg->PKC

```

El subgraf vertical inclou dos col·lididors, però no són col·lididors que propaguin la causalitat sinó que la dissipen, no són els col·lididors clàssics que integren causes i la transmeten, són col·lididors-fulla (integra senyals però no transmet cap efecte), un col·lididor integrador final de senyals, sense descendents, no obre cap camí nou.

```

#nodes arrel (sense pares)
arrels <- V(g_bosc_vertical)$name[degree(g_bosc_vertical, mode="in") == 0]
#nodes fulla (sense fills)
fulles <- V(g_bosc_vertical)$name[degree(g_bosc_vertical, mode="out") == 0]
#nodes col·lididors (>= 2 pares)
collididors <- V(g_bosc_vertical)$name[degree(g_bosc_vertical, mode="in") >= 2]

noms_nodes <- V(g_bosc_vertical)$name

V(g_bosc_vertical)$role <- ifelse(!is.na(match(noms_nodes,collididors )), "col·lididor-fulla",
                                    ifelse(!is.na(match(noms_nodes, arrels)), "arrel",
                                           ifelse(!is.na(match(noms_nodes, fulles )), "fulla", "mitjancer"
                                           )))

V(g_bosc_vertical)$color <- ifelse(V(g_bosc_vertical)$role == "col·lididor-fulla",
                                      "orangered",
                                      ifelse(V(g_bosc_vertical)$role == "arrel" , "green3",
                                             ifelse(V(g_bosc_vertical)$role == "fulla", "deepskyblue",
                                                   )))  

                                     "gold"

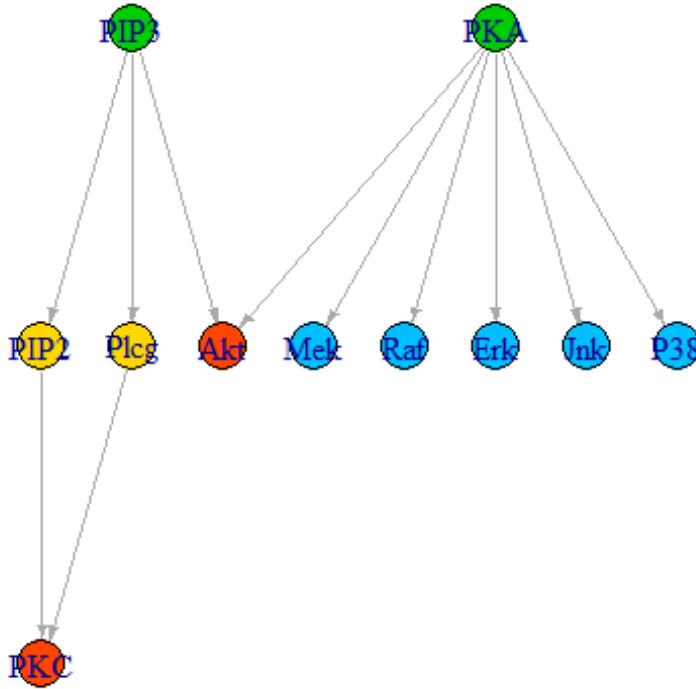
```

```

par(mar=c(0,0,1,0))
plot(g_bosc_vertical,
      layout=layout_as_tree(g_bosc_vertical, root = arrels),
      edge.arrow.size=0.4,
      vertex.size=15,
      main ="subgraf bosc vertical dirigit amb collididors-fulla")

```

## subgraf bosc vertical dirigit amb collididors-fulla



Tanmateix, encara que l'Akt és de natura estructural perquè uneix els dos arbres dirigits verticals, cal respectar la definició d'arbre i treballar amb un bosc dirigit non-connex (on cada component sigui un arbre); el segon, PKC, no uneix pas els arbres del bosc perquè pertany sols a l'arbre d'arrel PIP3. Sigui com sigui un arbre dirigit no pot contenir col·lididors perquè viola la seva estructura fonamental que és d'un sol pare per node; cal doncs restituir la propietat d'arbre a dins del DAG vertical.

Aquí hom tria eliminar les arestes PKA -> Akt i Plcg -> PKC segons el criteri estadístic d'obtenir el màxim de tests d'independències condicionals coherents amb les dades. Així, en treure aquestes dues arestes, hom desfà els col·lididors-fulla i el model millora en coherència perquè deixen de violar el principi <d'un pare per node> propi d'un arbre. La implicació o dependència causal a un arbre són o bé cadena (A->B->C) o bé A->(B,C), el bosc resultant és una estructura de transmissió unidireccional sense convergència d'arestes a un node (entre qualsevol parell de nodes hi ha exactament un únic camí dirigit).

El subgraf vertical és compost de cadenes de verticals de 2 a 3 nodes

```

g_bosc_vertical1 <- subgraph_from_edges(g_bosc_vertical, eids = c(1:4,6:10))

arrels1 <- V(g_bosc_vertical1)$name[degree(g_bosc_vertical1, mode="in") == 0]
fulles1 <- V(g_bosc_vertical1)$name[degree(g_bosc_vertical1, mode="out") == 0]
noms_nodes1 <- V(g_bosc_vertical1)$name

V(g_bosc_vertical1)$role <- ifelse(!is.na(match(noms_nodes1, arrels1)), "arrel",
                                         ifelse(!is.na(match(noms_nodes1,fulles1 )), "fulla", "mitjancer"))

```

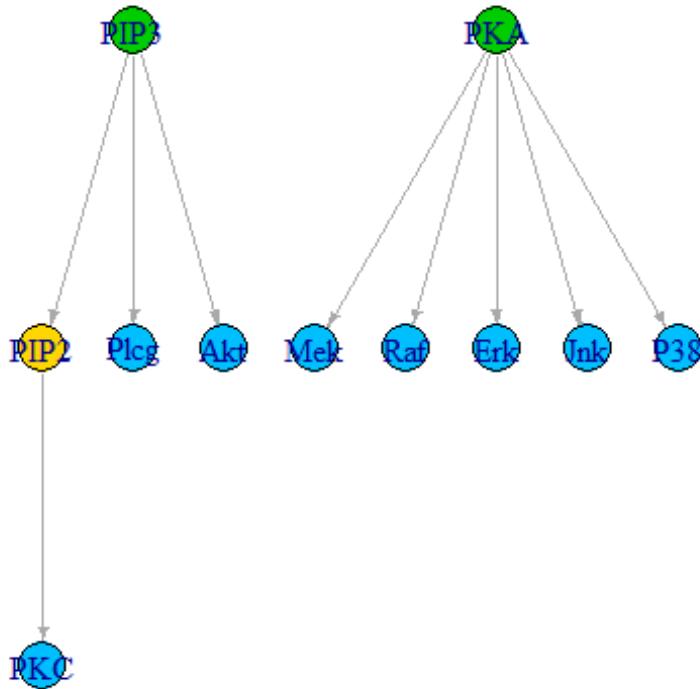
```

V(g_bosc_vertical1)$color <- ifelse(V(g_bosc_vertical1)$role == "arrel" , "green3",
                                    ifelse(V(g_bosc_vertical1)$role == "fulla", "deepskyblue",
                                          ))
"gold"

par(mar=c(0,0,1,0))
plot(g_bosc_vertical1,
      layout=layout_as_tree(g_bosc_vertical1, , root = arrels),
      edge.arrow.size=0.4,
      vertex.size=15,
      main ="subgraf bosc vertical dirigit sense col·lididors-fulla")

```

### subgraf bosc vertical dirigit sense col·lididors-fulla



## 3. SUBGRAF RETROLATERAL

El graf retrolateral sortint és non-connex (no tots els parells de vèrtexs són units com a mínim per un camí); és esperable que en sigui, de non-connex, perquè les connexions laterals no cal que mantinguin la continuïtat del flux vertical principal. En canvi, per definició, els arbres d'un bosc han d'esser connexos (el bosc pot esser non-connex tanmateix). El subgraf retrolateral representa relacions locals o complementàries entre blocs de la xarxa

```

(g_retrolateral <- igraph::subgraph_from_edges(g_bosc, E(g_bosc)[E(g_bosc)$type %in%
c("col·lididor", "lateral")], delete.vertices = TRUE))

## IGRAPH 235cd85 DN-- 8 7 --
## + attr: name (v/c), nivell (v/n), type (e/c)
## + edges from 235cd85 (vertex names):
## [1] Mek ->Erk   PKC ->Jnk   PKC ->Mek   PKC ->P38   PKC ->Raf   Plcg->PIP2  Raf ->Mek

```

Hom reporta que hi han dos components connexos: un en forma de cadena mínima (dos nodes) i el segon en forma de xarxa (6 nodes): aquest segon component és interessant d'estudiar perquè conté el Feed-Forward Loop (FFL) PKC-Raf-Mek (amb el mitjancer Raf) versus PKC-Mek (camí directe entre PKC i Mek), la resta són arestes lineals (no creen camins redundants). El layout de Sugiyama calcula per nivells a fi de mantenir la direccionalitat acíclica del DAG.

Hom ja intueix que el node central del retrolateral és PKC

```
#nodes arrel (sense pares)
arrels <- V(g_retrolateral)$name[degree(g_retrolateral, mode="in") == 0]
#nodes fulla (sense fills)
fulles <- V(g_retrolateral)$name[degree(g_retrolateral, mode="out") == 0]
#nodes col·lididors (>= 2 pares)
collididors <- V(g_retrolateral)$name[degree(g_retrolateral, mode="in") >= 2]

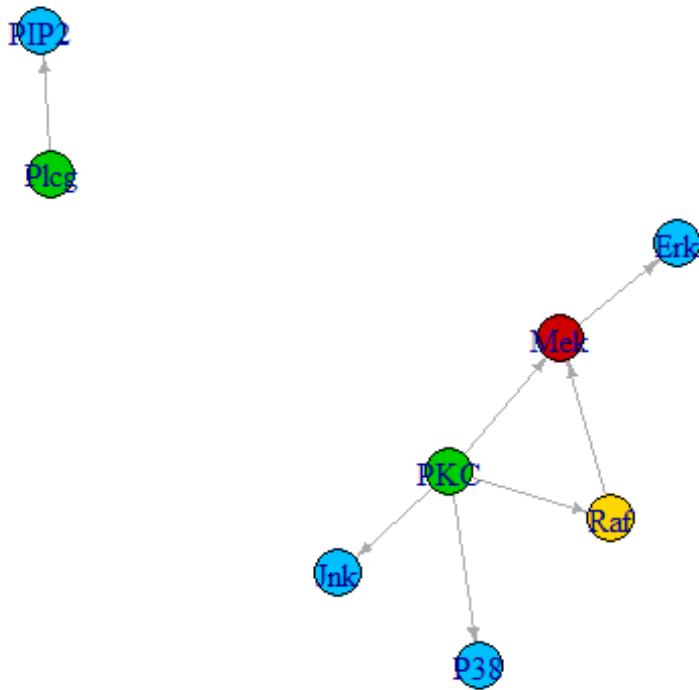
noms_nodes <- V(g_retrolateral)$name

V(g_retrolateral)$role <- ifelse(!is.na(match(noms_nodes,collididors )), "col·lididor",
                                    ifelse(!is.na(match(noms_nodes, arrels)), "arrel",
                                           ifelse(!is.na(match(noms_nodes,fulles )), "fulla", "mitjancer")))

V(g_retrolateral)$color <- ifelse(V(g_retrolateral)$role == "col·lididor", "red3",
                                      ifelse(V(g_retrolateral)$role == "arrel" , "green3",
                                             ifelse(V(g_retrolateral)$role == "fulla", "deepskyblue",
                                                   "gold"))

set.seed(777)
par(mar=c(0,0,1,0))
plot(g_retrolateral,
      layout=layout_with_gem(g_retrolateral),
      edge.arrow.size=0.4,
      vertex.size=15,
      main ="subgraf retrolateral non-connex dirigit")
```

## subgraf retrolateral non-connex dirigit



Però ara cal incloure al lateral les dues arestes sobrants dels col·lididors-fulla trencats al vertical:

```

#g_retrolateral1 <- g_retrolateral
#g_retrolateral1 <- igraph::add_edges(g_retrolateral1, c("PKA", "Akt"))

# Copiem l'original
g_retrolateral1 <- g_retrolateral

# Nodes que falten:
nodes_falten <- c("PKA", "Akt")

# Afegim els nodes (si no hi són)
g_retrolateral1 <- igraph::add_vertices(g_retrolateral1,
                                         nv = length(nodes_falten),
                                         name = nodes_falten)

# Areistes que falten:
arestes_falten <- c("PKA", "Akt", "PIP2", "PKC")

# Afegim les areistes
g_retrolateral1 <- igraph::add_edges(g_retrolateral1, arestes_falten)

g_retrolateral1

## IGRAPH 2376bf7 DN-- 10 9 --
## + attr: name (v/c), nivell (v/n), role (v/c), color (v/c), type (e/c)
## + edges from 2376bf7 (vertex names):
## [1] Mek ->Erk  PKC ->Jnk  PKC ->Mek  PKC ->P38  PKC ->Raf  Plcg->PIP2  Raf ->Mek
## [8] PKA ->Akt  PIP2->PKC
  
```

Si hom descompon el subgraf retrolateral en patrons causals bàsics irreductibles de teoria de grafs (és a dir, de mínima complexitat topològica: calen tres nodes adjacents en la direcció de les arestes, sols dues arestes i un únic node intermedi; són acíclics i non-redundants, sols hi ha una via entre A i C); si hom fa aquesta descomposició elemental del subgraf retrolateral, tots són composts per arestes lateroverticals o retrolaterals, sempre hi ha com a mínim una aresta lateral (amb tres nodes hom pot construir tots els tipus causals: forques o confonedors, col·lididors, cadenes... encara que segons la definició formal en teoria de grafs una cadena també pugui esser composta d'un o més nodes, però en aquest model hom pren un mínim de 3 nodes).

Amb dos nodes, hon no pot distingir cap d'aquests patrons, amb dos nodes sols hi ha una direcció simple sense context estructural. A més a més, segons el model geomètric, calen 3 nodes adjacents per construir un pla causal (determinat per dos vectors), per calcular un angle (entre fluxos o entre un flux i l'eix de Z de referència) o per definir la normal (que dóna el pla de col·lididor o forca)

```
set.seed(777)
par(mar=c(0,0,1,0))

arrels <- V(g_retrolateral1)$name[degree(g_retrolateral1, mode="in") == 0]
#nodes fulla (sense fills)
fulles <- V(g_retrolateral1)$name[degree(g_retrolateral1, mode="out") == 0]
#nodes col·lididors (>= 2 pares)
collididors <- V(g_retrolateral1)$name[degree(g_retrolateral1, mode="in") >= 2]

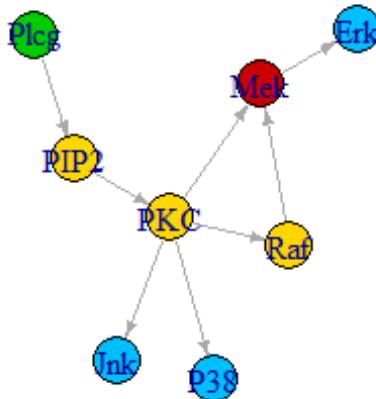
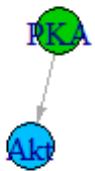
noms_nodes <- V(g_retrolateral1)$name

V(g_retrolateral1)$role <- ifelse(!is.na(match(noms_nodes,collididors )), "col·lididor",
                                    ifelse(!is.na(match(noms_nodes, arrels)), "arrel",
                                           ifelse(!is.na(match(noms_nodes,fulles )), "fulla", "mitjancer"
)))))

V(g_retrolateral1)$color <- ifelse(V(g_retrolateral1)$role == "col·lididor", "red3",
                                      ifelse(V(g_retrolateral1)$role == "arrel", "green3",
                                             ifelse(V(g_retrolateral1)$role == "fulla", "deepskyblue",
                                                   "gold"))

plot(g_retrolateral1,
      layout=layout_with_gem(g_retrolateral1),
      edge.arrow.size=0.4,
      vertex.size=15,
      main ="subgraf retrolateral non-connex dirigit")
```

## subgraf retrolateral non-connex dirigit



A cop d'ull hom ja veu que el node implicat en més de tipus causals elementals és el node PKC: implicat en una forca múltiple, una cadena i un col·lididor. És el node que més distorsiona el flux causal general i cal tenir molt en compte.

Havent aïllada l'estructura vertical, hom examina el retrolateral, no amb finalitat causal definitiva sinó com a pas exploratori per identificar patrons de confusió, col·lisionadors, redundàncies o dependències laterals que podrien comprometre la inferència, tot fent una anàlisi separada del subgraf vertical, és a dir, la proposta d'entrada és analitzar-ho per dimensions aïllades a fi de construir un mapa de riscos causals abans d'afrontar el model complet. L'anàlisi progressiva de les implicacions de les arestes horizontals sobre el subgraf vertical, sense perdre de vista quines interaccions aporten informació vertical i quines introduceixen soroll lateral.

```

detectar_motius_elementals <- function(g) {

  motius <- list()

  # Per cada node B (possible node medial)
  for (b in V(g)) {
    b_name <- V(g)[b]$name

    # Nodes que apunten cap a B (A -> B)
    in_nodes <- names(neighbors(g, b, mode = "in"))

    # Nodes que surten de B (B -> C)
    out_nodes <- names(neighbors(g, b, mode = "out"))

    # (1) CADENA: A -> B -> C
    if (length(in_nodes) == 1 && length(out_nodes) == 1) {
      motius[[length(motius) + 1]] <- list(
        in_node = in_nodes[1],
        out_node = out_nodes[1]
      )
    }
  }
}
  
```

```

        tipus = "Cadena",
        nodes = c(in_nodes, b_name, out_nodes)
    )
}

# (2) FORCA: B -> A i B -> C
if (length(out_nodes) >= 2 && length(in_nodes) == 0) {
    comb <- combn(out_nodes, 2)
    for (i in 1:ncol(comb)) {
        motius[[length(motius) + 1]] <- list(
            tipus = "Forca",
            nodes = c(comb[, i][1], b_name, comb[, i][2])
        )
    }
}

# (3) COL·LISIONADOR: A -> B <- C
if (length(in_nodes) >= 2 && length(out_nodes) == 0) {
    comb <- combn(in_nodes, 2)
    for (i in 1:ncol(comb)) {
        motius[[length(motius) + 1]] <- list(
            tipus = "Col·lisionador",
            nodes = c(comb[, i][1], b_name, comb[, i][2])
        )
    }
}

motius_df <- do.call(rbind, lapply(motius, as.data.frame))
return(motius_df)
}

motius_lateral <- detectar_motius_elementals(g_retrolateral1)
print(motius_lateral)

##      tipus nodes
## 1 Cadena Plcg
## 2 Cadena PIP2
## 3 Cadena PKC
## 4 Cadena PKC
## 5 Cadena Raf
## 6 Cadena Mek

components(g_retrolateral)

## $membership
##   Mek PIP2  PKC Plcg  Raf  Erk  Jnk  P38
##   1     2     1     2     1     1     1     1
##
## $csize
## [1] 6 2
##
## $no
## [1] 2

```

```
components(g_retrolateral1)
```

```
## $membership
##   Mek PIP2 PKC Plcg Raf Erk Jnk P38 PKA Akt
##   1     1    1    1    1    1    1    1    2    2
##
## $csize
## [1] 8 2
##
## $no
## [1] 2
```

Abans de continuar, cal comprovar que com a mínim el graf lateral sobretot al final comparteix com a mínim una arrel o node d'enllaç causal amb el vertical a fi de garantir la compatibilitat topològica i causal amb el model global. Si hom fa la comprovació a aquest estadi, la llista no és buida, hi ha un node arrel compartit i el lateral, per tant, no roman aïllat i pot esser entès com una extensió causal del pathway i no pas com una via independenta. Ara per ara, hi han dos subsistemes, vertical i retrolateral, que — a més de compartir nodes interns, un node arrel del retrolateral és a dins del vertical també—, amb la qual cosa són connectats causalment. El mínim de nodes del retrolateral que hom pot trobar-hi és l'aplec de fulles del subgraf vertical.

```
nodes_comuns <- intersect(V(g_bosc_vertical1)$name, V(g_retrolateral1)$name)
arrels_comunes <- intersect(nodes_comuns, setdiff(V(g_bosc_vertical1)$name,
ends(g_bosc_vertical1, es=E(g_bosc_vertical1))[,2]))
arrels_comunes

## [1] "PKA"
```

Però, a més a més, també hi han camins actius, cosa que causa que els subgrafts encara siguin més dependents.

```
arestes_df <- as.data.frame(dagitty::edges(pathway_net))
igraph_pathway_net <- graph_from_data_frame(arestes_df, directed = TRUE)

nodes_ver <- V(g_bosc_vertical1)$name
nodes_lat <- V(g_retrolateral1)$name

distant_gen <- distances(igraph_pathway_net, mode="out")
distant_sub <- distant_gen[nodes_ver, nodes_lat, drop=FALSE]
connexions_df <- as.data.frame(as.table(distant_sub))
connexions_df <- connexions_df[is.finite(connexions_df$Freq),]
colnames(connexions_df) <- c("origen", "destinacio", "llargaria_cami")
(connexions_df <- connexions_df[order(connexions_df$llargaria_cami),])

##      origen destinacio llargaria_cami
## 1       Mek       Mek          0
## 13      PIP2      PIP2          0
## 27      PKC      PKC          0
## 39      Plcg      Plcg          0
## 51      Raf      Raf          0
## 63      Erk      Erk          0
## 76      Jnk      Jnk          0
## 88      P38      P38          0
## 92      PKA      PKA          0
## 108     Akt      Akt          0
## 4       PKA      Mek          1
```

## 5	PKC	Mek	1
## 7	Raf	Mek	1
## 14	PIP3	PIP2	1
## 17	Plcg	PIP2	1
## 24	PIP2	PKC	1
## 28	Plcg	PKC	1
## 36	PIP3	Plcg	1
## 48	PKA	Raf	1
## 49	PKC	Raf	1
## 56	Mek	Erk	1
## 59	PKA	Erk	1
## 70	PKA	Jnk	1
## 71	PKC	Jnk	1
## 81	PKA	P38	1
## 82	PKC	P38	1
## 102	PIP3	Akt	1
## 103	PKA	Akt	1
## 2	PIP2	Mek	2
## 6	Plcg	Mek	2
## 25	PIP3	PKC	2
## 46	PIP2	Raf	2
## 50	Plcg	Raf	2
## 60	PKC	Erk	2
## 62	Raf	Erk	2
## 68	PIP2	Jnk	2
## 72	Plcg	Jnk	2
## 79	PIP2	P38	2
## 83	Plcg	P38	2
## 3	PIP3	Mek	3
## 47	PIP3	Raf	3
## 57	PIP2	Erk	3
## 61	Plcg	Erk	3
## 69	PIP3	Jnk	3
## 80	PIP3	P38	3
## 58	PIP3	Erk	4

## 4. ANÀLISI DELS SUBGRAFS VERTICAL I RETROLATERAL

### Introducció

Havent descompost amb èxit el pathway en els dos subgrafs, vertical i retrolateral, abans de començar el procés de conversió geomètrica, primer topològica i després empírica amb la normalització espacial, a fi d'obtenir un model geomètric 3D amb un espai mètric coherent, on els angles representen orientacions causals i els mòduls representen intensitats de flux comparables. cal una anàlisi prèvia no sols amb cada subgraf sinó també amb el pathway\_net com a DAG general per tal de comprovar empirícament la coherència de cadascun segons les dades.

Per tant, hom podria analitzar empíricament tots dos subgrafs en termes de coherència, tant del vertical a causa del flux causal vertical del qual es deriva i també el de retroflux inclòs en el subgraf retrolateral (hom es val d'un usatge sistemàtic dels tests d'independència condicional amb la funció localTest(), la qual assereix a la hipòtesi bàsica  $H_0$ : X és independenta de Y condicionat en Z, si la

p-vàlua és molt petita  $H_0$  pot esser rebutjada, és a dir, hi ha prou evidència empírica per dir que hi ha dependència estadística).

Una aplicació d'aquesta anàlisi és, per exemple, decidir quina aresta mantenir en el vertical i quina derivar al subgraf retrolateral quan cal descompondre els col·lididors fulla a fi d'obtenir el subgraf vertical.

## 4.1 Vertical amb col·lididors-fulla

Si hom assagés totes les independències condicionals implícites del model amb els col·lididors-fulla —contravenint la definició d'arbre— amb el conjunt de dades: en total surten 45 independències condicionals (la funció localTest() construeix -segons el model vertical- la grandària del basis set de d-separació; és a dir: un DAG amb  $p$  variables/nodes, el basis set conté un test per a cada parella de variables que no són connectades per una aresta, més en concret, per cada parella non-adjacent; i  $m$  és el nombre d'arestes que ja hi ha al subgraf, ja són connectades directament per una aresta i, per tant, per definició no són independents i no cal comprovar-les amb un test d'independència)

```
## tests = - m = - 11 = - 11 = 44
```

```
##
```

Hi han 44 parelles però la funció localTests() en dóna 45 , per tant, hi ha una parella duplicada: "Akt\_\_PKC" (vegeu dessota\*\*\*)

```
dag_ver_AC <- dagitty('dag {
  PIP2->PKC
  PIP3->Akt
  PIP3->PIP2
  PIP3->Plcg
  PKA ->Akt
  PKA ->Erk
  PKA ->Jnk
  PKA ->Mek
  PKA ->P38
  PKA ->Raf
  Plcg->PKC
}')
```

```
node_status(dag_ver_AC)

## # A DAG with 11 nodes and 11 edges
## #
## # A tibble: 18 × 9
##   name      x      y direction to      xend     yend circular status
##   <chr>    <dbl>  <dbl> <fct>    <chr>    <dbl>  <dbl> <lgl>  <fct>
## 1 Akt     -0.0512  0.171 <NA>     <NA>     NA     NA FALSE   <NA>
## 2 Erk      1.61    -3.06 <NA>     <NA>     NA     NA FALSE   <NA>
## 3 Jnk     -0.737   -3.38 <NA>     <NA>     NA     NA FALSE   <NA>
## 4 Mek     -1.18    -2.10 <NA>     <NA>     NA     NA FALSE   <NA>
## 5 P38      0.508   -3.75 <NA>     <NA>     NA     NA FALSE   <NA>
## 6 PIP2    -1.11     3.73 --> PKC     -0.700   4.90 FALSE   <NA>
## 7 PIP3    -0.357    2.33 --> Akt     -0.0512  0.171 FALSE   <NA>
## 8 PIP3    -0.357    2.33 --> PIP2    -1.11    3.73 FALSE   <NA>
## 9 PIP3    -0.357    2.33 --> Plcg   -0.00325 3.87 FALSE   <NA>
## 10 PKA     0.284   -2.12 --> Akt     -0.0512  0.171 FALSE   <NA>
## 11 PKA     0.284   -2.12 --> Erk     1.61    -3.06 FALSE   <NA>
```

## 12	PKA	0.284	-2.12	->	Jnk	-0.737	-3.38	FALSE	<NA>
## 13	PKA	0.284	-2.12	->	Mek	-1.18	-2.10	FALSE	<NA>
## 14	PKA	0.284	-2.12	->	P38	0.508	-3.75	FALSE	<NA>
## 15	PKA	0.284	-2.12	->	Raf	1.70	-1.70	FALSE	<NA>
## 16	PKC	-0.700	4.90	<NA>	<NA>	NA	NA	FALSE	<NA>
## 17	Plcg	-0.00325	3.87	->	PKC	-0.700	4.90	FALSE	<NA>
## 18	Raf	1.70	-1.70	<NA>	<NA>	NA	NA	FALSE	<NA>

\*\*\*aquí hom pot veure la parella duplicada (Akt // PKC) abans esmentada

Segons el model gràfic, aquestes independències condicionals haurien d'esser validades empíricament amb les dades; aquestes són les relacions esperables segons el model topològic.

`impliedConditionalIndependencies(dag_ver_AC)`

## Akt	_  _	Erk		PKA
## Akt	_  _	Jnk		PKA
## Akt	_  _	Mek		PKA
## Akt	_  _	P38		PKA
## Akt	_  _	PIP2		PIP3
## Akt	_  _	PKC		PIP2, Plcg
## Akt	_  _	PKC		PIP3
## Akt	_  _	Plcg		PIP3
## Akt	_  _	Raf		PKA
## Erk	_  _	Jnk		PKA
## Erk	_  _	Mek		PKA
## Erk	_  _	P38		PKA
## Erk	_  _	PIP2		
## Erk	_  _	PIP3		
## Erk	_  _	PKC		
## Erk	_  _	Plcg		
## Erk	_  _	Raf		PKA
## Jnk	_  _	Mek		PKA
## Jnk	_  _	P38		PKA
## Jnk	_  _	PIP2		
## Jnk	_  _	PIP3		
## Jnk	_  _	PKC		
## Jnk	_  _	Plcg		
## Jnk	_  _	Raf		PKA
## Mek	_  _	P38		PKA
## Mek	_  _	PIP2		
## Mek	_  _	PIP3		
## Mek	_  _	PKC		
## Mek	_  _	Plcg		
## Mek	_  _	Raf		PKA
## P38	_  _	PIP2		
## P38	_  _	PIP3		
## P38	_  _	PKC		
## P38	_  _	Plcg		
## P38	_  _	Raf		PKA
## PIP2	_  _	PKA		
## PIP2	_  _	Plcg		PIP3
## PIP2	_  _	Raf		
## PIP3	_  _	PKA		
## PIP3	_  _	PKC		PIP2, Plcg
## PIP3	_  _	Raf		

```

## PKA _||_ PKC
## PKA _||_ Plcg
## PKC _||_ Raf
## Plcg _||_ Raf

```

Ara hom veu quines independències condicionades es compleixen i quines no. El primer test de tots rebutja amb molta d'evidència empírica que hi hagi independència entre Akt i Erk, fins i tot quan hom condiciona per PKA.

```
(ind_con_AC <- localTests(x = dag_ver_AC, data = dades, type = "cis"))
```

			estimate	p.value	2.5%	97.5%
##						
## Akt	_  _	Erk   PKA	0.897496773	0.000000e+00	0.964151270	0.975748774
## Akt	_  _	Jnk   PKA	-0.063864202	4.918173e-12	-0.081912252	-0.045774540
## Akt	_  _	Mek   PKA	0.062165709	1.771451e-11	0.044072715	0.080218169
## Akt	_  _	P38   PKA	0.164431813	7.605615e-72	0.146731388	0.182044958
## Akt	_  _	PIP2   PIP3	-0.013637436	1.407006e-01	-0.031772289	0.004506392
## Akt	_  _	PKC   PIP2, Plcg	-0.003427899	7.111872e-01	-0.021570048	0.014716506
## Akt	_  _	PKC   PIP3	-0.003981517	6.671376e-01	-0.022122632	0.014162219
## Akt	_  _	Plcg   PIP3	-0.063859201	4.937006e-12	-0.081907264	-0.045769530
## Akt	_  _	Raf   PKA	-0.023945372	9.680488e-03	-0.042069817	-0.005805182
## Erk	_  _	Jnk   PKA	-0.021982334	1.755458e-02	-0.040109057	-0.003841153
## Erk	_  _	Mek   PKA	0.073688222	1.534375e-15	0.055619903	0.091708763
## Erk	_  _	P38   PKA	0.194746123	9.498845e-101	0.177241385	0.212168393
## Erk	_  _	PIP2	-0.010855459	2.409214e-01	-0.028991686	0.007287912
## Erk	_  _	PIP3	0.032064527	5.304309e-04	0.013930695	0.050177295
## Erk	_  _	PKC	0.023248146	1.201204e-02	0.005108365	0.041372639
## Erk	_  _	Plcg	-0.065263860	1.664515e-12	-0.083307427	-0.047177799
## Erk	_  _	Raf   PKA	-0.016429114	7.592956e-02	-0.034561528	0.001714110
## Jnk	_  _	Mek   PKA	0.021022716	2.313625e-02	0.002881102	0.039150502
## Jnk	_  _	P38   PKA	0.128469609	2.965852e-44	0.110585753	0.146275891
## Jnk	_  _	PIP2	0.011273915	2.232637e-01	-0.006869425	0.029409837
## Jnk	_  _	PIP3	0.057507448	5.003424e-10	0.039406632	0.075570706
## Jnk	_  _	PKC	0.210998302	1.821313e-118	0.193615808	0.228308526
## Jnk	_  _	Plcg	-0.039341357	2.118966e-05	-0.057442336	-0.021214563
## Jnk	_  _	Raf   PKA	-0.064721663	2.542793e-12	-0.082767447	-0.046633723
## Mek	_  _	P38   PKA	0.124877243	6.929118e-42	0.106977836	0.142700632
## Mek	_  _	PIP2	0.042474902	4.410539e-06	0.024351732	0.060570218
## Mek	_  _	PIP3	-0.006696199	4.694617e-01	-0.024835118	0.011447128
## Mek	_  _	PKC	0.068699870	1.062547e-13	0.050621048	0.086734033
## Mek	_  _	Plcg	-0.035689617	1.148140e-04	-0.053796750	-0.017559051
## Mek	_  _	Raf   PKA	0.732128301	0.000000e+00	0.741124841	0.760521062
## P38	_  _	PIP2	-0.036118493	9.487773e-05	-0.054224929	-0.017988345
## P38	_  _	PIP3	0.021638230	1.939791e-02	0.003497668	0.039764562
## P38	_  _	PKC	0.488494531	0.000000e+00	0.476315918	0.504549890
## P38	_  _	Plcg	-0.047029326	3.695559e-07	-0.065115783	-0.028912060
## P38	_  _	Raf   PKA	-0.057716145	4.338136e-10	-0.075779674	-0.039614920
## PIP2	_  _	PKA	0.010978386	2.356352e-01	-0.007164976	0.029114524
## PIP2	_  _	Plcg   PIP3	0.094706327	1.058395e-24	0.076695572	0.112656686
## PIP2	_  _	Raf	0.046836748	4.123692e-07	0.028719217	0.064923595
## PIP3	_  _	PKA	0.124299265	1.628804e-41	0.106398171	0.142124606
## PIP3	_  _	PKC   PIP2, Plcg	0.049105879	1.105999e-07	0.030990002	0.067189598
## PIP3	_  _	Raf	0.030800015	8.741938e-04	0.012665157	0.048914637
## PKA	_  _	PKC	0.150639465	1.997989e-60	0.132863400	0.168330492
## PKA	_  _	Plcg	0.014767114	1.106438e-01	-0.003375725	0.032900238

```

## PKC _||_ Raf          0.021805821 1.847755e-02 0.003665335 0.039931966
## Plcg _||_ Raf         0.059748921 1.033532e-10 0.041652143 0.077806708

(Basis_set_verAC <- length(impliedConditionalIndependencies(dag_ver_AC)))

## [1] 45

Basis_set <- impliedConditionalIndependencies(dag_ver_AC)
Basis_set[6:7]

## Akt _||_ PKC | PIP2, Plcg
## Akt _||_ PKC | PIP3

parelles <- sapply(Basis_set, function(x) {
  variables <- unlist(x[c("X", "Y")])
  paste(sort(variables), collapse = " __")
})

(duplicats <- names(table(parelles)[table(parelles)>1]))
## [1] "Akt__PKC"

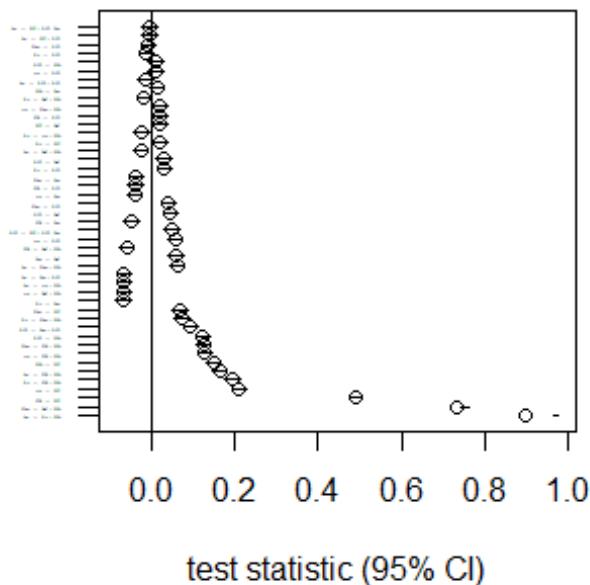
```

Hom representa gràficament els resultats dels tests d'independència condicional

```

ind_con_AC1 <- ind_con_AC[order(abs(ind_con_AC$estimate)),]
plotLocalTestResults(ind_con_AC1, axis.pars = list(cex.axis=0.05, las=1))

```



Aquesta pot esser una manera de quantificar quants de tests coherents hom obté si hom manté el col·lididor-fulla: 9 de coherents versus 36 de fallits

```

ver_coherent_AC <- ifelse(ind_con_AC$p.value<0.05, "violat" , "coherent")
table(ver_coherent_AC)

```

```

## ver_coherent_AC
## coherent violat
##      9      36

violacions <- subset(ind_con_AC, ver_coherent_AC == "violat")
(violacions <- violacions[order(violacions$p.value),])

##                                         estimate      p.value    2.5%   97.5%
## Akt _||_ Erk | PKA          0.89749677 0.000000e+00 0.964151270 0.975748774
## Mek _||_ Raf | PKA          0.73212830 0.000000e+00 0.741124841 0.760521062
## P38 _||_ PKC              0.48849453 0.000000e+00 0.476315918 0.504549890
## Jnk _||_ PKC              0.21099830 1.821313e-118 0.193615808 0.228308526
## Erk _||_ P38 | PKA          0.19474612 9.498845e-101 0.177241385 0.212168393
## Akt _||_ P38 | PKA          0.16443181 7.605615e-72 0.146731388 0.182044958
## PKA _||_ PKC              0.15063947 1.997989e-60 0.132863400 0.168330492
## Jnk _||_ P38 | PKA          0.12846961 2.965852e-44 0.110585753 0.146275891
## Mek _||_ P38 | PKA          0.12487724 6.929118e-42 0.106977836 0.142700632
## PIP3 _||_ PKA              0.12429927 1.628804e-41 0.106398171 0.142124606
## PIP2 _||_ Plcg | PIP3      0.09470633 1.058395e-24 0.076695572 0.112656686
## Erk _||_ Mek | PKA          0.07368822 1.534375e-15 0.055619903 0.091708763
## Mek _||_ PKC              0.06869987 1.062547e-13 0.050621048 0.086734033
## Erk _||_ Plcg              -0.06526386 1.664515e-12 -0.083307427 -0.047177799
## Jnk _||_ Raf | PKA          -0.06472166 2.542793e-12 -0.082767447 -0.046633723
## Akt _||_ Jnk | PKA          -0.06386420 4.918173e-12 -0.081912252 -0.045774540
## Akt _||_ Plcg | PIP3      -0.06385920 4.937006e-12 -0.081907264 -0.045769530
## Akt _||_ Mek | PKA          0.06216571 1.771451e-11 0.044072715 0.080218169
## Plcg _||_ Raf              0.05974892 1.033532e-10 0.041652143 0.077806708
## P38 _||_ Raf | PKA          -0.05771614 4.338136e-10 -0.075779674 -0.039614920
## Jnk _||_ PIP3              0.05750745 5.003424e-10 0.039406632 0.075570706
## PIP3 _||_ PKC | PIP2, Plcg 0.04910588 1.105999e-07 0.030990002 0.067189598
## P38 _||_ Plcg              -0.04702933 3.695559e-07 -0.065115783 -0.028912060
## PIP2 _||_ Raf              0.04683675 4.123692e-07 0.028719217 0.064923595
## Mek _||_ PIP2              0.04247490 4.410539e-06 0.024351732 0.060570218
## Jnk _||_ Plcg              -0.03934136 2.118966e-05 -0.057442336 -0.021214563
## P38 _||_ PIP2              -0.03611849 9.487773e-05 -0.054224929 -0.017988345
## Mek _||_ Plcg              -0.03568962 1.148140e-04 -0.053796750 -0.017559051
## Erk _||_ PIP3              0.03206453 5.304309e-04 0.013930695 0.050177295
## PIP3 _||_ Raf              0.03080002 8.741938e-04 0.012665157 0.048914637
## Akt _||_ Raf | PKA          -0.02394537 9.680488e-03 -0.042069817 -0.005805182
## Erk _||_ PKC              0.02324815 1.201204e-02 0.005108365 0.041372639
## Erk _||_ Jnk | PKA          -0.02198233 1.755458e-02 -0.040109057 -0.003841153
## PKC _||_ Raf              0.02180582 1.847755e-02 0.003665335 0.039931966
## P38 _||_ PIP3              0.02163823 1.939791e-02 0.003497668 0.039764562
## Jnk _||_ Mek | PKA          0.02102272 2.313625e-02 0.002881102 0.039150502

```

## 4.2 Vertical sense col·lididors-fulla

Si hom assagés totes les independències condicionals implícites del model AMB els col·lididors-fulla—contravenint la definició d'arbre— amb el conjunt de dades: en total surten 48 independències condicionals

```

g_bosc_vertical1 <- subgraph_from_edges(g_bosc_vertical, eids = c(2:4,6:11))
df_arestes1 <- igraph::as_data_frame(g_bosc_vertical1, what = "edges")
dag_g_bosc_arestes <- paste0("dag{ ", paste(paste(df_arestes1$from, "->",
df_arestes1$to), collapse=";"), "}")
# hom converteix un objecte igraph a un de dagitty

```

```

g_bosc_vertical2 <- dagitty(dag_g_bosc_arestes)
node_status(g_bosc_vertical2)

## # A DAG with 11 nodes and 9 edges
## #
## # A tibble: 17 × 9
##   name      x      y direction to    xend    yend circular status
##   <chr>   <dbl>  <dbl> <fct>    <chr>   <dbl>  <dbl> <lgl>   <fct>
## 1 Akt     1.81  0.948 <NA>     <NA>    NA    NA FALSE   <NA>
## 2 Erk    -1.50  3.25  <NA>     <NA>    NA    NA FALSE   <NA>
## 3 Jnk    -1.16  2.20  <NA>     <NA>    NA    NA FALSE   <NA>
## 4 Mek    -0.316 4.14  <NA>     <NA>    NA    NA FALSE   <NA>
## 5 P38     0.794  2.52  <NA>     <NA>    NA    NA FALSE   <NA>
## 6 PIP2    1.26  -0.395 <NA>     <NA>    NA    NA FALSE   <NA>
## 7 PIP3    0.750  0.368 ->      Akt     1.81  0.948 FALSE   <NA>
## 8 PIP3    0.750  0.368 ->      PIP2    1.26  -0.395 FALSE   <NA>
## 9 PIP3    0.750  0.368 ->      Plcg   -0.479  0.111 FALSE   <NA>
## 10 PKA   -0.254  3.09  ->     Erk    -1.50  3.25  FALSE   <NA>
## 11 PKA   -0.254  3.09  ->     Jnk    -1.16  2.20  FALSE   <NA>
## 12 PKA   -0.254  3.09  ->     Mek    -0.316 4.14  FALSE   <NA>
## 13 PKA   -0.254  3.09  ->     P38     0.794  2.52  FALSE   <NA>
## 14 PKA   -0.254  3.09  ->     Raf     0.951  3.60  FALSE   <NA>
## 15 PKC   -1.55   0.155 <NA>     <NA>    NA    NA FALSE   <NA>
## 16 Plcg  -0.479  0.111 ->     PKC    -1.55  0.155 FALSE   <NA>
## 17 Raf    0.951  3.60  <NA>     <NA>    NA    NA FALSE   <NA>

(ver_ind_con <- localTests(x = g_bosc_vertical2, data = dades, type = "cis"))

##                                     estimate      p.value      2.5%    97.5%
## Akt _||_ Erk                  0.902993735  0.000000e+00  0.973452628  0.9847412245
## Akt _||_ Jnk                 -0.042084073  5.396793e-06 -0.060180114 -0.0239604312
## Akt _||_ Mek                  0.071429756  1.081988e-14  0.053356993  0.0894561508
## Akt _||_ P38                  0.146384025  4.097659e-57  0.128585924  0.1640981919
## Akt _||_ PIP2 | PIP3          -0.013637436  1.407006e-01 -0.031772289  0.0045063921
## Akt _||_ PKA                  0.226368066  1.108752e-136 0.209111836  0.2435682463
## Akt _||_ PKC | Plcg           -0.003372355  7.156503e-01 -0.021513751  0.0147712617
## Akt _||_ PKC | PIP3           -0.003981517  6.671376e-01 -0.022122632  0.0141622189
## Akt _||_ Plcg | PIP3          -0.063859201  4.937006e-12 -0.081907264 -0.0457695295
## Akt _||_ Raf                  -0.004301928  6.421386e-01 -0.022442113  0.0138410872
## Erk _||_ Jnk | PKA            -0.021982334  1.755458e-02 -0.040109057 -0.0038411532
## Erk _||_ Mek | PKA            0.073688222  1.534375e-15  0.055619903  0.0917087627
## Erk _||_ P38 | PKA            0.194746123  9.498845e-101 0.177241385  0.2121683932
## Erk _||_ PIP2                 -0.010855459  2.409214e-01 -0.028991686  0.0072879118
## Erk _||_ PIP3                 0.032064527  5.304309e-04  0.013930695  0.0501772949
## Erk _||_ PKC                 0.023248146  1.201204e-02  0.005108365  0.0413726395
## Erk _||_ Plcg                -0.065263860  1.664515e-12 -0.083307427 -0.0471777992
## Erk _||_ Raf | PKA            -0.016429114  7.592956e-02 -0.034561528  0.0017141095
## Jnk _||_ Mek | PKA            0.021022716  2.313625e-02  0.002881102  0.0391505021
## Jnk _||_ P38 | PKA            0.128469609  2.965852e-44  0.110585753  0.1462758905
## Jnk _||_ PIP2                 0.011273915  2.232637e-01 -0.006869425  0.0294098368
## Jnk _||_ PIP3                 0.057507448  5.003424e-10  0.039406632  0.0755707059
## Jnk _||_ PKC                 0.210998302  1.821313e-118 0.193615808  0.2283085256
## Jnk _||_ Plcg                -0.039341357  2.118966e-05 -0.057442336 -0.0212145634
## Jnk _||_ Raf | PKA            -0.064721663  2.542793e-12 -0.082767447 -0.0466337232
## Mek _||_ P38 | PKA            0.124877243  6.929118e-42  0.106977836  0.1427006323
## Mek _||_ PIP2                 0.042474902  4.410539e-06  0.024351732  0.0605702176

```

## Mek	_   _	PIP3	-0.006696199	4.694617e-01	-0.024835118	0.0114471275
## Mek	_   _	PKC	0.068699870	1.062547e-13	0.050621048	0.0867340331
## Mek	_   _	Plcg	-0.035689617	1.148140e-04	-0.053796750	-0.0175590507
## Mek	_   _	Raf   PKA	0.732128301	0.000000e+00	0.741124841	0.7605210620
## P38	_   _	PIP2	-0.036118493	9.487773e-05	-0.054224929	-0.0179883449
## P38	_   _	PIP3	0.021638230	1.939791e-02	0.003497668	0.0397645622
## P38	_   _	PKC	0.488494531	0.000000e+00	0.476315918	0.5045498901
## P38	_   _	Plcg	-0.047029326	3.695559e-07	-0.065115783	-0.0289120600
## P38	_   _	Raf   PKA	-0.057716145	4.338136e-10	-0.075779674	-0.0396149204
## PIP2	_   _	PKA	0.010978386	2.356352e-01	-0.007164976	0.0291145243
## PIP2	_   _	PKC   Plcg	-0.009136613	3.236668e-01	-0.027274805	0.0090075933
## PIP2	_   _	PKC   PIP3	-0.018855821	4.164811e-02	-0.036985886	-0.0007133524
## PIP2	_   _	Plcg   PIP3	0.094706327	1.058395e-24	0.076695572	0.1126566864
## PIP2	_   _	Raf	0.046836748	4.123692e-07	0.028719217	0.0649235947
## PIP3	_   _	PKA	0.124299265	1.628804e-41	0.106398171	0.1421246064
## PIP3	_   _	PKC   Plcg	0.046033444	6.491795e-07	0.027914046	0.0641226758
## PIP3	_   _	Raf	0.030800015	8.741938e-04	0.012665157	0.0489146374
## PKA	_   _	PKC	0.150639465	1.997989e-60	0.132863400	0.1683304918
## PKA	_   _	Plcg	0.014767114	1.106438e-01	-0.003375725	0.0329002384
## PKC	_   _	Raf	0.021805821	1.847755e-02	0.003665335	0.0399319659
## Plcg	_   _	Raf	0.059748921	1.033532e-10	0.041652143	0.0778067077

Lleu millora sense els col·lididors-fulla

```
ver_coherent <- ifelse(ver_ind_con$p.value<0.05, "violat", "coherent")
table(ver_coherent)

## ver_coherent
## coherent    violat
##      11       37
```

Tot i que el grup sense col·lididors-fulla mostra una proporció lleugerament més alta de tests coherents (0.25 vs 0.20), la diferència no és estadísticament significativa

```
9/(9+36)
## [1] 0.2
12/(12+36)
## [1] 0.25

prop.test(x = c(12, 9), n = c(48, 45), correct = FALSE)

##
## 2-sample test for equality of proportions without continuity correction
##
## data: c(12, 9) out of c(48, 45)
## X-squared = 0.33214, df = 1, p-value = 0.5644
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.1193051 0.2193051
## sample estimates:
## prop 1 prop 2
## 0.25   0.20
```

### 4.3 Retrolateral

```

dag_retrolat1 <- dagitty('dag {
  Mek ->Erk
  PKC ->Jnk
  PKC ->Mek
  PKC ->P38
  PKC ->Raf
  Plcg->PIP2
  Raf ->Mek

  Plcg->PKC
  PKA ->Akt
}'')

node_status(dag_retrolat1)

## # A DAG with 10 nodes and 9 edges
## #
## # A tibble: 14 × 9
##   name      x      y direction to      xend      yend circular status
##   <chr>    <dbl>  <dbl> <fct>     <chr>    <dbl>  <dbl> <lgl>   <fct>
## 1 Akt     2.12  -1.43 <NA>       <NA>     NA     NA FALSE   <NA>
## 2 Erk     2.77   1.21 <NA>       <NA>     NA     NA FALSE   <NA>
## 3 Jnk    -0.755  1.65 <NA>       <NA>     NA     NA FALSE   <NA>
## 4 Mek     1.68   1.19 ->        Erk      2.77   1.21 FALSE   <NA>
## 5 P38     0.545  2.35 <NA>       <NA>     NA     NA FALSE   <NA>
## 6 PIP2    -1.05  -0.772 <NA>      <NA>     NA     NA FALSE   <NA>
## 7 PKA     1.26  -1.80 ->        Akt      2.12  -1.43 FALSE   <NA>
## 8 PKC     0.339  1.18 ->        Jnk    -0.755  1.65 FALSE   <NA>
## 9 PKC     0.339  1.18 ->        Mek     1.68   1.19 FALSE   <NA>
## 10 PKC    0.339  1.18 ->        P38     0.545  2.35 FALSE   <NA>
## 11 PKC    0.339  1.18 ->        Raf     1.14   0.451 FALSE   <NA>
## 12 Plcg   -0.559  0.145 ->      PIP2    -1.05  -0.772 FALSE   <NA>
## 13 Plcg   -0.559  0.145 ->      PKC     0.339  1.18 FALSE   <NA>
## 14 Raf     1.14   0.451 ->      Mek     1.68   1.19 FALSE   <NA>

(retrolat_ind_con <- localTests(x = dag_retrolat1, data = dades, type = "cis"))

##                               estimate      p.value      2.5%      97.5%
## Akt _||_ Erk             0.9029937349  0.000000e+00  0.9734526285  0.984741225
## Akt _||_ Jnk            -0.0420840727  5.396793e-06 -0.0601801142 -0.023960431
## Akt _||_ Mek             0.0714297558  1.081988e-14  0.0533569931  0.089456151
## Akt _||_ P38             0.1463840249  4.097659e-57  0.1285859240  0.164098192
## Akt _||_ PIP2            -0.0137591544  1.371731e-01 -0.0318931310  0.004383875
## Akt _||_ PKC             -0.0040760213  6.597147e-01 -0.0222163143  0.014066954
## Akt _||_ Plcg            -0.0635519722  6.227368e-12 -0.0816000685 -0.045462466
## Akt _||_ Raf             -0.0043019285  6.421386e-01 -0.0224421128  0.013841087
## Erk _||_ Jnk | PKC      -0.0033273994  7.192794e-01 -0.0214688164  0.014816208
## Erk _||_ Jnk | Mek      -0.0004526791  9.610008e-01 -0.0185952392  0.017690179
## Erk _||_ P38 | PKC      0.1844519209  2.437112e-90  0.1668763313  0.201941614
## Erk _||_ P38 | Mek      0.1638740920  2.310742e-71  0.1461704284  0.181490523
## Erk _||_ PIP2 | Plcg   -0.0029145615  7.528926e-01 -0.0210561611  0.015228957
## Erk _||_ PIP2 | PKC   -0.0106795997  2.486485e-01 -0.0288167303  0.007464559
## Erk _||_ PIP2 | Mek   -0.0144715398  1.179812e-01 -0.0326056942  0.003672137
## Erk _||_ PKA           0.2595964869  3.912551e-181  0.2426542690  0.276547106

```

```

## Erk _||_ PKC | Mek    0.0176055999 5.718066e-02 -0.0005372844 0.035736902
## Erk _||_ Plcg | PKC  -0.0655456540 1.338120e-12 -0.0835892374 -0.047459395
## Erk _||_ Plcg | Mek  -0.0625377810 1.341723e-11 -0.0805892841 -0.044445508
## Erk _||_ Raf  | Mek  -0.0816585822 9.544188e-19 -0.0996543146 -0.063610178
## Jnk _||_ Mek  | PKC  0.0109410287 2.372530e-01 -0.0072031138 0.029077971
## Jnk _||_ P38  | PKC  0.0227900701 1.380969e-02 0.0046492797 0.040915873
## Jnk _||_ PIP2 | Plcg 0.0162113463 7.989786e-02 -0.0019319346 0.034343961
## Jnk _||_ PIP2 | PKC  0.0131965556 1.539968e-01 -0.0049473416 0.031331769
## Jnk _||_ PKA   0.0878301136 1.866810e-21 0.0697995077 0.105804348
## Jnk _||_ Plcg | PKC  -0.0426640538 4.001450e-06 -0.0607597912 -0.024540337
## Jnk _||_ Raf  | PKC  -0.0629291623 9.999543e-12 -0.0809796530 -0.044837653
## Mek _||_ P38  | PKC  0.1011670444 5.603586e-28 0.0831772645 0.119092776
## Mek _||_ PIP2 | Plcg 0.0472158361 3.325840e-07 0.0290980509 0.065302688
## Mek _||_ PIP2 | PKC  0.0431071653 3.173489e-06 0.0249839928 0.061202070
## Mek _||_ PKA   0.0483664567 1.706354e-07 0.0302510671 0.066450171
## Mek _||_ Plcg | PKC  -0.0365465049 7.832827e-05 -0.0546530124 -0.018416003
## P38 _||_ PIP2 | Plcg -0.0306390045 9.309085e-04 -0.0487546341 -0.012503242
## P38 _||_ PIP2 | PKC  -0.0370824934 6.138214e-05 -0.0551881110 -0.018952532
## P38 _||_ PKA   -0.0596135079 1.138728e-10 -0.0776716302 -0.041516481
## P38 _||_ Plcg | PKC  -0.0601620129 7.693778e-11 -0.0782195446 -0.042065223
## P38 _||_ Raf  | PKC  -0.0837399415 1.226794e-19 -0.1017288306 -0.065697122
## PIP2 _||_ PKA   0.0109783861 2.356352e-01 -0.0071649764 0.029114524
## PIP2 _||_ PKC  | Plcg -0.0091366125 3.236668e-01 -0.0272748051 0.009007593
## PIP2 _||_ Raf  | PKC  0.0470172755 3.725312e-07 0.0288992158 0.065104531
## PIP2 _||_ Raf  | Plcg 0.0399106631 1.607761e-05 0.0217837241 0.058011414
## PKA _||_ PKC   0.1506394654 1.997989e-60 0.1328634000 0.168330492
## PKA _||_ Plcg  0.0147671144 1.106438e-01 -0.0033757254 0.032900238
## PKA _||_ Raf   0.0836694106 1.311546e-19 0.0656271734 0.101657765
## Plcg _||_ Raf | PKC  0.0595229433 1.217059e-10 0.0414249739 0.077582063

res_retrolat_coherent05 <- ifelse(retrolat_ind_con$p.value<0.05, "violat", "coherent")
table(res_retrolat_coherent05)

## res_retrolat_coherent05
## coherent  violat
##      15      30

```

## 4.4 Retro

```

dag_retro <- dagitty('dag {

PKC ->Jnk
PKC ->Mek
PKC ->P38
PKC ->Raf
}')


node_status(dag_retro)

## # A DAG with 5 nodes and 4 edges
## #
## # A tibble: 8 × 9
##   name      x      y direction to      xend     yend circular status
##   <chr>   <dbl>  <dbl> <fct>    <chr>   <dbl>  <dbl> <lgl>    <fct>
## 1 Jnk     2.42   0.595 <NA>      <NA>    NA     NA    FALSE    <NA>
## 2 Mek    -0.234  1.17  <NA>      <NA>    NA     NA    FALSE    <NA>

```

```

## 3 P38    1.38   2.21 <NA>      <NA>   NA    NA FALSE <NA>
## 4 PKC    1.09   0.884 ->       Jnk     2.42   0.595 FALSE <NA>
## 5 PKC    1.09   0.884 ->       Mek    -0.234  1.17 FALSE <NA>
## 6 PKC    1.09   0.884 ->       P38     1.38   2.21 FALSE <NA>
## 7 PKC    1.09   0.884 ->       Raf     0.802 -0.439 FALSE <NA>
## 8 Raf     0.802 -0.439 <NA>      <NA>   NA    NA FALSE <NA>

```

L'únic parell que manté la independència condicional (coherent amb la separació d'un DAG) és Jnk // Mek | PKC ( $p = 0.237$ ).

```
(retro_ind_con <- localTests(x = dag_retro, data = dades, type = "cis"))

##                                estimate      p.value      2.5%      97.5%
## Jnk _||_ Mek | PKC  0.01094103 2.372530e-01 -0.007203114  0.02907797
## Jnk _||_ P38 | PKC  0.02279007 1.380969e-02  0.004649280  0.04091587
## Jnk _||_ Raf | PKC -0.06292916 9.999543e-12 -0.080979653 -0.04483765
## Mek _||_ P38 | PKC  0.10116704 5.603586e-28  0.083177264  0.11909278
## Mek _||_ Raf | PKC  0.73316215 0.000000e+00  0.742338082  0.76169116
## P38 _||_ Raf | PKC -0.08373994 1.226794e-19 -0.101728831 -0.06569712
```

És la ràtio més baixa de totes, 1/6, cosa que té lògica perquè és just el contraflux a la causalitat, la informació de les dades va just en sentit contrari.

```
res_retro_coherent05 <- ifelse(retro_ind_con$p.value<0.05, "violat" , "coherent")
table(res_retro_coherent05)

## res_retro_coherent05
## coherent  violat
##      1      5
```

## 4.5 General

```
(pathway_ind_con <- localTests(x = pathway_net, data = dades, type = "cis"))

##                                estimate      p.value      2.5%
## Akt _||_ Erk | Mek, PKA      0.8970851604 0.000000e+00  0.963463790
## Akt _||_ Erk | PKA, PKC      0.8976677512 0.000000e+00  0.964436850
## Akt _||_ Erk | PIP2, PKA, Plcg  0.8969931314 0.000000e+00  0.963310055
## Akt _||_ Erk | PIP3, PKA      0.8979297031 0.000000e+00  0.964875186
## Akt _||_ Jnk | PKA, PKC      -0.0571097695 6.600962e-10 -0.075175525
## Akt _||_ Jnk | PIP2, PKA, Plcg -0.0667608395 5.140573e-13 -0.084802677
## Akt _||_ Jnk | PIP3, PKA      -0.0624895340 1.393854e-11 -0.080541934
## Akt _||_ Mek | PKA, PKC      0.0648078273 2.383684e-12  0.046719287
## Akt _||_ Mek | PIP2, PKA, Plcg 0.0603038930 6.970457e-11  0.042205815
## Akt _||_ Mek | PIP3, PKA      0.0617998837 2.328828e-11  0.043705410
## Akt _||_ P38 | PKA, PKC      0.2137041498 1.508917e-121 0.196341569
## Akt _||_ P38 | PIP2, PKA, Plcg 0.1616372551 1.965115e-69 0.143919205
## Akt _||_ P38 | PIP3, PKA      0.1655014964 9.050500e-73  0.147806559
## Akt _||_ PIP2 | PIP3          -0.0136374356 1.407006e-01 -0.031772289
## Akt _||_ PKC | PIP2, Plcg    -0.0034278992 7.111872e-01 -0.021570048
## Akt _||_ PKC | PIP3          -0.0039815169 6.671376e-01 -0.022122632
## Akt _||_ Plcg | PIP3          -0.0638592012 4.937006e-12 -0.081907264
## Akt _||_ Raf | PKA, PKC      -0.0235946248 1.080260e-02 -0.041720263
## Akt _||_ Raf | PIP2, PKA, Plcg -0.0196800065 3.351180e-02 -0.037810778
## Akt _||_ Raf | PIP3, PKA      -0.0233162749 1.177094e-02 -0.041442241
## Erk _||_ Jnk | PKA, PKC      -0.0190370631 3.973284e-02 -0.037167720
## Erk _||_ Jnk | Mek, PKA      -0.0236008250 1.078186e-02 -0.041726456
```

##	Erk	P38	PKA, PKC	0.2352229982	8.164557e-148	0.218043301
##	Erk	P38	Mek, PKA	0.1875178021	2.286637e-93	0.169962075
##	Erk	PIP2	PKC	-0.0106795997	2.486485e-01	-0.028816730
##	Erk	PIP2	Mek, PKA	-0.0173498184	6.090157e-02	-0.035482143
##	Erk	PIP3	PIP2, Plcg	0.0430115261	3.340170e-06	0.024887458
##	Erk	PIP3	PKC	0.0310118228	8.054193e-04	0.012876355
##	Erk	PIP3	Mek, PKA	0.0007349409	9.367273e-01	-0.017408781
##	Erk	PKC	Mek, PKA	-0.0212918847	2.143877e-02	-0.039420153
##	Erk	Plcg	PKC	-0.0655456540	1.338120e-12	-0.083589237
##	Erk	Plcg	Mek, PKA	-0.0691058037	7.644257e-14	-0.087140372
##	Erk	Raf	Mek, PKA	-0.1036019493	2.880355e-29	-0.121518799
##	Jnk	Mek	PKA, PKC	0.0087265133	3.458812e-01	-0.009418468
##	Jnk	P38	PKA, PKC	0.0321854183	5.057859e-04	0.014050133
##	Jnk	PIP2	PKC	0.0131965556	1.539968e-01	-0.004947342
##	Jnk	PIP3	PIP2, Plcg	0.0614569561	2.999791e-11	0.043361827
##	Jnk	PIP3	PKC	0.0487141516	1.392775e-07	0.030598484
##	Jnk	Plcg	PKC	-0.0426640538	4.001450e-06	-0.060759791
##	Jnk	Raf	PKA, PKC	-0.0679855460	1.913157e-13	-0.086023242
##	Mek	P38	PKA, PKC	0.1085003780	5.852406e-32	0.090535520
##	Mek	PIP2	PKC	0.0431071653	3.173489e-06	0.024983993
##	Mek	PIP3	PIP2, Plcg	-0.0122087025	1.872444e-01	-0.030345472
##	Mek	PIP3	PKC	-0.0099681582	2.815780e-01	-0.028105789
##	Mek	Plcg	PKC	-0.0365465049	7.832827e-05	-0.054653012
##	P38	PIP2	PKC	-0.0370824934	6.138214e-05	-0.055188111
##	P38	PIP3	PIP2, Plcg	0.0356793835	1.154907e-04	0.017547253
##	P38	PIP3	PKC	-0.0015846976	8.640847e-01	-0.019726843
##	P38	Plcg	PKC	-0.0601620129	7.693778e-11	-0.078219545
##	P38	Raf	PKA, PKC	-0.0722846553	5.224399e-15	-0.090310106
##	PIP2	PKA		0.0109783861	2.356352e-01	-0.007164976
##	PIP2	Raf	PKC	0.0470172755	3.725312e-07	0.028899216
##	PIP3	PKA		0.1242992653	1.628804e-41	0.106398171
##	PIP3	PKC	PIP2, Plcg	0.0491058792	1.105999e-07	0.030990002
##	PIP3	Raf	PKC	0.0298126545	1.276224e-03	0.011676257
##	PIP3	Raf	PIP2, Plcg	0.0147883923	1.101614e-01	-0.003355998
##	PKA	PKC		0.1506394654	1.997989e-60	0.132863400
##	PKA	Plcg		0.0147671144	1.106438e-01	-0.003375725
##	Plcg	Raf	PKC	0.0595229433	1.217059e-10	0.041424974
##				97.5%		
##	Akt	Erk	Mek, PKA	0.9750846669		
##	Akt	Erk	PKA, PKC	0.9760253392		
##	Akt	Erk	PIP2, PKA, Plcg	0.9749365408		
##	Akt	Erk	PIP3, PKA	0.9764490934		
##	Akt	Jnk	PKA, PKC	-0.0390067031		
##	Akt	Jnk	PIP2, PKA, Plcg	-0.0486755522		
##	Akt	Jnk	PIP3, PKA	-0.0443963911		
##	Akt	Mek	PKA, PKC	0.0828541553		
##	Akt	Mek	PIP2, PKA, Plcg	0.0783626161		
##	Akt	Mek	PIP3, PKA	0.0798540533		
##	Akt	P38	PKA, PKC	0.2309967871		
##	Akt	P38	PIP2, PKA, Plcg	0.1792682785		
##	Akt	P38	PIP3, PKA	0.1831090696		
##	Akt	PIP2	PIP3	0.0045063921		
##	Akt	PKC	PIP2, Plcg	0.0147165062		
##	Akt	PKC	PIP3	0.0141622189		
##	Akt	Plcg	PIP3	-0.0457695295		

```

## Akt _||_ Raf | PKA, PKC      -0.0054534693
## Akt _||_ Raf | PIP2, PKA, Plcg -0.0015362876
## Akt _||_ Raf | PIP3, PKA      -0.0051749741
## Erk _||_ Jnk | PKA, PKC      -0.0008938819
## Erk _||_ Jnk | Mek, PKA       -0.0054596728
## Erk _||_ P38 | PKA, PKC       0.2523594587
## Erk _||_ P38 | Mek, PKA       0.2049884262
## Erk _||_ PIP2 | PKC          0.0074645593
## Erk _||_ PIP2 | Mek, PKA      0.0007939216
## Erk _||_ PIP3 | PIP2, Plcg    0.0611073861
## Erk _||_ PIP3 | PKC          0.0491269139
## Erk _||_ PIP3 | Mek, PKA      0.0188781794
## Erk _||_ PKC | Mek, PKA       -0.0031496114
## Erk _||_ Plcg | PKC          -0.0474593949
## Erk _||_ Plcg | Mek, PKA      -0.0510263145
## Erk _||_ Raf | Mek, PKA       -0.0856197056
## Jnk _||_ Mek | PKA, PKC       0.0268657507
## Jnk _||_ P38 | PKA, PKC       0.0502995571
## Jnk _||_ PIP2 | PKC          0.0313317689
## Jnk _||_ PIP3 | PIP2, Plcg    0.0795119993
## Jnk _||_ PIP3 | PKC          0.0667979167
## Jnk _||_ Plcg | PKC          -0.0245403370
## Jnk _||_ Raf | PKA, PKC      -0.0499036325
## Mek _||_ P38 | PKA, PKC       0.1263972121
## Mek _||_ PIP2 | PKC          0.0612020704
## Mek _||_ PIP3 | PIP2, Plcg    0.0059361025
## Mek _||_ PIP3 | PKC          0.0081760331
## Mek _||_ Plcg | PKC          -0.0184160031
## P38 _||_ PIP2 | PKC          -0.0189525317
## P38 _||_ PIP3 | PIP2, Plcg    0.0537880840
## P38 _||_ PIP3 | PKC          0.0165584913
## P38 _||_ Plcg | PKC          -0.0420652232
## P38 _||_ Raf | PKA, PKC      -0.0542122957
## PIP2 _||_ PKA                0.0291145243
## PIP2 _||_ Raf | PKC          0.0651045308
## PIP3 _||_ PKA                0.1421246064
## PIP3 _||_ PKC | PIP2, Plcg    0.0671895980
## PIP3 _||_ Raf | PKC          0.0479294598
## PIP3 _||_ Raf | PIP2, Plcg    0.0329230513
## PKA _||_ PKC                0.1683304918
## PKA _||_ Plcg               0.0329002384
## Plcg _||_ Raf | PKC          0.0775820625

general_coherent05 <- ifelse(pathway_ind_con$p.value<0.05, "violat" , "coherent")
table(general_coherent05)

## general_coherent05
## coherent   violat
##        14        45

```

## TEST DE PROPORCIONS

En comparar la proporció de tests coherents entre el model general i el subgraf retrolateral, hom observa que el retrolateral va al capdavant; per exemple, per a un alpha de 0.05, hi ha un 34% al model lateral versus 24% al general, tot i que la diferència no és estadísticament significativa amb una p-vàlua=0.389 a causa de la grandària reduïda del total de tests, sí que referma la idea que la

simplificació estructural afavoreix la consistència causal sense comprometre la coherència global. Altrament dit, els resultats poden refermar la hipòtesi que la descomposició en subgrafs especialitzats pot reduir el soroll condicional i acréixer la robustesa causal.

```
prop.test(c(15,14), c(45,59))

##
## 2-sample test for equality of proportions with continuity correction
##
## data: c(15, 14) out of c(45, 59)
## X-squared = 0.74217, df = 1, p-value = 0.389
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.09890822 0.29099861
## sample estimates:
## prop 1   prop 2
## 0.3333333 0.2372881
```

El resultat d'aquest test (més robust per a mostres petites) és coherent amb el previ en el fet que no hi ha evidència estadística de diferència entre proporcions. L'IC de l'odds ratio inclou l'1 però no és el centre geomètric sinó 1.61, el fet és que hi ha molta de variabilitat, d'aquí un IC molt ample. El resultat mostra una tendència real: una relació positiva consistent segons el DAG, però l'amplada de l'IC no permet assenyalar aquesta direcció positiva amb confiança estadística.

```
fisher.test(matrix(c(15,45-15,14,59-14), nrow=2))

##
## Fisher's Exact Test for Count Data
##
## data: matrix(c(15, 45 - 15, 14, 59 - 14), nrow = 2)
## p-value = 0.3776
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.6195392 4.1663860
## sample estimates:
## odds ratio
## 1.599644
```

## 5. MODEL CAUSAL GEOMÈTRIC 3D

**##Grau d'entrada o in (dependència), grau de sortida o out (efecte) i nivell (jerarquia de flux causal)**

La descomposició causal vertical/retrolateral pot anar una mica més enllà de la teoria de grafs clàssica a fi d'entrar a dins de l'àmbit de la morfologia estructural causal tridimensional. Va més enllà perquè integra dues jerarquies (vertical i retrolateral), és acíclic però no sols jeràrquic (permets relacions horizontals dins d'un mateix nivell causal), preserva la disjunció d'arestes per no pas de nodes (el mateix node pot participar en dos plans causals).

Hom passa d'un plantejament d'*unió purament topològica* (no introduceix noves dimensions, sols reparteix la causalitat entre nivells o verticalitat, graus\_in o arestes entrants i graus\_out o arestes sortints) a un de topoempíric.

L'aparició de *l'espai causal 3D*, el producte vectorial de totes tres dimensions perquè cada node pot existir en totes tres dimensions alhora (no és solsment la unió de Vertical i Retrolateral), aquí hom

defineix un vector per a cada node a fi d'emplaçar-los a dins d'un espai tridimensional, ara ja no és la unió de subgrafs sinó un espai mètric on la distància entre nodes té un significat i les magnituds poden esser contínues i escalaritzables.

\$\$

Graf=(Vertexs, Arestes) $\Rightarrow$  G(V,E) \ subgraf distàncies: G\_D<sup>{sub}</sup>=(V,E\_{vertical}) \ subgraf graus : G\_G<sup>{sub}</sup>=G(V,E)-G<sup>{sub}</sup>(V,E\_{vertical})=(V,E\_{lateral}) \ E\_G=E\_{vertical} E\_{lateral} \ .\ Hom salta de la unió de subgrafs \ G = G\_{vertical} V\_{lateral} \ .\ al producte vectorial \ =(grau\_{in}, grau\_{out}, nivell)  $\Rightarrow$  G\_{3D}=G(R^3) \ .\ Finalment del vectorial al producte cartesià tancat \ (vegeu lasecció model del funcional/empíric deformable)\ (grau\_{in}, grau\_{out}, nivell) \cdot [c,d] \cdot [e,f]  $\Rightarrow$  (S^1 \cdot S^1) \cdot [e,f]=T^2 \cdot [e,f]

\$\$

Hom pot recórrer des del model més senzill com és una cadena tot passant per un DAG que sols té estructura d'arbre, fins a arribar a un DAG amb nivells i relacions retrolaterals.

## 5.1 SUBGRAF VERTICAL TOPOLOGIC

### 5.1.1 Model d'arbre sense els eixos normalitzats

```
deg_in <- degree(g_bosc_vertical1, mode = "in")
deg_out <- degree(g_bosc_vertical1, mode = "out")

coords <- cbind(
  X = deg_out, # eix de sortida
  Y = deg_in, # eix d'entrada
  Z = V(g_bosc_vertical1)$nivell # eix de profunditat vertical
)
coords

##      X Y Z
## Mek   0 1 1
## PIP2  0 1 1
## PIP3  3 0 0
## PKA   5 0 0
## PKC   0 1 2
## Plcg  1 1 1
## Raf   0 1 1
## Erk   0 1 1
## Akt   0 1 1
## Jnk   0 1 1
## P38   0 1 1
```

Els rangs són força desiguals (els eixos requereixen un escalatge)

```
apply(coords, 2, range)

##      X Y Z
## [1,] 0 0 0
## [2,] 5 1 2
```

Eixos sense escalatge

```
library(rgl)

deg_in <- degree(g_bosc_vertical1, mode = "in")
```

```

deg_out <- degree(g_bosc_vertical1, mode = "out")

coords <- cbind(
  X = deg_out, # eix de sortida
  Y = deg_in, # eix d'entrada
  Z = V(g_bosc_vertical1)$nivell # eix de profunditat vertical
)

open3d()

## wgl
## 1

plot3d(coords, col = "red3", size = 1, type = "s")

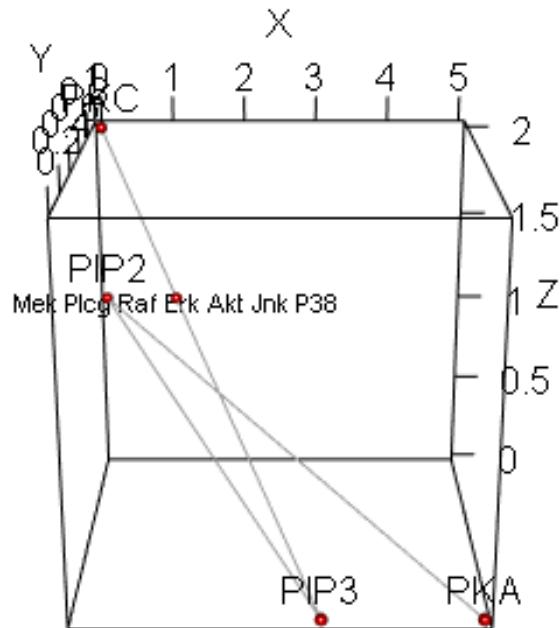
#afegeix arestes
edges <- as_edgelist(g_bosc_vertical1)
for (i in 1:nrow(edges)) {
  v1 <- which(V(g_bosc_vertical1)$name == edges[i,1])
  v2 <- which(V(g_bosc_vertical1)$name == edges[i,2])
  segments3d(rbind(coords[v1,], coords[v2,]), col = "grey60", lwd = 0.9)
}

text3d(coords[2:5,], texts = V(g_bosc_vertical1)$name[-c(1,6:11)], adj = c(0.5, -0.5))
text3d(coords[6,], texts = c("Mek Plcg Raf Erk Akt Jnk P38"), cex=0.7)

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



### 5.1.2 Model d'arbre amb els eixos normalitzats

```
#AQUÍ HOM CENTRA I TIPIFICA ELS EIXOS TOPOLOGICS
coordsn <- scale(coords)
#coodrsn <- apply(coordsn,2,function(col) col/max(abs(col)))
(coordsn <- apply(coordsn,2, function(col) col-min(col)))

##           X      Y      Z
## Mek  0.0000000 2.472066 1.854050
## PIP2 0.0000000 2.472066 1.854050
## PIP3 1.8045994 0.000000 0.000000
## PKA  3.0076656 0.000000 0.000000
## PKC  0.0000000 2.472066 3.708099
## Plcg 0.6015331 2.472066 1.854050
## Raf  0.0000000 2.472066 1.854050
## Erk  0.0000000 2.472066 1.854050
## Akt  0.0000000 2.472066 1.854050
## Jnk  0.0000000 2.472066 1.854050
## P38  0.0000000 2.472066 1.854050
```

Ara els eixos ja són comparables

```
apply(coordsn,2,range)

##           X      Y      Z
## [1,] 0.000000 0.000000 0.000000
## [2,] 3.007666 2.472066 3.708099
```

Tots els nodes són per damunt del pla de masses, cosa que afavoreix el flux causal vertical

```

open3d()

## wgl
##    4

plot3d(coordsn, col = "red3", size = 1, type = "s")

# afegix arestes
edges <- as_edgelist(g_bosc_vertical1)
for (i in 1:nrow(edges)) {
  v1 <- which(V(g_bosc_vertical1)$name == edges[i,1])
  v2 <- which(V(g_bosc_vertical1)$name == edges[i,2])
  segments3d(rbind(coordsn[v1,], coordsn[v2,]), col = "grey60", lwd = 0.9)
}

text3d(coordsn[2:5,], texts = V(g_bosc_vertical1)$name[-c(1,6:11)], adj = c(0.5, -0.5))
text3d(coordsn[6,], texts = c("Mek Plcg Raf Erk Akt Jnk P38"), cex=0.7)

bbox <- c(
  min(coordsn[, "X"]), max(coordsn[, "X"]),
  min(coordsn[, "Y"]), max(coordsn[, "Y"]),
  min(coordsn[, "Z"]), max(coordsn[, "Z"])
)
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

centre <- colMeans(coordsn) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "gold", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(if(centre[1]-dx <xmin) {centre - c(centre[1],0,0)} else {centre -
  c(centre[1]-xmin,0,0)},
  if(centre[1]+dx >xmax) {centre + c(centre[1],0,0)} else {centre +
  c(xmax-centre[1],0,0)}),
  col="orange3", lwd=0.9)
segments3d(rbind(if(centre[2]-dy <ymin) {centre - c(0,centre[2],0)} else {centre -
  c(0,centre[2]-ymin,0)},
  if(centre[2]+dy >ymax) {centre + c(0,centre[2],0)} else {centre +
  c(0,ymax-centre[2],0)}),
  col="orange3", lwd=0.9)
segments3d(rbind(if(centre[3]-dz <zmin) {centre - c(0,0,centre[3])} else {centre -
  c(0,0,centre[3]-zmin)},
  if(centre[3]+dz >zmax) {centre + c(0,0,centre[3])} else {centre +
  c(0,0,zmax-centre[3])}),
  col="orange3", lwd=0.9)

#segments3d(rbind(abs(centre - c(0, dy, 0)), centre + c(0, dy, 0)), col="orange3",

```

```

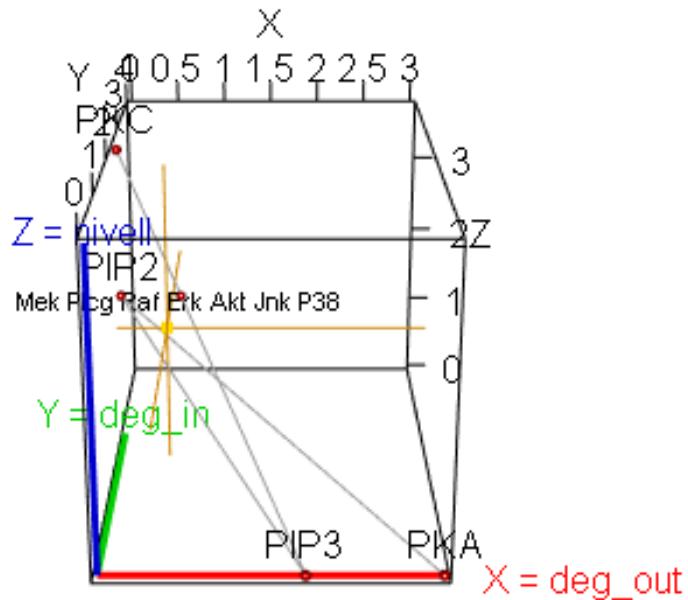
lwd=0.9)
#segments3d(rbind(abs(centre - c(0, 0, dz)), centre + c(0, 0, dz)), col="orange3",
lwd=0.9)

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



## 5.2 SUBGRAF RETROLATERAL TOPOLÒGIC

### ##5.2.1 Model retrolateral sense els eixos normalitzats

L'eix de profunditat vertical: cal incloure la cadena vertical espúria de dos nodes PKA=0 i Akt=1 amb V(g\_bosc)\$nivell

```

deg_in <- degree(g_retrolateral1, mode = "in")
deg_out <- degree(g_retrolateral1, mode = "out")
g_retrolateral2 <- g_retrolateral1

```

```

V(g_retrolateral2)$nivell <- V(g_bosc)$nivell[match(V(g_retrolateral1)$name,
V(g_bosc)$name)]

coords_retrolat <- cbind(
  X = deg_out, # eix de sortida
  Y = deg_in,# eix d'entrada
  Z = V(g_retrolateral2)$nivell
)
coords_retrolat

##      X Y Z
## Mek  1 2 1
## PIP2 1 1 1
## PKC  4 1 2
## Plcg 1 0 1
## Raf   1 1 1
## Erk   0 1 1
## Jnk   0 1 1
## P38   0 1 1
## PKA   1 0 0
## Akt   0 1 1

edges <- as_edgelist(g_retrolateral1)

for (i in 1:nrow(edges)) {
  n1 <- edges[i, 1]
  n2 <- edges[i, 2]

  if (!n1 %in% rownames(coords_retrolat))
    print(paste("NODE INEXISTENT:", n1))

  if (!n2 %in% rownames(coords_retrolat))
    print(paste("NODE INEXISTENT:", n2))
}

apply(coords_retrolat, 2, range)

##      X Y Z
## [1,] 0 0 0
## [2,] 4 2 2

```

Eixos sense escalatge; hom assenyalà les dues arestes espúries verticals (pseudoverticals) com a blaves

```

edges <- as_edgelist(g_retrolateral2)

#spuria1 <- which(edges[,1]=="PKA" & edges[,2]=="Akt")
#spuria2 <- which(edges[,1]=="Plcg" & edges[,2]=="PKC")
#vertexs <- edges[-c(esuria1,esuria2),]

vertexs <- as_edgelist(g_retrolateral2)

open3d()

## wgl
##    7

```

```

plot3d(coords_retrolat, col = "red3", size = 1, type = "s")

for (i in 1:nrow(vertexs)) {
  v1 <- which(V(g_retrolateral2)$name == vertexs[i,1])
  v2 <- which(V(g_retrolateral2)$name == vertexs[i,2])
  segments3d(rbind(coords_retrolat[v1,], coords_retrolat[v2,]), col = "grey60", lwd = 0.9)
}
segments3d(rbind(coords_retrolat[9,], coords_retrolat[10,]), col="skyblue", lwd=1.3)
points3d(coords_retrolat[9,1],coords_retrolat[9,2],coords_retrolat[9,3],col = "skyblue",
size = 2)

segments3d(rbind(coords_retrolat[4,], coords_retrolat[3,]), col="skyblue", lwd=1.3)
points3d(coords_retrolat[4,1],coords_retrolat[4,2],coords_retrolat[4,3],col = "skyblue",
size = 2)

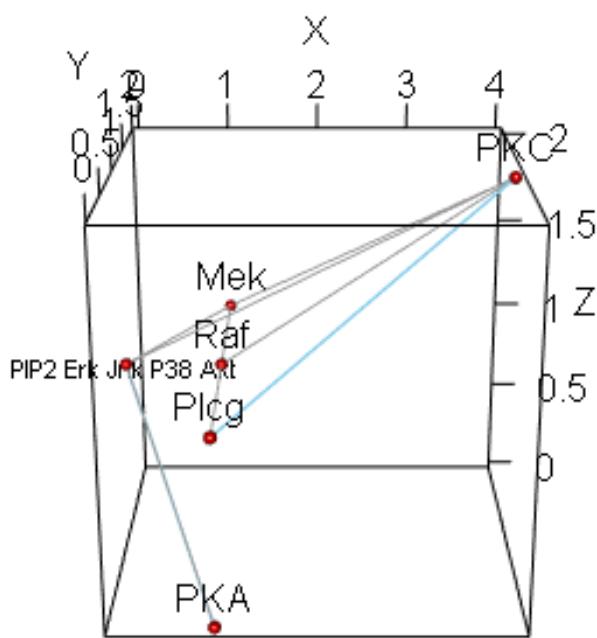
aspect3d(1,1,1)

text3d(coords_retrolat[c(1,3,4,5,9),], texts = V(g_retrolateral2)$name[c(1,3,4,5,9)], adj = c(0.5, -0.5))
text3d(coords_retrolat[6,], texts = c("PIP2 Erk Jnk P38 Akt"), cex=0.7)

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



###5.2.2 Model retrolateral amb els eixos normalitzats

```

coords_retrolatn <- scale(coords_retrolat)
(coords_retrolatn <- apply(coords_retrolatn, 2, function(col) col-min(col)))

##           X         Y         Z
## Mek  0.8352691 3.523321 2.121320
## PIP2 0.8352691 1.761661 2.121320
## PKC  3.3410763 1.761661 4.242641
## Plcg 0.8352691 0.000000 2.121320
## Raf  0.8352691 1.761661 2.121320
## Erk  0.0000000 1.761661 2.121320
## Jnk  0.0000000 1.761661 2.121320
## P38  0.0000000 1.761661 2.121320
## PKA  0.8352691 0.000000 0.000000
## Akt  0.0000000 1.761661 2.121320

apply(coords_retrolatn, 2, range)

##           X         Y         Z
## [1,] 0.000000 0.000000 0.000000
## [2,] 3.341076 3.523321 4.242641

```

Eixos amb escalatge; hom assenyala les dues arestes espúria verticals o pseudoverticals com a blaves si hom pren com a marc de referència el subgraf vertical (arestes provenents de la partió dels dos nodes col·lididors-fulla durant la configuració del subgraf bosc vertical). També cal tenir en compte corregir les arestes verticals que tenen retroflux (les arestes que van en sentit contrari al flux causal vertical); i en segon lloc, les arestes pseudoverticals.

```

edges <- as_edgelist(g_retrolateral2)

espuria1 <- which(edges[,1]=="PKA" & edges[,2]=="Akt")
espuria2 <- which(edges[,1]=="Plcg" & edges[,2]=="PKC")
vertexs <- edges[-c(espuria1,espuria2),]

open3d()

## wgl
## 10

plot3d(coords_retrolatn, col = "red3", size = 1, type = "s")

for (i in 1:nrow(vertexs)) {
  v1 <- which(V(g_retrolateral2)$name == vertexs[i,1])
  v2 <- which(V(g_retrolateral2)$name == vertexs[i,2])
  segments3d(rbind(coords_retrolatn[v1,], coords_retrolatn[v2,]), col = "grey60", lwd = 0.9)
}
segments3d(rbind(coords_retrolatn[9,], coords_retrolatn[10,]), col="skyblue", lwd=0.9)
points3d(coords_retrolatn[9,1],coords_retrolatn[9,2],coords_retrolatn[9,3],col = "skyblue", size = 1)

segments3d(rbind(coords_retrolatn[4,], coords_retrolatn[3,]), col="skyblue", lwd=0.9)
points3d(coords_retrolatn[4,1],coords_retrolatn[4,2],coords_retrolatn[4,3],col = "skyblue", size = 1)

aspect3d(1,1,1)

```

```

text3d(coords_retrolatn[c(1,3,4,5,9),], texts = v(g_retrolateral2$name[c(1,3,4,5,9)]),
adj = c(0.5, -0.5))
text3d(coords_retrolatn[6,], texts = c("PIP2 Erk Jnk P38 Akt"), cex=0.7)

bbox <- c(
  min(coords_retrolatn[, "X"]), max(coords_retrolatn[, "X"]),
  min(coords_retrolatn[, "Y"]), max(coords_retrolatn[, "Y"]),
  min(coords_retrolatn[, "Z"]), max(coords_retrolatn[, "Z"]))
)
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

centre <- colMeans(coords_retrolatn) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "gold", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(if(centre[1]-dx <xmin) {centre - c(centre[1],0,0)} else {centre -
c(centre[1]-xmin,0,0)},
                  if(centre[1]+dx >xmax) {centre + c(centre[1],0,0)} else {centre +
c(xmax-centre[1],0,0)}),
                  col="orange3", lwd=0.9)
segments3d(rbind(if(centre[2]-dy <ymin) {centre - c(0,centre[2],0)} else {centre -
c(0,centre[2]-ymin,0)},
                  if(centre[2]+dy >ymax) {centre + c(0,centre[2],0)} else {centre +
c(0,ymax-centre[2],0)}),
                  col="orange3", lwd=0.9)
segments3d(rbind(if(centre[3]-dz <zmin) {centre - c(0,0,centre[3])} else {centre -
c(0,0,centre[3]-zmin)},
                  if(centre[3]+dz >zmax) {centre + c(0,0,centre[3])} else {centre +
c(0,0,zmax-centre[3])}),
                  col="orange3", lwd=0.9)

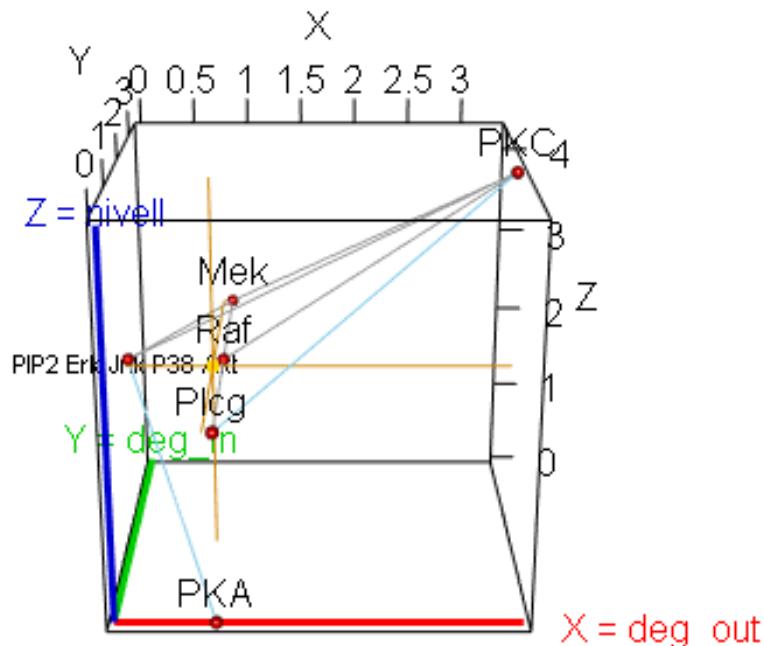
#segments3d(rbind(abs(centre - c(0, dy, 0)), centre + c(0, dy, 0)), col="orange3",
#lwd=0.9)
#segments3d(rbind(abs(centre - c(0, 0, dz)), centre + c(0, 0, dz)), col="orange3",
#lwd=0.9)

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))

rglwidget()

```

```
## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead
```



## 5.3 MODEL GENERAL TOPOLOGIC

### 5.3.1 Model general topològic sense els eixos normalitzats

Hom comença per visualitzar la descomposició en 3D: l'eix Z com la profunditat causal (nivell o distància, els fluxos ascendents/descendents), els eixos X-Y com els plans retrolaterals d'interacció i de retroflux causal (els nodes es connecten transversalment). Les coordenades topològiques són les relacions d'adjacència i jerarquia, cosa que implica una estructura i connectivitat, propietats topològiques del graf. Però abans de fer geometria causal (càlculs geomètrics causals) cal convertir les coordenades en termes d'espai mètric, cal que tinguin significat segons el joc de dades de l'estudi de Sachs et al., en distància (comparances longituds vectorials causals, mòduls), en angles (entre vectors directors)... equivalències entre DAGs i el model geomètric causal a partir d'eines estadístiques com el p\_vàlua o localTest() adaptades al model geomètric p\_geom o localGeomTest().

Ara tot el model topològic (verticalitat i retrolateralitat)

```
deg_in <- degree(g_bosc, mode = "in")
deg_out <- degree(g_bosc, mode = "out")

coords3D <- cbind(
  X = deg_out, # eix de sortida
  Y = deg_in, # eix d'entrada
  Z = V(g_bosc)$nivell # eix de profunditat vertical
)
```

```

coords3D[order(coords3D[,1],coords3D[,2],coords3D[,3]),]

##      X Y Z
## Erk    0 2 1
## Akt    0 2 1
## Jnk    0 2 1
## P38    0 2 1
## PIP2   1 2 1
## Raf    1 2 1
## Mek    1 3 1
## Plcg   2 1 1
## PIP3   3 0 0
## PKC    4 2 2
## PKA    6 0 0

apply(coords3D,2,range)

##      X Y Z
## [1,] 0 0 0
## [2,] 6 3 2

```

El gràfic amb dades sense normalitzar reflecteix l'estructura topològica del DAG, el flux en 3D (també en el model de Tor de qui hom parla més endavant). Caldria visualitzar els nodes coincidents a l'espai sense sobreposició (de fet es resol amb la normalització espacial). Sense normalitzar, cada eix té una escala pròpria, cosa que fa que la geometria depengui de l'escala de mesura i distorsiona el contingut causal (per exemple, l'angle i longitud canvien, si hom en canvia la unitat de mesura), el flux causal en 3D no reflecteix relacions reals sinó disparitats d'escala.

```

deg_in <- degree(g_bosc, mode = "in")
deg_out <- degree(g_bosc, mode = "out")

# -eixos segons graus i nivell vertical
coords3D <- cbind(
  X = deg_out, # eix de sortida
  Y = deg_in,# eix d'entrada
  Z = V(g_bosc)$nivell # eix de profunditat vertical
)

rownames(coords3D) <- V(g_bosc)$name # molt important per evitar desalineació

open3d()

## wgl
## 13

plot3d(coords3D,
        col = "red3",
        size = 1,
        type = "s")

# hom afegeix arestes coherentment segons noms
edges <- as_edgelist(g_bosc, names = TRUE)
edges <- apply(edges, 2, as.character)

for (i in seq_len(nrow(edges))) {

```

```

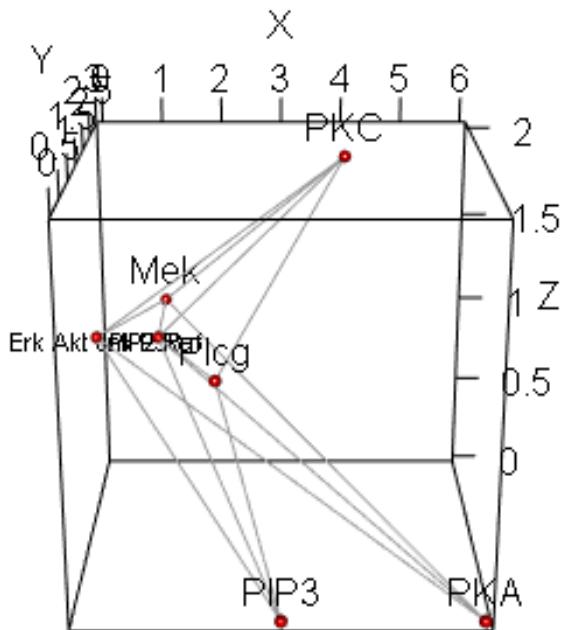
p1 <- as.numeric(coords3D[edges[i, 1], ])
p2 <- as.numeric(coords3D[edges[i, 2], ])
segments3d(rbind(p1, p2), col = "grey60", lwd = 0.9)
}

text3d(coords3D[c(1,3:6),], texts = V(g_bosc)$name[-c(2,7:11)], adj = c(0.5, -0.5))
text3d(coords3D[8:11,], texts = c("Erk Akt Jnk P38"), cex=0.7)
text3d(coords3D[c(2,7),], texts = c("PIP2 Raf"), cex=0.7)

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



### 5.3.2 Model general topològic amb els eixos normalitzats

```

#AQUÍ HOM CENTRA I TIPIFICA ELS EIXOS TOPOLOGICS
coords3Dn <- scale(coords3D)
#coords3Dn <- apply(coords3Dn,2,function(col)col/max(abs(col)))
(coords3Dn <- apply(coords3Dn,2, function(col) col-min(col)+1e-2))

##           X      Y      Z
## Mek  0.5193466 3.255291 1.864050
## PIP2  0.5193466 2.173527 1.864050
## PIP3  1.5380398 0.010000 0.010000
## PKA   3.0660796 0.010000 0.010000
## PKC   2.0473864 2.173527 3.718099
## Plcg  1.0286932 1.091764 1.864050
## Raf   0.5193466 2.173527 1.864050

```

```

## Erk  0.0100000 2.173527 1.864050
## Akt  0.0100000 2.173527 1.864050
## Jnk  0.0100000 2.173527 1.864050
## P38  0.0100000 2.173527 1.864050

apply(coords3Dn, 2, range)

##           X         Y         Z
## [1,] 0.010000 0.010000 0.010000
## [2,] 3.06608 3.255291 3.718099

```

Ara ja amb eixos normalitzats

```

open3d()

## wgl
## 16

rgl.clear()

## Warning in rgl.clear(): 'rgl.clear' is deprecated.
## Use 'clear3d' instead.
## See help("Deprecated")

plot3d(coords3Dn,
       col = "red3",
       size = 0.5,
       type = "s")

# hom afegeix arestes coherentament segons noms
edges <- as_edgelist(g_bosc, names = TRUE)
edges <- apply(edges, 2, as.character)

for (i in seq_len(nrow(edges))) {
  p1 <- as.numeric(coords3Dn[edges[i, 1], ])
  p2 <- as.numeric(coords3Dn[edges[i, 2], ])
  segments3d(rbind(p1, p2), col = "grey60", lwd = 0.9)
}

text3d(coords3Dn[c(1,3:6),], texts = V(g_bosc)$name[-c(2,7:11)], adj = c(0.5, -0.5))
text3d(coords3Dn[8:11], texts = c("Erk Akt Jnk P38"), cex=0.7)
text3d(coords3Dn[c(2,7),], texts = c("PIP2 Raf"), cex=0.7)

bbox <- c(
  min(coords3Dn[, "X"]), max(coords3Dn[, "X"]),
  min(coords3Dn[, "Y"]), max(coords3Dn[, "Y"]),
  min(coords3Dn[, "Z"]), max(coords3Dn[, "Z"])
)
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

centre <- colMeans(coords3Dn) # mitjana de cada eix X,Y,Z
# punt del centre

```

```

points3d(centre[1], centre[2], centre[3], color = "gold", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(if(centre[1]-dx <xmin) {centre - c(centre[1],0,0)} else {centre -
c(centre[1]-xmin,0,0)},
                  if(centre[1]+dx >xmax) {centre + c(centre[1],0,0)} else {centre +
c(xmax-centre[1],0,0)}),
                  col="orange3", lwd=0.9)
segments3d(rbind(if(centre[2]-dy <ymin) {centre - c(0,centre[2],0)} else {centre -
c(0,centre[2]-ymin,0)},
                  if(centre[2]+dy >ymax) {centre + c(0,centre[2],0)} else {centre +
c(0,ymax-centre[2],0)}),
                  col="orange3", lwd=0.9)
segments3d(rbind(if(centre[3]-dz <zmin) {centre - c(0,0,centre[3])} else {centre -
c(0,0,centre[3]-zmin)},
                  if(centre[3]+dz >zmax) {centre + c(0,0,centre[3])} else {centre +
c(0,0,zmax-centre[3])}),
                  col="orange3", lwd=0.9)

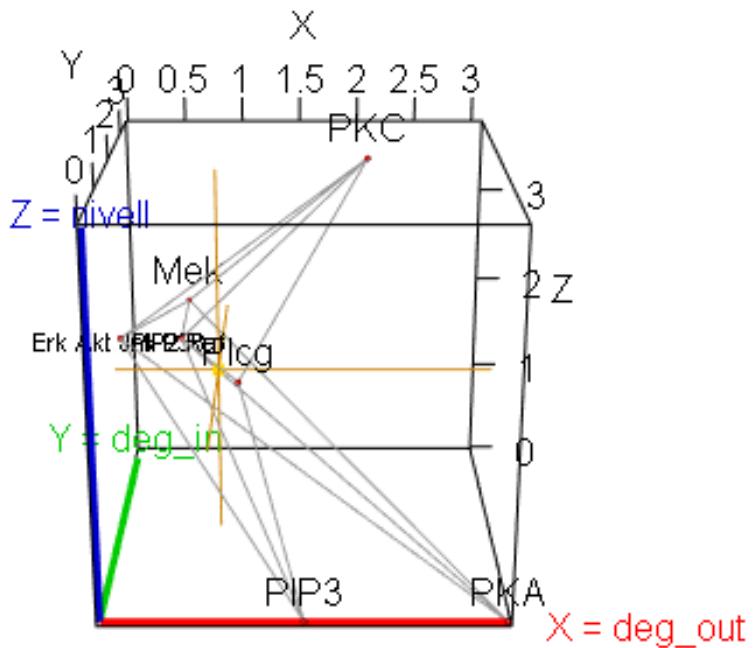
#segments3d(rbind(abs(centre - c(0, dy, 0)), centre + c(0, dy, 0)), col="orange3",
#Lwd=0.9)
#segments3d(rbind(abs(centre - c(0, 0, dz)), centre + c(0, 0, dz)), col="orange3",
#Lwd=0.9)

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



## 5.4 PROJECCIÓ I TRANSFORMACIÓ GEOMÈTRICA (MODEL FINAL TOPOEMPÍRIC)

*De  $R^{11}$  a  $R^3$ : traduir dades multidemnsionals a una representació geomètrica 3D (+temps)*

Cada variable empírica és una dimensió pròpria, això vol dir que les dades viuen a dins d'un espai de dimensió  $R^{11}$  on cada mostra (observació) és un punt  $x=(x_1, x_2, \dots, x_{11})$ , però cap coordenada d'aquest espai és espacial ni causal, és un espai empíric cru. Tanmateix el model geomètric introduceix una estructura que redueix les 11 dimencions a 3, formalment és  $f: R^{11} \rightarrow R^3$  per mitjà d'una immersió topològica o immersió geomètrica causal a fi de fer visible les relacions estructurals (hom entén per topològica les coordenades aspcionals derivades de les tres dimensions que extreuen tota la informació causal del DAG 2D). Dins d'un model topològic, els nodes s'organitzen per nivells i graus; en canvi, en el registre empíric hi circula un volum d'informació crua que recorre la mateixa estructura geomètrica, on no hi ha directament les observacions empíriques, sinó un espai de propietats derivades; cada punt a 3D és ara un node (una variable empírica) i la seves coordenades ( $X, Y, Z$ ) provenen de la seva estructura causal (grau d'entrada/dependència, grau de sortida/efecte i nivell/jerarquia de flux causal).

# CADENA DE TRANSFORMACIONS GEOMÈTRIQUES A PARTIR DE LES DADES

Hom afaiçona les coordenades topològiques (les que provenen del pathway sense cicles) segons les dades, és a dir, hom reubica els nodes a partir d'estadístics que resumeixen la informació continguda en les dades: d'aquí que hom aprofiti mesures de tendència central, de variabilitat, de correlació...

## *Normalització de la distribució espacial causal topològica (escalatge de coordenades)*

Perquè els tres eixos (X, Y, Z) representin magnituds comparables de variació causal; és a dir, l'objectiu és que el rang de valors de cada dimensió axial sigui comparable en termes de magnitud amb la resta. Posar X, Y i Z en la mateixa escala numèrica a fi d'evitar biaixos d'unitat de mesura, totes tres dimensions aporten un pes geomètric comparable.

Per exemple, si el nivell jeràrquic Z domina, el model deixa d'esser un espai isòtrop de causalitat. Cal fer que les dependències entre variables-node siguin comparables en magnitud geomètrica, si les coordenades Z no són escalades respecte de X i Y, llavors les variacions verticals entre nivells tenen molt més de pes geomètric que no pas les desviacions retrolaterals dins d'un mateix nivell i el flux resultant tendeixi sempre més cap a Z encara que les dades no ho justifiquin.

Se'n diu anisotropia geomètrica: un espai on una direcció (Z) té un pes més gran que les altres; d'aquí que calgui un espai isòtrop de causalitat, un espai geomètric on totes les direccions causals, siguin verticals o retrolaterals són comparables en magnitud i contribueixen de manera equilibrada al flux resultant. Hom entén com aquell en què cap eix (X, Y o Z) domina el comportament geomètric del model. És evident que a nivells intermigts hi ha molta més presència de connexions retrolaterals que no pas a d'altres nivells, per exemple, al nivell de nodes-arrel on hi ha sempre grau\_IN zero.

Sense normalització, una diferència d'1 unitat en Z pot equivaldre a 100 en X o Y (si les escales són molt diferents) i això fa que qualsevol distància sigui del tot sotmesa per Z, cal que totes tres tinguin el mateix ordre de magnitud (comparteixin la mateixa escala). Per tant, així, les distàncies euclidianes siguin comesurables, és a dir, que reflecteixen relacions reals sense distorsions d'escala. Ara el centre geomètric (0,0,0) és el punt de referència del flux causal global (després el centre de masses serà els valors mitjans de cada eix a fi de treballar sols amb escales positives) i les variacions de cada eix tenen la mateixa unitat de mesura.

## *Normalització de la distribució espacial causal empírica: estadístics per tal de recollir el màxim d'informació empírica de cada variable de les dades i transmetre-la a dins de l'espai geomètric*

Cada coordenada dels nodes del model geomètric 3D és una transformació geomètrica d'un subespai empíric multidimensional (no una única variable), en la qual hom pot mesurar direcció, angles, fluxos... Cal traduir les mesures empíriques de cada variable en termes geomètrics, cosa que implica sintetitzar les dades crues a informació estructural. Per exemple, és el mateix procediment que fan els físics: traduir dades multidimensionals a una representació geomètrica 3D (+temps), cosa que implica una metrització, reduir un espai de moltes de dimensions a un espai 3D amb propietats mètriques que mantenen les relacions causals i dinàmiques.

Amb les dades empíriques no cal fer l'escalatge (centrar i tipificar) per a cada columna-variable tal com és fet abans amb les coordenades topològiques, perquè el rang de valors és molt semblant entre totes onze variables. Havent fet l'escalatge dels eixos del model topològic pur, la normalització

de la distribució espacial causal empírica dels nodes és definida segons una cadena d'operacions de transformació geomètrica. Hom aplica aquesta cadena (hom la podria formalitzar com un producte matricial, on cada matriu és una transformació espacial de les coordenades prèvies).

Aquesta cadena d'operacions pot esser tan llarga i elaborada com hom vulgui: aquí hom opta per dues eines bàsiques de l'estadística per a dur a terme dues transformacions a fi d'associar les mesures estadístiques de mitjana i desviació tipus a la posició geomètrica relativa dels nodes i associar la correlació de les variables a la llargària de les arestes del model geomètric.

## **==>NORMALITZACIÓ RADIAL: PRIMERA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI. Mitjana/desviació tipus global**

## **==>NORMALITZACIÓ INDIVIDUAL: SEGONA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI. Correlació**

(per mitjà de l'ajustament de la llargària d'aresta per aresta)

A partir de la Matriu de correlació d'arestes, hom treballa amb dataframes perquè cal identificar cada aresta entre nodes (a més a més de les coordenades de cada node) i això és més fàcil de fer directament (l'objectiu ara no és automatitzar el procés sinó calcular les noves i difinitives coordenades un cop feta la segona transformació geomètrica)

$$M_{corr\_arestes} = M_{correlacions} \cdot M_{adjacent\ DFrefós} = DF_{corr\_arestes} + DF_{coords\_topoemp1}$$

La correlació entre variables, un refinament més de caràcter local, és associada per a cada aresta individual, perquè hom tradueix en termes geomètrics la correlació com a longitud del mòdul (mòdul més llarg, més de força o d'associació), cosa que intervé en cada aresta de forma individual. Aquesta segona associació de la correlació amb les arestes sols és possible per als nodes que són lligats per una aresta; la correlació és com un factor d'allargament o escurçament de les arestes individuals. Per exemple, la longitud de les arestes (d'on es deriven els mòduls, atès que el mòdul és una expressió de força causal efectiva) pugui esser entesa segons la informació empírica a fi que el flux causal reflecteixi la força real de transmissió observada i no sols la distància topològica empírica a priori.

La segona metrització diagonal reescala, doncs, localment i de manera independenta cada aresta, sense alterar la jerarquia causal ni la connexió estructural entre nodes. En canvi, una metrització plena (no diagonal) pot descriure dependències creuades entre nodes sense aresta, donant lloc a un espai causal anisotòpic. Tanmateix el model aquí presentat empra una versió filtrada, en què els valors empírics sols s'apliquen a les arestes existents al model geomètric topològic, cosa que garanteix que les correccions empíriques deformen localment el camp causal.

Així, la primera transformació implica una deformació en la radialitat de cada node, és a dir, totes les arestes (no importa si l'aresta és in o out) en són afectades; en canvi, cada valor de la segona transformació afecta una única aresta entre dos nodes adjacents (l'origen és fix, s'allarga o s'escurça en el sentit de l'aresta).

Així, totes dues transformacions geomètriques defineixen la variabilitat local geomètrica topològica com la suma de la variabilitat radial d'un node i la variabilitat internodal de cada aresta; i la variabilitat total com la suma de totes les radials i les seves respectives internodals.

Totes dues transformacions consecutives vers les coordenades topològiques escalades tenen com a punt de referència un centre de masses (vector de mitjanes de les coordenades calculat després d'haver fetes les transformacions) que mostra com la xarxa de punts s'expandeix o es contreu respecte d'aquest centre. En aquest estudi, una gran part dels nodes s'expandeix, també hi han acostaments al centre de masses però són molt tímids, és clar que hi ha una expansió, un

allunyament del centre de masses i els nodes guanyen pes i visibilitat. Cal afegir que hi han moltes de possibilitats perquè la cadena d'operacions pugui esprémer de forma òptima tota la informació de les dades i traduir-la en termes geomètrics. Hom pot trobar la traducció en termes geomètrics de les dades empíriques a angles, a superfícies... El disseny de la metrització podria aprofitar els signes de cada transformació a fi de fer una lectura més sofisticada de la configuració final de les coordenades topoempíriques finals. També, si hi han dades d'altres assaigs que fossin compatibles, hom podria aprofitar-les per seguir amb la cadena d'operacions, aprofitar tota la informació numèrica que hi ha a l'abast.

## *Normalització vectorial: no hi és inclosa*

transforma cada vector en un vector unitari, cosa que sembla incoherent, sols justificable si els angles són a prop de 0 o de 90 graus (segons una geometria non-euclidiana, vegeu Annex1), els quals poden esser problemàtics per al càlcul d'angles, cosa que hom pot resoldre fent una anàlisi prèvia dels angles continguts al model i establir un valor determinat si n'hi ha cap, angle de risc. La normalització unitària —tot i que no distorsiona els angles entre vectors— no sembla adient perquè elimina tota la informació de força causal (el mòdul real). Un vector unitari no conté ni profunditat jeràrquica ni intensitat causal, sols és en essència direccional, sols dóna orientació.

## *LA METÀFORA BAYESIANA: empírica estructural a priori i empírica quantitativa a posteriori*

la informació topològica juga el mateix paper que una prèvia (prior) en inferència bayesiana, és a dir, el coneixement causal previ, allò que sabem del sistema abans de registrar les dades específiques; és empíric (prové d'assaigs previs o de la literatura biològica), però no és pas numèric (no té magnituds o intensitats assignades), és una estructura empírica non-quantitativa. Les dades de l'estudi són l'evidència observada (l'equivalent a la versemblança o likelihood) perquè contenen magnituds reals (nivells d'expressió) que poden refermar, refinjar o contradir parcialment la distribució espacial topològica prèvia. A la fi, el procés geomètric, aquí anomenat geomètric causal o espai topològic-empíric, és una posteriorització. El model estima una distribució espacial a posteriori de coherència geomètrica entre la topològica i l'empírica.

→EMPÍRICA ESTRUCTURAL (de natura qualitativa o relacional): aquesta és la que va del pathway al DAG 2D o al model topològic 3D. La informació a priori és el que Pearl anomena “història causal” (“background causal knowledge”), és una informació estructural prèvia, que no és numèrica però oisí empírica, atès que prové de l'experiència científica acumulada, de l'observació en diversos assaigs i de coneixement del mecanisme biològic. Hom parteix d'un coneixement empíric de base, és informació empírica non-numèrica però sí factual.

→EMPÍRICA QUANTITATIVA (de natura numèrica o observacional): les dades afaiçonen les coordenades 3D topològiques en termes d'estructura, de força de flux (la longitud de les arestes), etc.

D'aquí que el núvol de nodes connectats per arestes dirigides és en si mateix significatiu; les mesures de cada angle, mòdul, sentit... són per si mateixes explicatives perquè el flux d'informació correspon a l'eix vertical Z (flux general vertical extret en essència del subgraf vertical), però condicionat per la connectivitat topològica d'un node (graus in i out), perquè és traduïda a dins d'un espai angular local definit pels vectors de flux incident i emergent; aquesta estructura angular determina la divergència o convergència del flux causal i, per tant, la seva distribució a la coherència geomètrica global del sistema.

#### 5.4.1 Model general topoempíric amb la primera transformació geomètrica node-radial de mu/sigma\_global

Normalització espacial de les dades topològiques a partir d'un estadístic que extreu informació del joc de dades de l'estudi de l'article de Sachs et al., les dades train, amb escalatge previ. El darrer salt és passar del model geomètric causal (estructural) al model topoempíric, és a dir, l'estructura a priori (DAG 2D) que recull la informació de les dades reals observades per mitjà d'una cadena de transformacions. Quan hom normalitza, la magnitud de cada dimensió (nivell, in, out) és adaptada a la seva variabilitat observada, talment com fer aprenentatge geomètric de la causalitat. Ara cada node del DAG geomètric representa una variable observada (per exemple, una proteïna, un gen,...) a dins d'un espai causal geomètricament significatiu i cada aresta una relació de dependència.

Aquestes coordenades són transformades a l'espai 3D causal a fi d'obtenir el model geomètric topoempíric: l'espai (X,Y,Z) ja no és una jerarquia de síntesi empírica a priori (proveïda pel pathway) sinó una estructura deformada pel flux causal observat: ara la distància entre nodes reflecteix diferències reals d'influència causal; els angles i longituds amb localGeomTest() són angles i longituds tridimensionals empíricament significatius, comparables amb p\_vàlues reals.

Aquest model topoempíric causal fusiona la topologia causal a priori amb les dades experimentals a posteriori a dins d'un mateix espai geomètric. Per exemple, en biologia estructural, hom descriu la forma d'una proteïna o d'una molècula, però els models biològics són descriptius no pas causals, amb aquest model cada node té una posició geomètrica proporcional al seu pes causal real segons les dades a priori del pathway i les dades de l'estudi en qüestió i cada angle o longitud mesura consistència causal entre fluxos; altrement dit, *hom converteix la xarxa bioquímica en una superfície causal viva, deformable segons la dinàmica observada*. Hom pot comparar l'estructura geomètrica causal d'una via senyalitzadora amb la seva estructura tridimensional molecular; veure quins canvis topològics (en arestes in/out) corresponen a canvis físics (plegament, interacció...); o detectar punts de transició on la geometria causal es trenca (indicadors de canvi funcional, mutacions, inhibicions...)

Normalització espacial per metrització radial: no cal centrar i tipificar perquè tots els nodes es mouen dins d'un rang molt homogeni (entre 0.7 i 4.5). El millor estadístic per captar el màxim d'informació empírica de les dades registrades és el quotient de la mitjana de cada columna per la desviació típica global, és la intensitat radial mitjana ponderada.

```
# hom pesa amb dades empíriques (quotient de la mitjana de cada columna per la desviació típica global)
m_means <- colMeans(dades, na.rm = TRUE)
sigma_global <- sd(unlist(dades), na.rm = TRUE)
dades_diag <- m_means/sigma_global
dades_diag <- dades_diag[rownames(coords3Dn)]
coords3Dn_diag <- coords3Dn*dades_diag # coincideix que el producte per files és correcte: (N*3) * N
coords3Dn_diag <- diag(dades_diag) %*% coords3Dn # el producte de cada fila de la matriu de coordenades per la matriu diagonal cal formalitzar-la així
rownames(coords3Dn_diag) <- rownames(coords3Dn)
coords3Dn_diag

##          X          Y          Z
## Mek    3.56525080 22.34717517 12.79647227
## PIP2   3.35604330 14.04544137 12.04558036
## PIP3   9.09922158  0.05916116  0.05916116
## PKA    21.19079368  0.06911364  0.06911364
## PKC    13.60035355 14.43828175 24.69854441
## Plcg   7.04011000  7.47174833 12.75707306
## Raf    3.68692626 15.43022601 13.23319236
```

```

## Erk  0.06096218 13.25029828 11.36365377
## Akt  0.06337969 13.77575030 11.81428927
## Jnk  0.06490635 14.10757318 12.09886554
## P38  0.06457962 14.03655694 12.03796095

```

No cal centrar i tipificar dades (totes les columnes-node són gairebé a dins d'un mateix rang)

```

apply(dades,2,range)

##          Raf      Mek      Plcg      PIP2      PIP3      Erk      Akt      PKA
## [1,] 1.711091 1.539091 1.236145 1.325193 1.395194 1.092741 1.120213 0.7119922
## [2,] 4.500000 4.500000 4.500000 4.500000 4.500000 4.500000 4.500000 4.500000
##          PKC      P38      Jnk
## [1,] 1.294027 0.6936238 1.64896
## [2,] 4.500000 4.5000000 4.50000

apply(coords3Dn_diag,2,range)

##          X          Y          Z
## [1,] 0.06096218 0.05916116 0.05916116
## [2,] 21.19079368 22.34717517 24.69854441

stopifnot(all(v(g_bosc)$name %in% rownames(coords3Dn_diag)))

open3d()

## wgl
## 19

plot3d(coords3Dn_diag[, c("X","Y","Z")], col = "red", size = 0.2, type = "s")

edges <- as_edgelist(g_bosc, names = TRUE) # torna noms, no indexes
for (i in seq_len(nrow(edges))) {
  p0 <- as.numeric(coords3Dn_diag[edges[i, 1], c("X","Y","Z")]) # origen per NOM
  p1 <- as.numeric(coords3Dn_diag[edges[i, 2], c("X","Y","Z")]) # destí per NOM
  segments3d(rbind(p0, p1), col = "grey60", lwd = 0.9)
}

offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag))
text3d(coords3Dn_diag[, "X"]+0.02,
       coords3Dn_diag[, "Y"],
       coords3Dn_diag[, "Z"]+offset,
       texts = rownames(coords3Dn_diag),
       adj = c(0, 0.2),
       cex=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

```

```

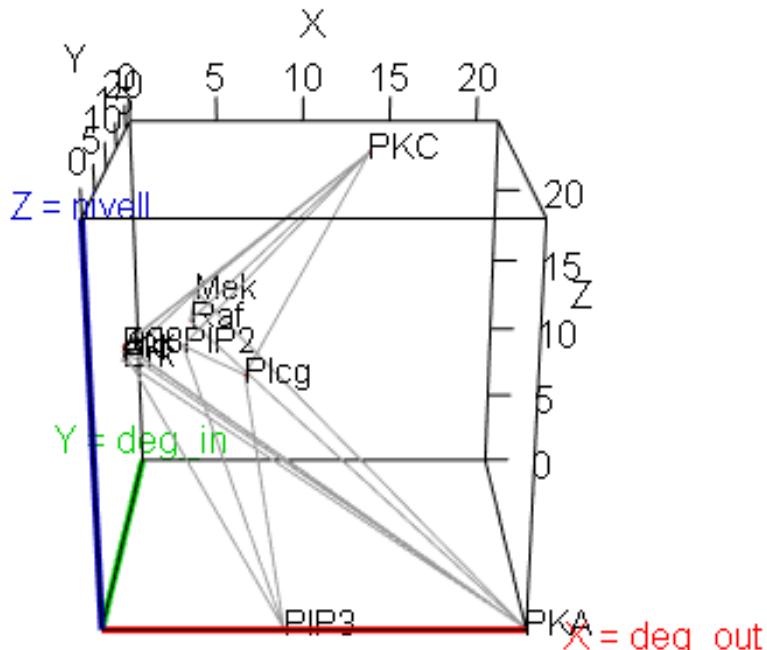
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



#### 5.4.2 Model general topoempíric amb la segona projecció aresta-local de correlacions

Segona transformació geomètrica de la cadena d'operacions: les correlacions entre variables són associades a les arestes entre nodes (sols les arestes que són al model, no pas arestes absents).

```

# matriu d'adjacència (dirigida) seguint l'ordre de coords3D
nodes <- rownames(coords3Dn_diag)
A <- as.matrix(as_adjacency_matrix(g_bosc, type = "both", attr = NULL))

# matriu de correlacions empíriques entre variables
R <- cor(dades[, nodes, drop = FALSE], use = "pairwise.complete.obs", method = "pearson")

# filtrat per estructura: sols on hi ha aresta (element a element)
R_filt <- R * A # zeros on no hi ha aresta

R_filt_df <- as.data.frame(as.table(R_filt))

```

```

colnames(R_filt_df) <- c("des", "fins", "corr")
R_filt_df <- R_filt_df[R_filt_df$corr != 0,]
R_filt_df

##      des   fins      corr
## 4     PKA   Mek  0.048366457
## 5     PKC   Mek  0.068699870
## 7     Raf   Mek  0.732754099
## 14    PIP3  PIP2  0.225912342
## 17    Plcg  PIP2  0.122102384
## 46    PIP2  PKC -0.007702225
## 50    Plcg  PKC  0.011182880
## 58    PIP3  Plcg  0.135894703
## 70    PKA   Raf  0.083669411
## 71    PKC   Raf  0.021805821
## 78    Mek   Erk  0.083634460
## 81    PKA   Erk  0.259596487
## 91    PIP3  Akt -0.002099523
## 92    PKA   Akt  0.226368066
## 103   PKA   Jnk  0.087830114
## 104   PKC   Jnk  0.210998302
## 114   PKA   P38 -0.059613508
## 115   PKC   P38  0.488494531

```

Hom calcula la llargària de cada aresta segons la darrera matriu de coordenades

```

edges <- as_edgelist(g_bosc, names = TRUE)

# funció d'utilitat per a Longituds
edge_lengths <- function(coords3Dn_diag, edges){
  des <- edges[,1]
  fins <- edges[,2]

  p1 <- coords3Dn_diag[des, , drop = FALSE]
  p2 <- coords3Dn_diag[fins, , drop = FALSE]

  vec <- p2 - p1
  lon <- sqrt(rowSums(vec^2))

  data.frame(
    des = des,
    fins = fins,
    dx = vec[,1], # dx dy i dz dibuixen un moviment (ÉS LA DIRECCIÓN I MAGNITUD), no pas
    una coordenada
    dy = vec[,2],
    dz = vec[,3],
    length = lon,
    row.names = NULL
  )
}

edges_df <- edge_lengths(coords3Dn_diag, edges)
#(edges_df <- edges_df[order(edges_df$fins, decreasing = FALSE),])

df_fos <- R_filt_df %>% left_join(edges_df, by=c("des", "fins"))
df_fos$llarg_nova <- (df_fos$corr*df_fos$length) + df_fos$length

```

```

df_fos$prop <- df_fos$llarg_nova / df_fos$length
(df_fos <- df_fos[order(df_fos$llarg_nova, decreasing = TRUE),])

##      des fins      corr        dx        dy        dz    length
## 12   PKA  Erk  0.259596487 -21.1298315 13.1811846 11.2945401 27.345567
## 14   PKA  Akt  0.226368066 -21.1274140 13.7066367 11.7451756 27.788283
## 1   PKA  Mek  0.048366457 -17.6255429 22.2780615 12.7273586 31.128081
## 15   PKA  Jnk  0.087830114 -21.1258873 14.0384595 12.0297519 28.073055
## 9    PKA  Raf  0.083669411 -17.5038674 15.3611124 13.1640787 26.751488
## 18   PKC  P38  0.488494531 -13.5357739 -0.4017248 -12.6605835 18.538310
## 17   PKA  P38 -0.059613508 -21.1262141 13.9674433 11.9688473 28.011778
## 4    PIP3 PIP2  0.225912342 -5.7431783 13.9862802 11.9864192 19.294413
## 16   PKC  Jnk  0.210998302 -13.5354472 -0.3307086 -12.5996789 18.495124
## 13   PIP3  Akt -0.002099523 -9.0358419 13.7165891 11.7551281 20.198374
## 2    PKC  Mek  0.068699870 -10.0351027 7.9088934 -11.9020721 17.461764
## 8    PIP3 Plcg  0.135894703 -2.0591116 7.4125872 12.6979119 14.846661
## 6    PIP2  PKC -0.007702225 10.2443102 0.3928404 12.6529640 16.284892
## 10   PKC  Raf  0.021805821 -9.9134273 0.9919443 -11.4653520 15.189282
## 7    Plcg  PKC  0.011182880  6.5602435 6.9665334 11.9414714 15.302553
## 3    Raf  Mek  0.732754099 -0.1216755 6.9169492 -0.4367201 6.931790
## 11   Mek  Erk  0.083634460 -3.5042886 -9.0968769 -1.4328185 9.853232
## 5    Plcg PIP2  0.122102384 -3.6840667 6.5736930 -0.7114927 7.569149

##      llarg_nova      prop
## 12   34.44438 1.2595965
## 14   34.07866 1.2263681
## 1    32.63364 1.0483665
## 15   30.53871 1.0878301
## 9    28.98977 1.0836694
## 18   27.59417 1.4884945
## 17   26.34190 0.9403865
## 4    23.65326 1.2259123
## 16   22.39756 1.2109983
## 13   20.15597 0.9979005
## 2    18.66138 1.0686999
## 8    16.86424 1.1358947
## 6    16.15946 0.9922978
## 10   15.52050 1.0218058
## 7    15.47368 1.0111829
## 3    12.01109 1.7327541
## 11   10.67730 1.0836345
## 5    8.49336 1.1221024

```

# reajust de la llargària de cada aresta segons la correlació (columna 'prop')

```

# hom itera per cada aresta
for (i in seq_len(nrow(df_fos))) {
  des <- df_fos$des[i]
  fins <- df_fos$fins[i]

  # índexs dels nodes
  v1 <- which(rownames(coords3Dn_diag) == des)
  v2 <- which(rownames(coords3Dn_diag) == fins)

  # posicions originals
  p1 <- coords3Dn_diag[v1, ]
  p2 <- coords3Dn_diag[v2, ]
}
```

```

# vector direccional (de <des> cap a <fins>)
v <- p2 - p1

# Factor d'ajust
f <- df_fos$prop[i]

# Nova posició del node d'arribada, allarga/escurça segons el signe del factor
p2_new <- p1 + v * f

# hom actualitza coordenades del node de destinació ==> SOLS EL NODE DESTINACIÓ
coords3Dn_diag[v2, ] <- p2_new
}

min <- apply(coords3Dn_diag, 2, min)
(coords3Dn_diag_corr1 <- sweep(coords3Dn_diag, 2, min, FUN="-"))

##          X           Y           Z
## Mek    7.997961 29.315418380 11.497698618
## PIP2   7.526047 18.211423207 14.727365584
## PIP3  15.140769  0.000000000  0.000000000
## PKA   27.232341  0.009952487  0.009952487
## PKC   19.628501 14.467310719 24.676170477
## Plcg  12.801835  8.419918510 14.423490868
## Raf    8.017768 16.705511579 14.051138412
## Erk    0.000000 15.550560292 14.465575026
## Akt    1.351367 16.784021406 14.383598584
## Jnk    1.003503 15.471793377 10.660703422
## P38   1.147557 12.960205699  5.449356274

open3d()

## wgl
## 22

plot3d(coords3Dn_diag_corr1,
        col = "red3",
        size = 0.5,
        type = "s")
edges <- df_fos[, c("des", "fins")]

for (i in seq_len(nrow(edges))) {
  v1 <- which(rownames(coords3Dn_diag_corr1) == edges$des[i])
  v2 <- which(rownames(coords3Dn_diag_corr1) == edges$fins[i])
  segments3d(rbind(coords3Dn_diag_corr1[v1, ], coords3Dn_diag_corr1[v2, ]),
             col = "grey60",
             lwd = 0.9)
}

offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr1))
text3d(coords3Dn_diag_corr1[, "X"]+0.02,
       coords3Dn_diag_corr1[, "Y"],
       coords3Dn_diag_corr1[, "Z"]+offset,
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2),
       cex=0.9)

```

```

centre <- colMeans(coords3Dn_diag_corr1) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orange", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orange3", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orange3", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orange3", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

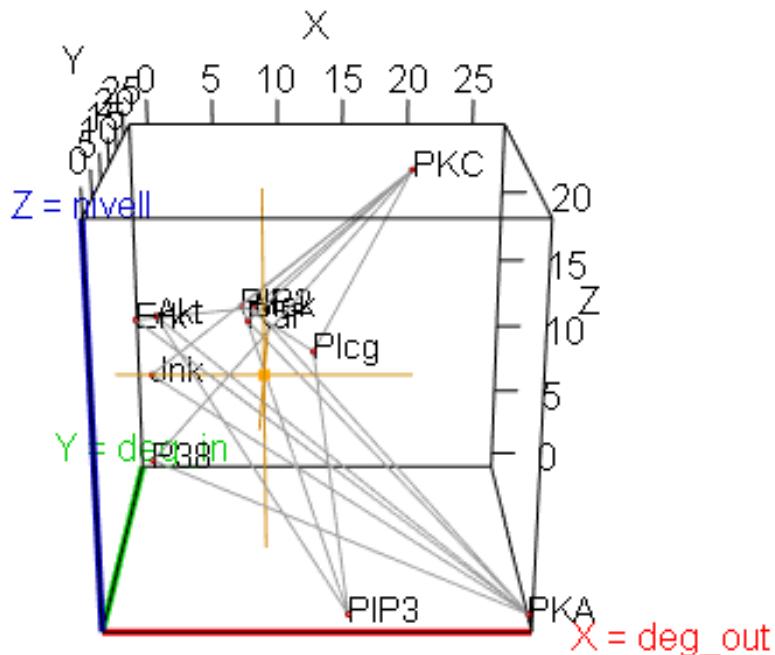
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



## 5.5 FLUX CAUSAL DEL MODEL GENERAL TOPOEMPÍRIC AMB DUES TRANSFORMACIONS GEOMÈTRIQUES: LA MITJANA VECTORIAL GLOBAL

*Introducció*

### A. MITJANA DELS VECTORS DIRECTORS SOLS RESPECTE DE Z

Fer la mitjana vatorial global de tots els vectors directors provinents tant dels nodes fulla pertanyents del subgraf vertical (cadenes de 3 o 2 nodes SENSE retroflux al model geomètric 3D) com dels nodes fulles pertanyents del subgraf retrolateral (col·lididors AMB retroflux al model geomètric 3D).

ANGLE RESPECTE DE Z DE LA MITJANA DELS VECTORS DIRECTORS: calcula una direcció mitjana real (no una mitjana d'angles): el resultat és l'angle entre el vector mitjà resultant i l'eix Z.

hom pot calcular per a tota parella de nodes disjunts la diferència entre els angles de flux AMB I SENSE condicionar de cada subgraf; a més a més, cal tenir en compte la longitud del vector director.

Si hom distingeix entre arestes verticals i laterals, hom s'adona que els graus de sortida i entrada (in/out) són projeccions globals; és a dir, un node pot rebre o treure arestes sense distingir quins són verticals o laterals. Per exemple, un node pot rebre totes les connexions d'entrada laterals llevat d'una de vertical i apareixer amb un grau\_in molt alt, tot i que no sigui gaire funcionalment causal dins del flux vertical principal. Per tant, pot esser interessant distingir entre graus d'in i graus d'out a dins de cada tipus d'arestes.

Al subgraf vertical no hi poden haver col·lididors-fulla per definició d'arbre/bosc, al DAG general podria semblar que hi ha un col·lididor-fulla vertical però no és real si hom accepta les definicions que impliquen la descomposició del DAG general en subgrafts vertical i retrolateral. Tanmateix, sí que cal

tenir en compte els col·líditors-fulla retroverticals (una aresta de cada, retro i vertical, que xoquen al node fulla) del model geomètric 3D per tal de corregir el flux vertical boscoïde segons les branques que tinguin contraflux a l'espai de transició entre els subgrafs vertical i retrolateral (vegeu annex1).

Plot 3D que mostra en groc el retroflux o pseudoretroflux i en blau el flux vertical si hom es basa en la definició d'arbre/bosc

```

deg_in <- degree(g_bosc, mode = "in")
deg_out <- degree(g_bosc, mode = "out")

open3d()

## wgl
## 25

plot3d(coords3Dn_diag_corr1, col = "orangered", size = 1, type = "s")

edges <- as_edgelist(g_bosc)

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges), function(i) {
  e <- paste(edges[i, 1], edges[i, 2], sep = "_")
  if (e %in% edges_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges_lat) {
    return("gold") # connexió retrolateral (retroflux i reguladora)
  }
})

for (i in 1:nrow(edges)) {
  v1 <- which(V(g_bosc)$name == edges[i, 1])
  v2 <- which(V(g_bosc)$name == edges[i, 2])
  segments3d(rbind(coords3Dn_diag_corr1[v1, ], coords3Dn_diag_corr1[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr1))
text3d(coords3Dn_diag_corr1[, "X"]+0.02,
       coords3Dn_diag_corr1[, "Y"],
       coords3Dn_diag_corr1[, "Z"]+1,
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2),
       cex=0.9)

centre <- colMeans(coords3Dn_diag_corr1) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)

```

```

# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

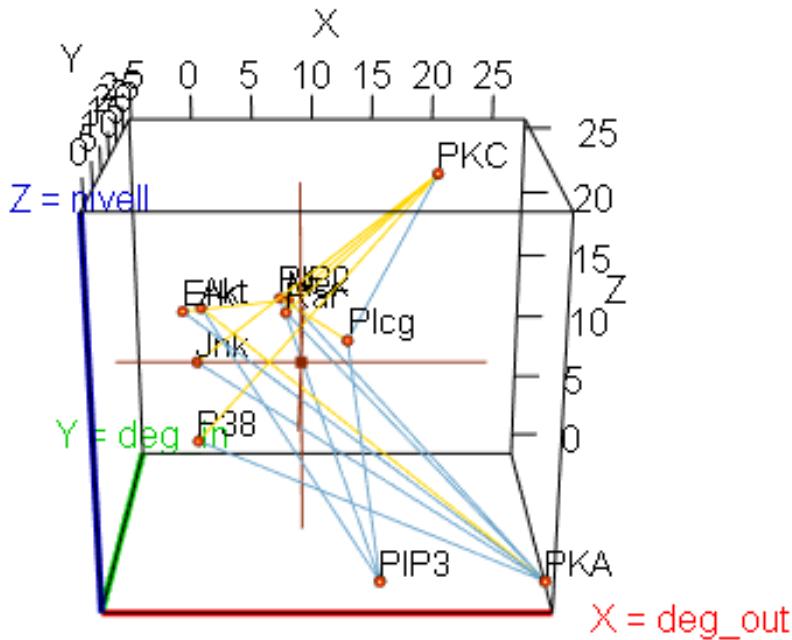
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



**B. MITJANA DELS VECTORS DIRECTORS RESPECTE DE Z, X i Y:** fer la mitjana vectorial global de tots els vectors directors directors provinents de tots els nodes fulla del subgraf retrolateral (siguin verticals, retroverticals o lateroverticals al model geomètric 3D)

La definició de flux de causalitat que hom lliga sols amb l'eix vertical Z (A) és senzilla però amb biaix perquè no és l'òptima, aquí sols treballa l'eix Z vertical. Aquesta no deixa d'esser bona via exploratòria o simplificada, útil en cas d'ambigüïtat. Però de fet n'hi ha una segona, de manera, més holística i robusta (B): el flux causal més potent i robust és tenir en compte alhora: 1. Z: Una vertical MÀXIMA

2. □Y: uns graus\_IN MÀXIMA 3. □X: uns graus\_OUT MÍNIMA

De fet, els nodes qui compleixen amb exactitud aquests requisits d'optimització són els nodes fulla del subgraf retrolateral (que justament són els col·lididors-fulla del model causal geomètric 3D), per tant, el flux òptim en totes tres dimensions és el resultat de calcular la mitjana vectorial global entre tots els vectors directors dels col·lididors-fulla. És la mitjana de les orientacions dels vectors directors a dins de l'esfera de direccions que té com a origen la coordenada mitjana dels nodes arrels.

Amb (B) hom pot llegir el problema causal com a optimització geomètrica, perquè optimitzant els tres eixos hom pot llistar les vies causal prioritàries; fins i tot, hom pot detectar arestes absents al model de treball: si tots dos nodes tenen 0 outs, el mateixos IN però una diferència d'alçada, és probable que hi hagi un node absent entre ells; també la situació que tots dos tinguin 0 outs, el mateixos IN, sense diferència d'alçada però un IN col·lididor vertical i l'altre un IN col·lididor retrovertical, aquesta asimetria juga a favor d'una aresta absent del col·lididor retrovertical cap al vertical. Però hom treballa amb (B) perquè hom obté un flux causal del model general únic en l'esfera de direccions, perquè és l'únic vector resultant de maximitzar Z i Y tot minimitzant alhora X.

## C. D'ALTRES PER ARA REBUTJATS

Hom descarta per ara l'estrategia de fixar el valor de Z —a fi de calcular l'angle entre nodes del mateix nivell vertical, aquí l'angle mesuraria la coherència del flux lateral, com s'alineen o es dispersen els vectors directors individuals a dins d'un mateix estrat vertical—; també és descartada l'estrategia de fer la regressió als nodes o als punts migs de les arestes verticals del model geomètric 3D.

## DUES VIES DE VALIDACIÓ LOCAL DEL MODEL CAUSAL GEOMÈTRIC

Atès que hom parteix de l'evidència empírica de localTest() del DAG en 2D i dels resultats de l'article de Sachs et al., hom assaja estratègies de càlcul del flux causal i del seu subsegüent anàleg geomètric localGeomTest() i comparar quina estratègia de càlcul fa que localTest() i localGeomTest() siguin coherents com una primera aproximació de validació local del mètode causal geomètric.

És per això que hom podria calcular: → o bé l'angle del flux causal holístic sols respecte de Z (un únic angle respecte l'eix Z) i comparar-lo amb l'angle de flux local a fi de comprovar que que aquest darrer convergeix al flux causal general

→ o bé els angles del flux causal holístic respecte de Z i YX (dos angles respecte l'eix Z i el pla YX a l'esfera unitària) i comparar-los amb els angles del flux local en estudi a fi de comprovar que que aquest darrer convergeix al flux causal general

Havent determinat l'angle o angles (segons si el flux és calculat o bé sols respecte Z o bé segons Z i YX), hom formalitza aquesta convergència amb localGeomTest() i en compara els resultats amb localTest() i/o els resultats de l'article de Sachs et al.

Observació: per mitjà de la simulació de dades, hom podria establir l'angle (Z) o angles (Z i XY) òptims del flux causal; hi ha diversos algoritmes emprats en el mètode bayesià que potser també hom els podria adaptar/implementar en aquest mètode causal geomètric.

## 5.6 ASSAIG DE VALIDACIÓ A ESCALA LOCAL A PARTIR DEL TIPUS CAUSAL CADENA DEL MODEL GENERAL TOPOEMPÍRIC (per mitjà de LocalTest() de 2D versus la diferència d'angles dels fluxos locals $\theta_{flux\_SENSE\_CONDICIONAMENT}$ i $\theta_{flux\_AMB\_CONDICIONAMENT}$ )

### 5.6.1 INCLINACIÓ RESPECTE L'EIX Z: FUNCIONS DE CÀLCUL D'ANGLES DELS TIPUS CADENA I COL·LIDIDOR i TESTS D'ASSAIGS

Hom pren com a flux vertical el del subgraf local respecte de l'eix Z (no pas respecte del flux general del model 3D, el qual encara no ha estat calculat).

calcular l'angle del flux non-condicionat

```
theta_flux_vectorial_Z_fisic <- function(graph, coords) {  
  E <- as_edgelist(graph)  
  
  u_bar <- c(0,0,0) #flux total: suma vectorial de fluxos individuals  
  
  for(i in seq_len(nrow(E))) {  
    pA <- as.numeric(coords[E[i,1], c("X","Y","Z")])  
    pB <- as.numeric(coords[E[i,2], c("X","Y","Z")])  
    v <- pB - pA #cada vector  
    u_bar <- u_bar + (pB - pA) #homafegeix la seva contribució de flux a la suma
```

```

total
}

len_u <- sqrt(sum(u_bar^2)) #mòdul
flux_z <- u_bar[3]           #component vertical

if (len_u == 0 ){
  theta_z <- NA_real_ #no hi ha flux
  signe <- 0} else {
  theta_z <- acos( abs(flux_z) / len_u ) * 180/pi # angle respecte z
  signe <- sign(flux_z)
}
return(list(u_bar=u_bar, len_u=len_u, flux_z=flux_z, theta_z = theta_z, signe=signe))
}

environment(theta_flux_vectorial_Z_fisic) <- .GlobalEnv

```

Ara cal calcular l'angle del flux condicionat (el node medial de la cadena esdevé un col·lididor)

```

crea_collididor_artificial <- function(graph, chain_nodes) {
  # cal que sigui un vector amb els 3 noms de la cadena
  stopifnot(length(chain_nodes) == 3)
  A <- chain_nodes[1]
  B <- chain_nodes[2]
  C <- chain_nodes[3]

  # hom crea un subgraf amb les dues arestes entrants a B
  g_col <- make_empty_graph(directed = TRUE) |>
    add_vertices(3, name = c(A, B, C)) |>
    add_edges(c(A, B, C, B)) # A->B i C->B

  return(g_col)
}

environment(crea_collididor_artificial) <- .GlobalEnv

theta_flux_colididor_Z_fisic <- function(g_col, coords3D) {
  z <- c(0, 0, 1) # eix de referència (vertical)

  E <- as_edgelist(g_col) # edges del col·lididor (A->B i C->B)

  stopifnot(nrow(E) == 2) # cal que tingui exactament dues arestes entrants

  # identifica els nodes

  A <- E[1, 1]
  B <- E[1, 2] # node col·lididor (mateix destí per a les dues arestes)

  C <- E[2, 1]
  # coordenades

  pA <- as.numeric(coords3D[A, c("X", "Y", "Z")])
  pB <- as.numeric(coords3D[B, c("X", "Y", "Z")])
  pC <- as.numeric(coords3D[C, c("X", "Y", "Z")])

```

```
v_AB <- pB - pA # vectors entrants reals non unitaris (A -> B, C -> B)
```

```
v_CB <- pB - pC
```

```
len_AB <- sqrt(sum(v_AB^2))  
len_CB <- sqrt(sum(v_CB^2))
```

```
cos_angle <- sum(v_AB * v_CB) / (len_AB * len_CB)  
angle <- acos(cos_angle)
```

```
k <- cos(angle / 2)
```

```
d <- v_AB + v_CB  
d <- d / sqrt(sum(d^2))
```

```
len_r <- (len_AB + len_CB) * k  
r_in <- d * len_r #vector resultant entrant
```

```
flux_z <- r_in[3]
```

```
if (len_r == 0) {  
  theta_z <- NA_real_ #no hi ha flux  
  signe <- 0  
} else {  
  theta_z <- acos(abs(flux_z) / len_r) * 180 / pi# angle respecte z  
  signe <- sign(flux_z)  
  signe <- sign(flux_z)  
}  
  
return(list(  
  r_in = r_in,  
  len_r = len_r,  
  flux_z = flux_z,  
  theta_z = theta_z,  
  signe = signe  
))  
}
```

```
environment(theta_flux_collididor_Z_fisic) <- .GlobalEnv
```

Joc d'assaig per a les funcions prèvies Els resultats prevists són correcte: una cadena del tot vertical dóna un angle de 90 graus

```
library(igraph)
```

```
# coordenades: tots 3 punts a l'eix X, pla XY, Z = 0  
coords3D_test <- data.frame(  
  X = c(0, 1, 2),  
  Y = c(0, 0, 0),
```

```

Z = c(0, 0, 0),
row.names = c("A", "B", "C")
)

# cadena A -> B -> C
g_chain <- graph_from_edgelist(matrix(c("A", "B", "B", "C"), ncol = 2, byrow = TRUE),
directed = TRUE)

res_chain <- theta_flux_vectorial_Z_fisic(g_chain, coords3D_test)
res_chain # esperat: ~ 90º (vector totalment horitzontal)

## $u_bar
## [1] 2 0 0
##
## $len_u
## [1] 2
##
## $flux_z
## [1] 0
##
## $theta_z
## [1] 90
##
## $signe
## [1] 0

# col·lididor A -> B <- C
g_col <- graph_from_edgelist(matrix(c("A", "B", "C", "B"), ncol = 2, byrow = TRUE),
directed = TRUE)

res_col <- theta_flux_collididor_Z_fisic(g_col, coords3D_test)
res_col # aquí és molt probable que surti NaN: r_in = (1,0,0) + (-1,0,0) = (0,0,0)

## $r_in
## [1] NaN NaN NaN
##
## $len_r
## [1] 1.224606e-16
##
## $flux_z
## [1] NaN
##
## $theta_z
## [1] NaN
##
## $signe
## [1] NaN

# Línia bàsicament sobre l'eix X, però el tercer punt lleugerament desplaçat
coords3D_test2 <- data.frame(
  X = c(0, 1, 2),
  Y = c(0, 0, 0),
  Z = c(0, 0, 0.2),
  row.names = c("A", "B", "C")
)

```

```

# cadena A -> B -> C (gairebé horizontal)
g_chain2 <- graph_from_edgelist(matrix(c("A", "B", "B", "C"), ncol = 2, byrow = TRUE),
directed = TRUE)

res_chain2 <- theta_flux_vectorial_Z_fisic(g_chain2, coords3D_test2)
res_chain2 # ha de sortir un angle proper a 90º però no exactament

## $u_bar
## [1] 2.0 0.0 0.2
##
## $len_u
## [1] 2.009975
##
## $flux_z
## [1] 0.2
##
## $theta_z
## [1] 84.28941
##
## $signe
## [1] 1

# col·lididor A -> B <- C amb les mateixes coords
g_col2 <- graph_from_edgelist(matrix(c("A", "B", "C", "B"), ncol = 2, byrow = TRUE),
directed = TRUE)

res_col2 <- theta_flux_collididor_Z_fisic(g_col2, coords3D_test2)
res_col2 # físic dóna θ (l'oposat: avall), que vol dir que l'angle respecte l'eix Z,
per tant angle de 90 graus i signe negatiu és cap avall , és a dir, de retroflux

## $r_in
## [1] 0.0000000 0.0000000 -0.1990267
##
## $len_r
## [1] 0.1990267
##
## $flux_z
## [1] -0.1990267
##
## $theta_z
## [1] 0
##
## $signe
## [1] -1

```

Ara un joc de tests per als mòduls sortints dels vectors directors (lineal i físic donarien els mateixos resultats perquè son dades perfectament simètriques)

```

coords3D_test <- data.frame(
  X = c(0, 1, 2),
  Y = c(0, 0, 0),
  Z = c(0, 0, 0),
  row.names = c("A", "B", "C")
)

```

```

coords3D_test2 <- data.frame(
  X = c(0, 1, 2),
  Y = c(0, 0, 0),
  Z = c(0, 0, 2),
  row.names = c("A", "B", "C")
)

res_chain <- theta_flux_vectorial_Z_fisic(g_chain, coords3D_test)

# mòdul del vector resultant calculat per la funció
sqrt(sum(res_chain$u_bar^2))

## [1] 2

# mòdul real entre A i C
sqrt(sum((coords3D_test["C",] - coords3D_test["A",])^2))

## [1] 2

res_chain2 <- theta_flux_vectorial_Z_fisic(g_chain2, coords3D_test2)

sqrt(sum(res_chain2$u_bar^2))

## [1] 2.828427

sqrt(sum((coords3D_test2["C",] - coords3D_test2["A",])^2))

## [1] 2.828427

res_col2 <- theta_flux_collididor_Z_fisic(g_col2, coords3D_test2)

sqrt(sum(res_col2$r_in^2))      # mòdul del vector resultant entrant

## [1] 1.701302

# mòdul teòric
v_AB <- as.numeric(coords3D_test2["B",] - coords3D_test2["A",])
v_CB <- as.numeric(coords3D_test2["B",] - coords3D_test2["C",])
sqrt(sum((v_AB + v_CB)^2))

## [1] 2

# Longitud real del camí

longitud_cami_fisic <- function(graf, coord){
edges <- as_edgelist(graf)
total <- 0

for(i in 1:nrow(edges)){
p1 <- as.numeric(coord[edges[i,1], c("X", "Y", "Z")])
p2 <- as.numeric(coord[edges[i,2], c("X", "Y", "Z")])

v <- p2-p1
total <- total+sqrt(sum(v^2)) #vector de desplaçament + mòdul del vector
}
return(total)
}

```

```
environment(longitud_cami_fisic) <- .GlobalEnv
```

## 5.6.1 INCLINACIÓ RESPECTE L'EIX XY: FUNCIONS DE CÀLCUL D'ANGLES DELS TIPUS CADENA I COL·LIDIDOR

Amb les funcions definides, si l'angle theta és 0 és paral·lel a l'eix Z, és a dir, és un vector del tot vertical; si surt 90, llavors és del tot horitzontal (Z és 0). Ara bé, si el que interessa veure és l'elevació per damunt del pla base XY, llavord cal restar a theta 90 graus, d'aquesta manera es posa el focus no pas en la inclinació respecte l'eix Z vertical (que pot esser contraintuitiva) sinó en l'elevació respecte el pla XY (que és molt més intuïtiva respecte el plot3D).

```
theta_flux_vectorial_XY_fisic <- function(graph, coords) {  
  
  E <- as_edgelist(graph)  
  
  u_bar <- c(0,0,0) #flux total: suma vectorial de fluxos individuals  
  
  for(i in seq_len(nrow(E))) {  
    pA <- as.numeric(coords[E[i,1], c("X","Y","Z")])  
    pB <- as.numeric(coords[E[i,2], c("X","Y","Z")])  
    v <- pB -pA #cada vector  
    u_bar <- u_bar + (pB - pA) #hom afegeix la seva contribució de flux a la suma  
  }  
  
  len_u <- sqrt(sum(u_bar^2)) #mòdul  
  flux_z <- u_bar[3] #component vertical  
  
  if (len_u == 0){  
    theta_z <- NA_real_ #no hi ha flux  
    signe <- 0} else {  
    theta_z <- acos( abs(flux_z) / len_u ) * 180/pi # angle respecte z  
    theta_xy <- 90 - theta_z  
    signe <- sign(flux_z)}  
  }  
  return(list(u_bar=u_bar, len_u=len_u, flux_z=flux_z, theta_xy = theta_xy,  
signe=signe))  
}  
  
environment(theta_flux_vectorial_XY_fisic) <- .GlobalEnv
```

Ara cal calcular l'angle del flux condicionat (el node medial de la cadena esdevé un col·lididor)

```
crea_collididor_artificial <- function(graph, chain_nodes) {  
  # cal que sigui un vector amb els 3 noms de la cadena  
  stopifnot(length(chain_nodes) == 3)  
  A <- chain_nodes[1]  
  B <- chain_nodes[2]  
  C <- chain_nodes[3]  
  
  # hom crea un subgraf amb les dues arestes entrants a B  
  g_col <- make_empty_graph(directed = TRUE) |>  
  add_vertices(3, name = c(A, B, C)) |>  
  add_edges(c(A, B, C, B)) # A->B i C->B
```

```

    return(g_col)
}

environment(crea_collididor_artificial) <- .GlobalEnv

theta_flux_collididor_XY_fisic <- function(g_col, coords3Dn_diag_corr1) {
  # eix de referència (vertical)
  z <- c(0, 0, 1)

  # edges del col·lididor (A->B i C->B)
  E <- as_edgelist(g_col)
  stopifnot(nrow(E) == 2) # cal que tingui exactament dues arestes entrants

  # identifica els nodes
  A <- E[1, 1]
  C <- E[2, 1]
  B <- E[1, 2] # node col·lididor (mateix destí per a les dues arestes)

  # coordenades
  pA <- as.numeric(coords3Dn_diag_corr1[A, c("X", "Y", "Z")])
  pB <- as.numeric(coords3Dn_diag_corr1[B, c("X", "Y", "Z")])
  pC <- as.numeric(coords3Dn_diag_corr1[C, c("X", "Y", "Z")])

  # vectors entrants reals non unitaris (A -> B, C -> B)
  v_AB <- pB - pA
  v_CB <- pB - pC

  r_in <- v_AB + v_CB #vector resultant entrant
  len_r <- sqrt(sum(r_in^2))
  flux_z <- r_in[3]

  if (len_r == 0 ){
    theta_z <- NA_real_ #no hi ha flux
    signe <- 0} else {
    theta_z <- acos( abs(flux_z) / len_r ) * 180/pi # angle respecte z
    theta_xy <- 90-theta_z
    signe <- sign(flux_z)
  }

  return(list(r_in=r_in, len_r=len_r, flux_z=flux_z, theta_xy = theta_xy, signe=signe))
}

environment(theta_flux_collididor_XY_fisic) <- .GlobalEnv

```

## 5.6.2 SUBGRAF AïLLAT: CADENA VERTICAL DE N=3 NODES AMB TEST NEGATIU del model geomètric topoempíric

Hom agafa com a mostra una cadena vertical del DAG 2D, dos nodes non\_adjacents; després s'hi aplica el test d'independència condicionada per mitjà de la funció de dagitty d'R localTest() i hom compara el resultat del test amb el comportament del model geomètric.

Pel que fa al test, aquest resol en contra de la independència entre nodes en aquesta cadena si hom condiciona pel medial quan teòricament n'haurien d'esser, d'independents, perquè el flux que els uneix és tallat (de fet el test encara és més categòric perquè condiciona per tots dos nodes medials

entre PIP3 i PKC, tant per Plcg com per PIP2). Encara que hom condiciona pels medials, no hi ha independència, situació que es tradueix geomètricament en dos vectors directors (amb o sense condicionament) i els seus respectius angles respecte l'eix vertical Z.

Sense condicionament el sentit de les arestes segueixen dibuixant una cadena, el vector director de la qual té l'origen a PIP3 i la terminació en PKC amb un angle respecte el pla YX de 58.46. Però amb condicionament, el sentit de les arestes canvia, ara totes les arestes van cap al node medial Plcg i dibuixen un col·lididor com a origen del flux, el cap terminal avança més o menys cap a X=0 i Y=15 en direcció Y ascendente amb un ample respecte el pla YX de 23.77 graus.

hom assaja de comparar l'angle del flux local amb condicionament versus un flux general entès com a vector director de la cadena, comparar la diferència d'angles sols respecte l'eix vertical Z perquè tots dos angles són calculats sols respecte d'aquest eix. Així, doncs, la diferència entre el flux sense condicionament i el flux amb condicionament és de  $58.46 - 23.77 = 34.69$  graus.

Ob: cal restar-hi 90 graus si hom vol l'angle respecte el pla XY (aquest no és contraintuïtiu com sí que n'és l'angle respecte l'eix Y)

Hom agafa una cadena vertical com a subgraf del model, dibuixada de forma ideal com una recta a dins de 2D, però de fet al graf següent a dins de 3D, hom ja no pot analitzar la recta sols com a objecte en si mateix, autoreferencial, sinó que a dins de l'espai caldrà distingir quina mena de cadena (vertical, lateral, retrògada...)

```

cadena3n_vertical <- subgraph_from_edges(g_bosc_vertical, eids = c(4,11))

arrels1 <- V(cadena3n_vertical)$name[degree(cadena3n_vertical, mode="in") == 0]
fulles1 <- V(cadena3n_vertical)$name[degree(cadena3n_vertical, mode="out") == 0]
noms_nodes1 <- V(cadena3n_vertical)$name

V(cadena3n_vertical)$role <- ifelse(!is.na(match(noms_nodes1, arrels1)), "arrel",
                                         ifelse(!is.na(match(noms_nodes1, fulles1)), "fulla", "mitjancer"))

V(cadena3n_vertical)$color <- ifelse(V(cadena3n_vertical)$role == "arrel", "green3",
                                         ifelse(V(cadena3n_vertical)$role == "fulla",
                                                "deepskyblue", "gold"))

par(mar=c(0,0,1,0))
plot(cadena3n_vertical,
      layout=layout_as_tree(cadena3n_vertical, , root = arrels1),
      edge.arrow.size=0.4,
      vertex.size=15,
      main ="subgraf cadena vertical de 3 nodes")

```

## subgraf cadena vertical de 3 nodes



```
nodes_cadena <- c("PIP3", "Plcg", "PKC")
cadena3n <- induced_subgraph(g_bosc, vids = nodes_cadena)

(theta_cadena3n_Z <- theta_flux_vectorial_Z_fisic(cadena3n, coords3Dn_diag_corr1))

## $u_bar
## [1] 4.487732 14.467311 24.676170
##
## $len_u
## [1] 28.95438
##
## $flux_z
## [1] 24.67617
##
## $theta_z
## [1] 31.54352
##
## $signe
## [1] 1

cat("Angle del flux vertical de la cadena PIP3->Plcg->PKC:",
round(as.numeric(theta_cadena3n_Z$theta_z[1]), 2), "graus\n")

## Angle del flux vertical de la cadena PIP3->Plcg->PKC: 31.54 graus

collididor_artificial <- crea_collididor_artificial(g_bosc, c("PIP3", "Plcg", "PKC"))

(theta_col_Plcg_Z <- theta_flux_collididor_Z_fisic(collididor_artificial,
coords3Dn_diag_corr1))

## $r_in
## [1] -8.779050 2.272467 3.994911
##
```

```

## $len_r
## [1] 9.909346
##
## $flux_z
## [1] 3.994911
##
## $theta_z
## [1] 66.22501
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (collididor Plcg) respecte a Z: %.2f\n",
            theta_col_Plcg_Z$theta_z))

## Angle del flux condicionat (collididor Plcg) respecte a Z: 66.23

```

A theta\_col\_Plcg\_XY i a theta\_col\_Plcg\_XY, hom hi resta 90 graus a fi d'obtenir l'angle respecte el pla XY (aquest no és contraintuïtiu com sí que n'és l'angle respecte l'eix Y)

```

nodes_cadena <- c("PIP3", "Plcg", "PKC")
cadena3n <- induced_subgraph(g_bosc, vids = nodes_cadena)

(theta_cadena3n_XY <- theta_flux_vectorial_XY_fisic(cadena3n, coords3Dn_diag_corr1))

## $u_bar
## [1] 4.487732 14.467311 24.676170
##
## $len_u
## [1] 28.95438
##
## $flux_z
## [1] 24.67617
##
## $theta_xy
## [1] 58.45648
##
## $signe
## [1] 1

cat("Angle del flux vertical de la cadena PIP3->Plcg->PKC:",
    round(as.numeric(theta_cadena3n_XY$theta_xy), 2), "graus\n")

## Angle del flux vertical de la cadena PIP3->Plcg->PKC: 58.46 graus

collididor_artificial <- crea_collididor_artificial(g_bosc, c("PIP3", "Plcg", "PKC"))

(theta_col_Plcg_XY <- theta_flux_collididor_XY_fisic(collididor_artificial,
coords3Dn_diag_corr1))

## $r_in
## [1] -9.165600 2.372526 4.170811
##
## $len_r
## [1] 10.34566
##
## $flux_z

```

```

## [1] 4.170811
##
## $theta_xy
## [1] 23.77499
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (collididor Plcg) respecte a Z: %.2f\n",
            theta_col_Plcg_XY$theta_xy))

## Angle del flux condicionat (collididor Plcg) respecte a Z: 23.77

open3d()

## wgl
## 28

#aspect3d(1,1,1)

coords_CADENA <- coords3Dn_diag_corr1[c("PIP3", "Plcg", "PKC"), c("X", "Y", "Z")]

plot3d(coords_CADENA, col = "gold", size = 1, type = "s", box = TRUE)

# arestes direccionals
edges <- as_edgelist(cadena3n)
for (i in seq_len(nrow(edges))) {
  p0 <- coords_CADENA[edges[i, 1], ] # origen pel nom
  p1 <- coords_CADENA[edges[i, 2], ] # destí pel nom

  arrow3d(
    p0 = as.numeric(p0),
    p1 = as.numeric(p1),
    type = "rotation",
    col = "grey40",
    lwd = 2,
    width = 0.2,
    barblen = 0.025,
    n = 4
  )
}

text3d(coords_CADENA[,1] + 0.02,
       coords_CADENA[,2],
       coords_CADENA[,3] + 0.04,
       texts = rownames(coords_CADENA),
       adj = c(0,0.5))

#
# Longitud real del camí

len_flux <- longitud_cami_fisic(cadena3n, coords3Dn_diag_corr1)

#
# cal que la direcció sigui ascendent

```

```

# if (u_bar[3] < 0) u_bar <- -u_bar (és el flux global)
# origen i extrem de la segeta
res_flux <- theta_flux_vectorial_Z_fisic(cadena3n, coords3Dn_diag_corr1)
u_bar <- as.numeric(res_flux$u_bar)
len_flux <- sqrt(sum(u_bar^2)) # mòdul del vector resultant
#u_dir <- u_bar / sqrt(sum(u_bar^2)) # és una unitat de direcció , no afecta la distància
del mòdul del vector director
u_dir <- u_bar / len_flux # encara unitari

p0_flux <- coords_CADENA["PIP3",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux * u_dir

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "skyblue",
  type = "rotation",
  lwd = 3,
  width = 0.1,
  barrlen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] + 3,
       texts = "Flux causal local: 58.46 respecte al pla XY", color = "black")

# _____
bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

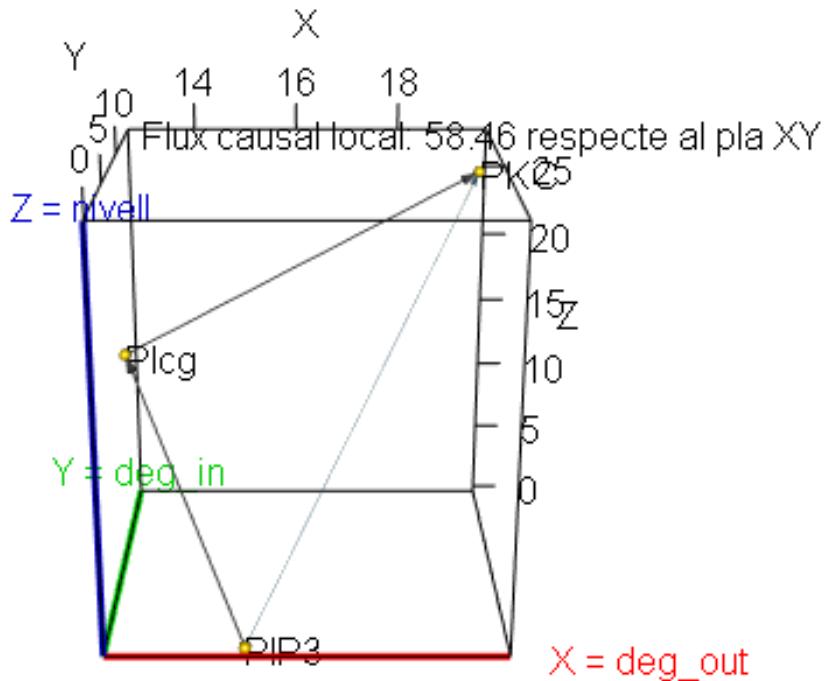
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



Al localTest() del DAG 2D, PIP3 és independent de PKC si hom condiciona per Plcg: si hom fixa el valor de Plcg (la variable mitjancera jo no transmet canvi), llavors PIP3 i PKC deixen de tenir dependència estadística. Plcg bloca el flux d'informació causal entre PIP3 i PKC

Al model geomètric 3D, CONDICIONAR VOL DIR QUE LES ARESTES VAGIN CAP AL NODE CONDICIONAT, SIGUI UN CADENA O UNA FORÇA, és a dir, tots els del node condicionat esdevenen . Ara bé, SI CONDICIONES PER UN COL·LIDIDOR, aleshores un esdevé .

Si hom condiciona pel node medial, llavors el flux es trenca i el vector de Plcg→PKC esdevé Plcg<→PKC

```

coords_CADENA <- coords3Dn_diag_corr1[c("PIP3", "Plcg", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux <- longitud_cami_fisic(cadena3n, coords3Dn_diag_corr1)

# SEGETA BLAVA
# origen i extrem de la segeta
res_flux <- theta_flux_vectorial_Z_fisic(cadena3n, coords3Dn_diag_corr1)
u_bar <- as.numeric(res_flux$u_bar)
#u_dir <- u_bar / sqrt(sum(u_bar^2))
len_flux_u <- sqrt(sum(u_bar^2)) # mòdul del vector resultant
u_dir <- u_bar / len_flux_u      # encara unitari

p0_flux <- coords_CADENA["PIP3",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux_u * u_dir

#

```

```

#SEGETA VERMELLA
res_cond <- theta_flux_collididor_Z_fisic(
  crea_collididor_artificial(g_bosc_vertical, c("PIP3", "Plcg", "PKC")),
  coords3Dn_diag_corr1)

u_cond <- res_cond$r_in # vector direccional condicionat: r_out <- r_in era correcte per
tal de calcular l'angle (perquè volem l'efecte reacció), però per dibuixar la direcció
del flux condicionat, el vector que hom vol mostrar és r_in, no r_out
len_flux_cond <- sqrt(sum(u_cond^2)) # mòdul real del flux condicionat (no del vector
unitari!)
u_dir_cond <- u_cond / len_flux_cond # vector unitari de direcció

p0_cond <- coords_CADENA["Plcg", c("X","Y","Z")] # origen del flux condicionat
p1_cond <- p0_cond + len_flux_cond * u_dir_cond # punt final del flux

#
# Mòduls reals dibuixats al plot
mod_blau_plot <- sqrt(sum((p1_flux - p0_flux)^2))
mod_vermell_plot <- sqrt(sum((p1_cond - p0_cond)^2))

cat("Mòdul blau real (plot) : ", mod_blau_plot, "\n")
## Mòdul blau real (plot) : 28.95438
cat("Mòdul vermell real (plot) : ", mod_vermell_plot, "\n")
## Mòdul vermell real (plot) : 9.909346

# bloc de codi fins a [3] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords <- rbind(coords_CADENA,
                      p0_flux, p1_flux, p0_cond, p1_cond )
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset <- apply(all_coords, 2, min)

# hom aplica el desplaçament
coords_CADENA <- sweep(coords_CADENA, 2, offset, "-")
p0_flux <- p0_flux - offset
p1_flux <- p1_flux - offset
p0_cond <- p0_cond - offset
p1_cond <- p1_cond - offset
# [3]

clear3d()
close3d()
open3d()

## wgl
## 31

```

```

# dibuixa els punts
plot3d(coords_CADENA, col = "gold", size = 1, type = "s", box = TRUE)

# arestes direccionals
edges <- as_edgelist(cadena3n)

cap1 <- coords_CADENA[edges[1, 1], ] # origen pel nom
cua1 <- coords_CADENA[edges[1, 2], ] # destí pel nom

cap2 <- coords_CADENA[edges[2, 2], ] # origen pel nom
cua2 <- coords_CADENA[edges[2, 1], ] # destí pel nom

arrow3d(
  p0 = as.numeric(cap1),
  p1 = as.numeric(cua1),
  type = "rotation",
  col = "grey40",
  lwd = 2,
  width = 0.2,
  barblen = 0.025,
  n = 4
)

arrow3d(
  p0 = as.numeric(cap2),
  p1 = as.numeric(cua2),
  type = "rotation",
  col = "grey40",
  lwd = 2,
  width = 0.2,
  barblen = 0.025,
  n = 4
)

text3d(coords_CADENA[,1] + 0.02,
       coords_CADENA[,2],
       coords_CADENA[,3] + 0.04,
       texts = rownames(coords_CADENA),
       adj = c(0,0.5))

# _____

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "blue",
  type = "rotation",
  lwd = 3,
  width = 0.25,
  barblen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] + 2,
       texts = "Flux causal local (theta_y=31.54 , theta_xy=58.46)", color = "black")

```

```

#=====

# seta vermella
arrow3d(
  p0 = p0_cond,
  p1 = p1_cond,
  col = "red3",
  type = "rotation",
  lwd = 3,
  width = 0.25,
  barblen = 0.02
)

aspect3d(1,1,1)
text3d(p1_cond[1], p1_cond[2], p1_cond[3] + 4,
       texts = sprintf("Flux condicional (theta_y=66.23 , theta_xy=23.77)"),
       color = "black")

#
# aspect3d(1,1,1)

# Límits actuals del sistema de coordenades rgl
bbox <- par3d("bbox") # retorna un vector amb xmin, xmax, ymin, ymax, zmin, zmax
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

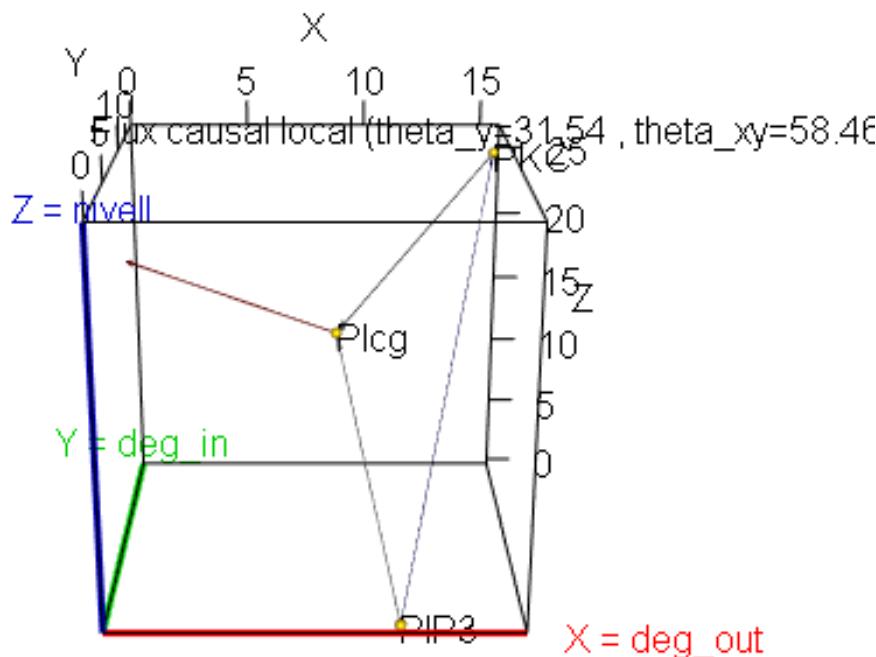
# eixos segons el contingut visible
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



El mòdul del vector

```
# vectors entrants al node Plcg
v_PIP3 <- coords3Dn_diag_corr1["Plcg", c("X", "Y", "Z")] -
  coords3Dn_diag_corr1["PIP3", c("X", "Y", "Z")]

v_PKC <- coords3Dn_diag_corr1["Plcg", c("X", "Y", "Z")] -
  coords3Dn_diag_corr1["PKC", c("X", "Y", "Z")]

cat("longitud dels dos vectors")

## longitud dels dos vectors

# mòduls reals
L_PIP3 <- sqrt(sum(v_PIP3^2))
L_PKC <- sqrt(sum(v_PKC^2))

L_PIP3

## [1] 16.86424

L_PKC

## [1] 13.72194

coords3Dn_diag_corr1["Plcg",] - coords3Dn_diag_corr1["PIP3",]

##          X          Y          Z
## -2.338934  8.419919 14.423491
```

```

coords3Dn_diag_corr1["Plcg",] - coords3Dn_diag_corr1["PKC",]

##           X           Y           Z
## -6.826666 -6.047392 -10.252680

# mòduls reals dibuixats al plot
mod_blaue_plot <- sqrt(sum((p1_flux - p0_flux)^2))
mod_vermell_plot <- sqrt(sum((p1_cond - p0_cond)^2))

cat("Mòdul blau real (plot) : ", mod_blaue_plot, "\n")
## Mòdul blau real (plot) : 28.95438

cat("Mòdul vermell real (plot) : ", mod_vermell_plot, "\n")
## Mòdul vermell real (plot) : 9.909346

```

La p-vàlua és petita en extrem, per tant, les dades són incompatibles amb H\_0. Hom REBUTJA que siguin independents condicionant per PKC H\_0: PIP3 || PKC | Plcg hi ha evidència empírica a fi de rebutjar la independència condicionant en Plcg entre PIP3 i PKC

```

cadena3n_vertical <- dagitty('dag {
bb="-0.5,-0.5,0.5,0.5"
Plcg [pos="12.801835, 8.419918510, 14.423490868"]
PIP3 [pos="15.140769, 0.000000000, 0.000000000"]
PKC [pos="19.628501, 14.467310719, 24.676170477"]
PIP3 -> Plcg
Plcg -> PKC
}
')

(pathway_ind_con <- localTests(x = cadena3n_vertical, data = dades[,c(3,5,9)], type =
"cis"))

##                               estimate      p.value      2.5%      97.5%
## PIP3 _||_ PKC | Plcg 0.04603344 6.491795e-07 0.02791405 0.06412268

```

### 5.6.3 SUBGRAF AïLLAT: CADENA LATERAL DE N=3 NODES AMB TEST NEGATIU del model geomètric topoempíric

```

cadena3n_lateral <- subgraph_from_edges(g_bosc, eids = c(1,18))

arrels1 <- V(cadena3n_lateral)$name[degree(cadena3n_lateral, mode="in") == 0]
fulles1 <- V(cadena3n_lateral)$name[degree(cadena3n_lateral, mode="out") == 0]
noms_nodes1 <- V(cadena3n_lateral)$name

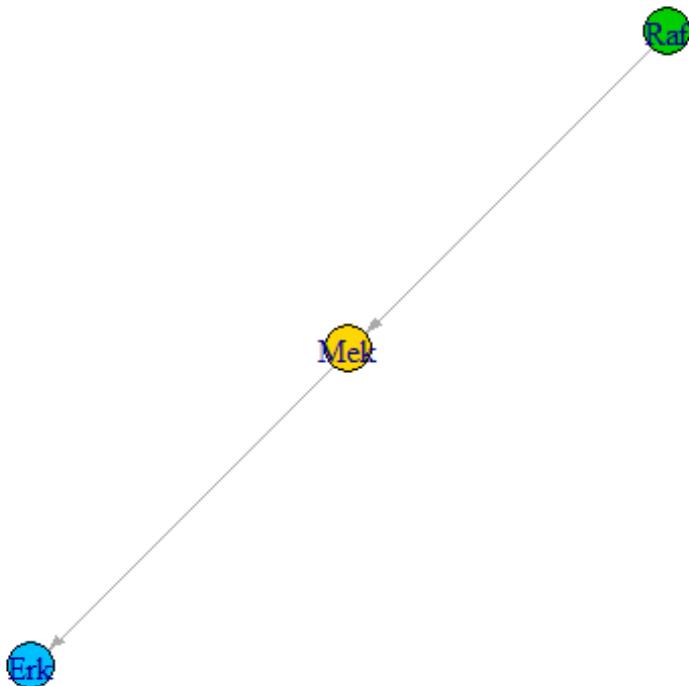
V(cadena3n_lateral)$role <- ifelse(!is.na(match(noms_nodes1, arrels1)), "arrel",
ifelse(!is.na(match(noms_nodes1,fulles1 )), "fulla", "mitjancer"))

V(cadena3n_lateral)$color <- ifelse(V(cadena3n_lateral)$role == "arrel" , "green3",
ifelse(V(cadena3n_lateral)$role == "fulla", "deepskyblue",
"gold"))

par(mar=c(0,0,1,0))
plot(cadena3n_lateral,
edge.arrow.size=0.4,
vertex.size=15,
main ="subgraf cadena lateral de 3 nodes")

```

## subgraf cadena lateral de 3 nodes



Aquí també hom rebutja la independència

```
cadena3n_lateral1 <- dagitty('dag {  
bb="-0.5,-0.5,0.5,0.5"  
Raf [pos="8.017768 , 16.705511579 , 14.051138412"]  
Mek [pos="7.997961 , 29.315418380 , 11.497698618"]  
Erk [pos="0.000000, 15.550560292, 14.465575026"]  
Raf -> Mek  
Mek -> Erk  
}  
)  
(pathway_ind_con <- localTests(x = cadena3n_lateral1, data = dades[,c(1,2,6)], type =  
"cis"))  
  
## estimate p.value 2.5% 97.5%  
## Erk _||_ Raf | Mek -0.08165858 9.544188e-19 -0.09965431 -0.06361018  
  
nodes_cadena_lat <- c("Raf", "Mek", "Erk")  
cadena3n_lateral <- induced_subgraph(g_bosc, vids = nodes_cadena_lat)  
  
(theta_cadena3n_lat <- theta_flux_vectorial_z_fisic(cadena3n_lateral,  
coords3Dn_diag_corr1))  
  
## $u_bar  
## [1] -8.0177678 -1.1549513 0.4144366  
##  
## $len_u  
## [1] 8.11112  
##  
## $flux_z  
## [1] 0.4144366  
##  
## $theta_z
```

```

## [1] 87.0712
##
## $signe
## [1] 1

cat("Angle del flux lateral de la cadena Raf->Mek->Erk:",
round(as.numeric(theta_cadena3n_lat$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena Raf->Mek->Erk: 87.07 graus

Ara cal calcular l'angle del flux condicionat (el node medial de la cadena esdevé un col·lididor)

collididor_artificial_lat <- crea_collididor_artificial(g_bosc, c("Raf", "Mek", "Erk"))

(theta_col_Mek <- theta_flux_collididor_Z_fisic(collididor_artificial_lat,
coords3Dn_diag_corr1))

## $r_in
## [1] 7.974481 26.362619 -5.518774
##
## $len_r
## [1] 28.0898
##
## $flux_z
## [1] -5.518774
##
## $theta_z
## [1] 78.66945
##
## $signe
## [1] -1

cat(sprintf("Angle del flux condicionat (theta_col_Mek PIP2) respecte a Z: %.2f\n",
theta_col_Mek$theta_z))

## Angle del flux condicionat (theta_col_Mek PIP2) respecte a Z: 78.67

print(as_edgelist(collididor_artificial_lat))

##      [,1]  [,2]
## [1,] "Raf" "Mek"
## [2,] "Erk" "Mek"

nodes_cadena_lat <- c("Raf", "Mek", "Erk")
cadena3n_lateral <- induced_subgraph(g_bosc, vids = nodes_cadena_lat)

(theta_cadena3n_lat <- theta_flux_vectorial_XY_fisic(cadena3n_lateral,
coords3Dn_diag_corr1))

## $u_bar
## [1] -8.0177678 -1.1549513  0.4144366
##
## $len_u
## [1] 8.11112
##
## $flux_z
## [1] 0.4144366
##

```

```

## $theta_xy
## [1] 2.928796
##
## $signe
## [1] 1

cat("Angle del flux lateral de la cadena Raf->Mek->Erk:",
round(as.numeric(theta_cadena3n_lat$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena Raf->Mek->Erk: 2.93 graus

collididor_artificial_lat <- crea_collididor_artificial(g_bosc, c("Raf", "Mek", "Erk"))

(theta_col_Mek <- theta_flux_collididor_XY_fisic(collididor_artificial_lat,
coords3Dn_diag_corr1))

## $r_in
## [1] 7.978155 26.374765 -5.521316
##
## $len_r
## [1] 28.10274
##
## $flux_z
## [1] -5.521316
##
## $theta_xy
## [1] 11.33055
##
## $signe
## [1] -1

cat(sprintf("Angle del flux condicionat (theta_col_Mek PIP2) respecte a Z: %.2f\n",
theta_col_Mek$theta_xy))

## Angle del flux condicionat (theta_col_Mek PIP2) respecte a Z: 11.33

print(as_edgelist(collididor_artificial_lat))

##      [,1]  [,2]
## [1,] "Raf" "Mek"
## [2,] "Erk" "Mek"

# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:
# el p1_flux té valors negatius i cal fer el recalc de les coordenades abans de fer el plot3d

nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)
coords_cadena3nRaf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y", "Z")]

# Longitud camí
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

```

```

p0_flux <- coords_cadena3nRaf_Mek_Erk["Raf", c("X", "Y", "Z")]
p1_flux <- p0_flux + len_flux * u_dir

coords_CADENA_lat <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y", "Z")]

# [1]

# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords <- rbind(coords_CADENA_lat,
                      p0_flux, p1_flux)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset1 <- apply(all_coords, 2, min)

# hom aplica el desplaçament
coords_CADENA_lat <- sweep(coords_CADENA_lat, 2, offset1, "-")
p0_flux <- p0_flux - offset1
p1_flux <- p1_flux - offset1

# [2]

open3d()

## wgl
## 34

# dibuixa els punts
plot3d(coords_CADENA_lat, col = "gold", size = 1, type = "s", box = TRUE)

# arestes direccionals
edges <- as_edgelist(cadena3n_Raf_Mek_Erk)
for (i in seq_len(nrow(edges))) {
  p0 <- coords_CADENA_lat[edges[i, 1], ] # origen pel nom
  p1 <- coords_CADENA_lat[edges[i, 2], ] # destí pel nom

  arrow3d(
    p0 = as.numeric(p0),
    p1 = as.numeric(p1),
    type = "rotation",
    col = "grey40",
    lwd = 2,
    width = 0.2,
    barblen = 0.025,
    n = 4
  )
}

text3d(coords_CADENA_lat[, 1] + 0.02,
       coords_CADENA_lat[, 2],
       coords_CADENA_lat[, 3] + 0.04,
       texts = rownames(coords_CADENA_lat),
       adj = c(0, 0.5))

```

```

# segeta blava de vector director calculat al bloc de codi [1]
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "blue",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] -0.25,
       texts = "Flux causal local", color = "black")
aspect3d(1,1,1)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

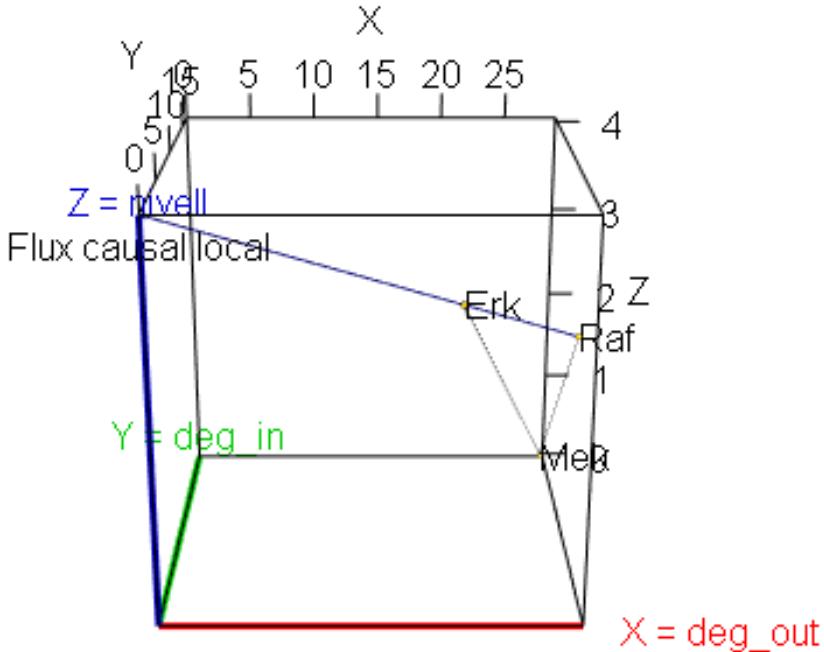
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



```

nodes_cadena_lat <- c("Raf", "Mek", "Erk")
cadena3n_lat <- induced_subgraph(g_bosc, vids = nodes_cadena_lat)
edges_lat <- as_edgelist(cadena3n_lat)

coords_CADENA_lat <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y", "Z")]

# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:
# el p1_flux té valors negatius i cal fer el recalc dels calculs abans de fer el plot3d
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)
coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y", "Z")]

# Longitud camí
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

p0_flux <- coords_cadena3n_Raf_Mek_Erk["Raf",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux * u_dir

#
#SEGETA VERMELLA

```

```

res_cond <- theta_flux_collididor_z_fisic(
  crea_collididor_artificial(g_bosc_vertical, c("Raf", "Mek", "Erk")),
  coords3Dn_diag_corr1)

u_cond <- res_cond$r_in # vector direccional condicionat: r_out <- r_in era correcte per
tal de calcular l'angle (perquè volem l'efecte reacció), però per dibuixar la direcció
del flux condicionat, el vector que hom vol mostrar és r_in, no r_out
len_flux_cond <- sqrt(sum(u_cond^2)) # mòdul real del flux condicionat (no del vector
unitari!)
u_dir_cond <- u_cond / len_flux_cond # vector unitari de direcció

p0_cond <- coords_CADENA_lat["Mek", c("X","Y","Z")] # origen del flux condicionat
p1_cond <- p0_cond + len_flux_cond * u_dir_cond # punt final del flux

# _____
# Mòduls reals dibuixats al plot
mod_blau_plot <- sqrt(sum((p1_flux - p0_flux)^2))
mod_vermell_plot <- sqrt(sum((p1_cond - p0_cond)^2))

cat("Mòdul blau real (plot) : ", mod_blau_plot, "\n")
## Mòdul blau real (plot) : 29.0599
cat("Mòdul vermell real (plot) : ", mod_vermell_plot, "\n")
## Mòdul vermell real (plot) : 28.0898

# bloc de codi fins a [3] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords <- rbind(coords_CADENA_lat,
                      p0_flux, p1_flux, p0_cond, p1_cond )
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset <- apply(all_coords, 2, min)

# hom aplica el desplaçament
coords_CADENA_lat <- sweep(coords_CADENA_lat, 2, offset, "-")
p0_flux <- p0_flux - offset
p1_flux <- p1_flux - offset
p0_cond <- p0_cond - offset
p1_cond <- p1_cond - offset
# [3] _____

clear3d()
close3d()
open3d()

## wgl
## 37

# dibuixa els punts
plot3d(coords_CADENA_lat, col = "gold", size = 1, type = "s", box = TRUE)

```

```

# arestes direccionals
edges <- as_edgelist(cadena3n)

cap1 <- coords_CADENA_lat[edges_lat[1, 1], ] # origen pel nom
cua1 <- coords_CADENA_lat[edges_lat[1, 2], ] # destí pel nom

cap2 <- coords_CADENA_lat[edges_lat[2, 2], ] # origen pel nom
cua2 <- coords_CADENA_lat[edges_lat[2, 1], ] # destí pel nom

arrow3d(
  p0 = as.numeric(cua1),
  p1 = as.numeric(cap1),
  type = "rotation",
  col = "grey40",
  lwd = 2,
  width = 0.2,
  barblen = 0.025,
  n = 4
)

arrow3d(
  p0 = as.numeric(cua2),
  p1 = as.numeric(cap2),
  type = "rotation",
  col = "grey40",
  lwd = 2,
  width = 0.2,
  barblen = 0.025,
  n = 4
)

text3d(coords_CADENA_lat[,1] + 0.02,
       coords_CADENA_lat[,2],
       coords_CADENA_lat[,3] + 0.04,
       texts = rownames(coords_CADENA_lat),
       adj = c(0,0.5))

# segeta blava de vector director calculat al bloc de codi [1]
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "blue",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] -0.25,
       texts = sprintf("Flux causal local(theta_y=87.07 , theta_xy=2.93)",
       color = "black"))

```

```

## Warning in sprintf("Flux causal local(theta_y=87.07 , theta_xy=2.93)", color =
## "black"): one argument not used by format 'Flux causal local(theta_y=87.07 ,
## theta_xy=2.93)'

#=====

# seteta vermelha
arrow3d(
  p0 = p0_cond,
  p1 = p1_cond,
  col = "red3",
  type = "rotation",
  lwd = 3,
  width = 0.25,
  barrlen = 0.02
)

aspect3d(1,1,1)
text3d(p1_cond[1], p1_cond[2], p1_cond[3] + 4,
       texts = sprintf("Flux condicional (theta_y=78.67 , theta_xy=11.33)",
                      color = "black"))

## Warning in sprintf("Flux condicional (theta_y=78.67 , theta_xy=11.33)", : one
## argument not used by format 'Flux condicional (theta_y=78.67 , theta_xy=11.33)'

#


---


aspect3d(1,1,1)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

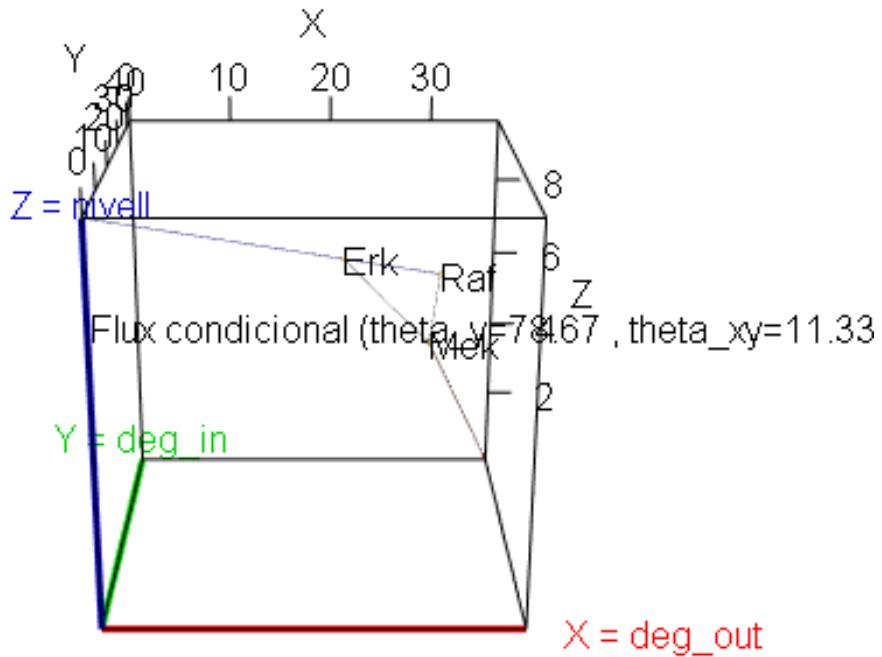
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



#### 5.6.4 BATERIA DE SUBGRAFS AÏLLATS de 3 NODES AMB TEST POSITIU del model geomètric topoempíric

Aquí hi han tots els tests positius de localTest(), és a dir, que les dades referen l'estructura del DAG 2D amb una p\_vàlua igual o més gran que 0.05.

```
LT <- localTests(pathway_net, data = dades, type = "cis")
(LT1 <- LT[c(14:16,25,29,34,36,43:44,48,51,56,58),1:2])

##                                     estimate   p.value
## Akt _||_ PIP2 | PIP3      -0.0136374356 0.1407006
## Akt _||_ PKC | PIP2, Plcg -0.0034278992 0.7111872
## Akt _||_ PKC | PIP3      -0.0039815169 0.6671376
## Erk _||_ PIP2 | PKC      -0.0106795997 0.2486485
## Erk _||_ PIP3 | Mek, PKA  0.0007349409 0.9367273
## Jnk _||_ Mek | PKA, PKC  0.0087265133 0.3458812
## Jnk _||_ PIP2 | PKC      0.0131965556 0.1539968
## Mek _||_ PIP3 | PIP2, Plcg -0.0122087025 0.1872444
## Mek _||_ PIP3 | PKC      -0.0099681582 0.2815780
## P38 _||_ PIP3 | PKC      -0.0015846976 0.8640847
## PIP2 _||_ PKA           0.0109783861 0.2356352
## PIP3 _||_ Raf | PIP2, Plcg 0.0147883923 0.1101614
## PKA _||_ Plcg          0.0147671144 0.1106438
```

D'aquests hom en calcula els angles dels 6 subgrafs aïllats amb 3 nodes, atès que les funcions aquí definides són de 3 nodes. Hom ja veu en què la majoria són implicats amb el node PKC, sobretot amb arestes de retrovertical.

```
(LT2 <- LT1[c(1,3:4,7,9:10),])
```

```

##                                estimate   p.value
## Akt _||_ PIP2 | PIP3 -0.013637436 0.1407006
## Akt _||_ PKC | PIP3 -0.003981517 0.6671376
## Erk _||_ PIP2 | PKC -0.010679600 0.2486485
## Jnk _||_ PIP2 | PKC  0.013196556 0.1539968
## Mek _||_ PIP3 | PKC -0.009968158 0.2815780
## P38 _||_ PIP3 | PKC -0.001584698 0.8640847

```

### P38 || PIP3 | PKC (planteable com a cadena PIP3->...->PKC->P38)

```

nodes_cadena_1 <- c("PIP3", "PKC", "P38")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_Z_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -18.480944 -1.507105 -19.226814
##
## $len_u
## [1] 26.71118
##
## $flux_z
## [1] -19.22681
##
## $theta_z
## [1] 43.9617
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena PIP2->PKC->P38:",
round(as.numeric(theta_cadena3n_1$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena PIP2->PKC->P38: 43.96 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "P38"))

(theta_col_PKC <- theta_flux_collididor_Z_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 22.96600 15.97255 43.89787
##
## $len_r
## [1] 52.05365
##
## $flux_z
## [1] 43.89787
##
## $theta_z
## [1] 32.50765
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_z))

```

```

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 32.51

print(as_edgelist(collididor_artificial_1))

## [,1] [,2]
## [1,] "PIP3" "PKC"
## [2,] "P38" "PKC"

nodes_cadena_1 <- c("PIP3", "PKC", "P38")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_XY_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -18.480944 -1.507105 -19.226814
##
## $len_u
## [1] 26.71118
##
## $flux_z
## [1] -19.22681
##
## $theta_xy
## [1] 46.0383
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena PIP3->PKC->P38:",
round(as.numeric(theta_cadena3n_1$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena PIP3->PKC->P38: 46.04 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "P38"))

(theta_col_PKC <- theta_flux_collididor_XY_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 22.96868 15.97442 43.90298
##
## $len_r
## [1] 52.05972
##
## $flux_z
## [1] 43.90298
##
## $theta_xy
## [1] 57.49235
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_xy))

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 57.49

```

```

print(as_edgelist(collididor_artificial_1))

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "P38"  "PKC"

####Jnk || PIP2 | PKC (plantejable com a cadena PIP2->PKC->Jnk)

nodes_cadena_1 <- c("PIP2", "PKC", "Jnk")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_Z_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -6.522544 -2.739630 -4.066662
##
## $len_u
## [1] 8.160079
##
## $flux_z
## [1] -4.066662
##
## $theta_z
## [1] 60.1084
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena PIP2->PKC->Jnk:",
round(as.numeric(theta_cadena3n_1$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena PIP2->PKC->Jnk: 60.11 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP2", "PKC", "Jnk"))

(theta_col_PKC <- theta_flux_collididor_Z_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 30.722464 -4.747824 23.960382
##
## $len_r
## [1] 39.24935
##
## $flux_z
## [1] 23.96038
##
## $theta_z
## [1] 52.37682
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_Mek$theta_z))
print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP2" "PKC"
## [2,] "Jnk"   "PKC"

nodes_cadena_1 <- c("PIP2", "PKC", "Jnk")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_XY_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -6.522544 -2.739630 -4.066662
##
## $len_u
## [1] 8.160079
##
## $flux_z
## [1] -4.066662
##
## $theta_xy
## [1] 29.8916
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena PIP2->PKC->Jnk:",
round(as.numeric(theta_cadena3n_1$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena PIP2->PKC->Jnk: 29.89 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP2", "PKC", "Jnk"))

(theta_col_PKC <- theta_flux_collididor_XY_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 30.727452 -4.748595 23.964272
##
## $len_r
## [1] 39.25572
##
## $flux_z
## [1] 23.96427
##
## $theta_xy
## [1] 37.62318
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_xy))

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 37.62

print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP2" "PKC"
## [2,] "Jnk"   "PKC"

Erk || PIP2 | PKC (plantejable com a cadena (?) PIP2->PKC[->]Erk)

nodes_cadena_1 <- c("PIP2", "PKC", "Erk")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_Z_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] 12.102454 -3.744112  9.948805
##
## $len_u
## [1] 16.10796
##
## $flux_z
## [1] 9.948805
##
## $theta_z
## [1] 51.85653
##
## $signe
## [1] 1

cat("Angle del flux lateral de la cadena PIP2->PKC->Erk:",
round(as.numeric(theta_cadena3n_1$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena PIP2->PKC->Erk: 51.86 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP2", "PKC", "Erk"))

(theta_col_PKC <- theta_flux_collididor_Z_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 31.72325 -4.82619 20.15450
##
## $len_r
## [1] 37.89275
##
## $flux_z
## [1] 20.1545
##
## $theta_z
## [1] 57.86724
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_z))

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 57.87

print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP2" "PKC"
## [2,] "Erk"   "PKC"

nodes_cadena_1 <- c("PIP2", "PKC", "Erk")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_XY_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] 12.102454 -3.744112  9.948805
##
## $len_u
## [1] 16.10796
##
## $flux_z
## [1] 9.948805
##
## $theta_xy
## [1] 38.14347
##
## $signe
## [1] 1

cat("Angle del flux lateral de la cadena PIP2->PKC->Erk:",
round(as.numeric(theta_cadena3n_1$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena PIP2->PKC->Erk: 38.14 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP2", "PKC", "Erk"))

(theta_col_PKC <- theta_flux_collididor_XY_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 31.730955 -4.827362 20.159400
##
## $len_r
## [1] 37.90196
##
## $flux_z
## [1] 20.1594
##
## $theta_xy
## [1] 32.13276
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_xy))

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 32.13

print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP2" "PKC"
## [2,] "Erk"   "PKC"



### Akt || PKC | PIP3 (plantejable com a cadena (?) PIP3->...->PKC[->]Akt)


nodes_cadena_1 <- c("PIP3", "PKC", "Akt")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_Z_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -13.78940 16.78402 14.38360
##
## $len_u
## [1] 26.05262
##
## $flux_z
## [1] 14.3836
##
## $theta_z
## [1] 56.48893
##
## $signe
## [1] 1

cat("Angle del flux lateral de la cadena PIP3->PKC->Akt:",
round(as.numeric(theta_cadena3n_1$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena PIP3->PKC->Akt: 56.49 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "Akt"))

(theta_col_PKC <- theta_flux_collididor_Z_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 22.67089 12.10044 34.82438
##
## $len_r
## [1] 43.27964
##
## $flux_z
## [1] 34.82438
##
## $theta_z
## [1] 36.42483
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_Mek$theta_z))
print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "Akt"   "PKC"

nodes_cadena_1 <- c("PIP3", "PKC", "Akt")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_XY_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -13.78940 16.78402 14.38360
##
## $len_u
## [1] 26.05262
##
## $flux_z
## [1] 14.3836
##
## $theta_xy
## [1] 33.51107
##
## $signe
## [1] 1

cat("Angle del flux lateral de la cadena PIP3->PKC->Akt:",
round(as.numeric(theta_cadena3n_1$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena PIP3->PKC->Akt: 33.51 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "Akt"))

(theta_col_PKC <- theta_flux_collididor_XY_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 22.76487 12.15060 34.96874
##
## $len_r
## [1] 43.45905
##
## $flux_z
## [1] 34.96874
##
## $theta_xy
## [1] 53.57517
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_xy))

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 53.58

print(as_edgelist(collididor_artificial_1))

```

```
##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "Akt"   "PKC"
```

## Mek || PIP3 | PKC (planteable com a cadena PIP3->...->PKC->Mek)

```
nodes_cadena_1 <- c("PIP3", "PKC", "Mek")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_Z_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -11.63054 14.84811 -13.17847
##
## $len_u
## [1] 23.00887
##
## $flux_z
## [1] -13.17847
##
## $theta_z
## [1] 55.05735
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena PIP3->PKC->Mek:",
round(as.numeric(theta_cadena3n_1$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena PIP3->PKC->Mek: 55.06 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "Mek"))

(theta_col_PKC <- theta_flux_collididor_Z_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 16.0545391 -0.3792912 37.7049620
##
## $len_r
## [1] 40.98239
##
## $flux_z
## [1] 37.70496
##
## $theta_z
## [1] 23.06977
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_Mek$theta_z))
print(as_edgelist(collididor_artificial_1))
```

```

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "Mek"   "PKC"

nodes_cadena_1 <- c("PIP3", "PKC", "Mek")
cadena3n_1 <- induced_subgraph(g_bosc, vids = nodes_cadena_1)

(theta_cadena3n_1 <- theta_flux_vectorial_XY_fisic(cadena3n_1, coords3Dn_diag_corr1))

## $u_bar
## [1] -11.63054 14.84811 -13.17847
##
## $len_u
## [1] 23.00887
##
## $flux_z
## [1] -13.17847
##
## $theta_xy
## [1] 34.94265
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena PIP3->PKC->Mek:",
round(as.numeric(theta_cadena3n_1$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena PIP3->PKC->Mek: 34.94 graus

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "Mek"))

(theta_col_PKC <- theta_flux_collididor_XY_fisic(collididor_artificial_1,
coords3Dn_diag_corr1))

## $r_in
## [1] 16.1182720 -0.3807969 37.8546423
##
## $len_r
## [1] 41.14508
##
## $flux_z
## [1] 37.85464
##
## $theta_xy
## [1] 66.93023
##
## $signe
## [1] 1

cat(sprintf("Angle del flux condicionat (theta_col_PKC) respecte a Z: %.2f\n",
theta_col_PKC$theta_xy))

## Angle del flux condicionat (theta_col_PKC) respecte a Z: 66.93

print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "Mek"   "PKC"

```

## 5.6.5 CONCLUSIONS SOBRE L'ASSAIG DE VALIDACIÓ LOCAL

Si no hi ha errors de càlcul ni de plantejament, no és correcte comparar l'angle del flux local del vector director del col·lididor (amb condicionament) versus el flux local del vector director de la cadena (sense condicionament) com a aproximació del flux causal general; CAL FER LA COMPARANÇA AMB EL FLUX CAUSAL GENERAL o bé respecte l'eix vertical Z com a aproximació o bé respecte dels tres eixos com a mètode robust.

Les diferències d'angles respecte de Z no estableixen un llindar per a destriar un test positiu d'un altre de negatiu.

Un altra cosa fóra convertir aquests subgrafs com a grafs, és a dir, com a exemples trivials, un model geomètric basat sols en 3 nodes amb un joc de dades de sols 3 columnes. Entendre si l'heurística geomètrica aquí definida és consistent amb la semàntica causal en condicions pures.

Idealment un cadena de 3 nodes, diguem-ne canònica té un angle respecte de Z de 66 graus. Però és ideal perquè és una cadena del tot recta, en un DAG 2D biològic real les cadenes de 3 nodes no són rectes; a més, a dins d'un DAG 2D hom no veu la disposició entre nivells ni tampoc lateralment.

```

$$ Node =(grau_{in}, grau_{out}, nivell)\\ Node_A=(1,0,1)\\ Node_B=(2,1,1)\\
Node_C=(3,1,0)\\.\\ vector \space \Delta X \space \Delta Y \space \Delta Z \space Angle \space
amb \space Z \\ AB \space [1, 1, 0] \space 90^\circ \\ BC \space [1, 0,-1] \space 45^\circ \\ AC
\space [2, 1,-1] \space 66^\circ \\ $$ 

```

## 5.7 ASSAIG DE VALIDACIÓ LOCAL (*l'aresta absent entre Akt i Erk segons reporten SACHS et al.) A PARTIR DEL FLUX CAUSAL GENERAL*

### *Introducció*

Analitzant l'estruccura geomètrica del model causal topoempíric hom en descriu els diversos trets:

→ Hi han dues vies de flux causal encapçalades pels dos nodes arrels dels dos arbres del bosc: PIP3 i PKA → PIP3 no té cap camí de flux cap a una fulla pròpria final (PKC acaba essent un node medial), sols un node col·lididor-fulla vertical amb PKA → PKA té un únic camí de flux cap a una fulla pròpria final però retroverticalitzada per la cadena Raf->Mek->Erk, la resta de nodes finals són col·liditors retroverticals (cosa que debilita en gran mesura el flux tant de PKA com de PIP3 però els seus fluxos de sentit contrari xoquen als nodes col·liditors).

Concloent, els dos nodes terminals de flux verticals finals que semblen d'entrada operatius són un col·lididor-fulla (Akt) de PIP3 i PKA i un node-fulla (Erk) però retroverticalitzat de PKA. Hom ja veu que el node PKC és el taló d'Aquil·les del model justament perquè és el principal obstacle del flux causal vertical.

LA QÜESTIÓ CLAU és si el node fulla vertical retroverticalitzat Erk és connectat a l'únic node col·lididor de flux vertical. L'efecte dels nodes arrels convergeix o no a Akt? Caldrà veure si l'angle de flux general respecte de les cadenes Raf->Mek->Erk retrovertical o Raf->Mek->Erk lateral convergeixen.

Que en termes d'inferència causal clàssica fóra si Akt i Erk són independents si hom condiciona per PKA.

## 5.7.1 FLUX CAUSAL LOCAL de la cadena lateral en el marc del model geomètric topoempíric general

Aquí també hom rebutja la independència

```
cadena3n_vertical2 <- dagitty('dag {  
bb="-0.5,-0.5,0.5,0.5"  
Akt [pos="1.351367, 16.784021406, 14.383598584"]  
PKA [pos="27.232341 , 0.009952487, 0.009952487"]  
Erk [pos="0.000000, 15.550560292, 14.465575026"]  
PKA -> Akt  
PKA -> Erk  
}  
)  
(pathway_ind_con <- localTests(x = cadena3n_vertical2, data = dades[,c(7,8,6)], type =  
"cis"))  
  
## estimate p.value 2.5% 97.5%  
## Akt _||_ Erk | PKA 0.8974968 0 0.9641513 0.9757488
```

Cal anar en compte amb l'escalatge non-uniforme dels eixos de rgl tot i tenir el mateix rang, rgl no dibuixa els eixos amb la mateixa distància física (la distància entre ticks); és a dir, l'eix Y té increments de 5 unitats com la resta d'eixos, però cada 5 unitats a l'eix Y són físicament més curtes que no pas les mateixes unitats a l'eix X o Z. D'aquí que el vector resultant vector director blau sembli molt més llarg del que veritablement és.

```
# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:  
# el p1_flux té valors negatius i cal fer el recalc de les coordenades abans de fer el plot3d  
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")  
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)  
coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y", "Z")]  
  
# Longitud camí  
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)  
  
res_flux <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)  
u_bar <- as.numeric(res_flux$u_bar)  
u_dir <- u_bar / sqrt(sum(u_bar^2))  
  
p0_flux <- coords_cadena3n_Raf_Mek_Erk["Raf",c("X","Y","Z")]  
p1_flux <- p0_flux + len_flux * u_dir  
  
# [1]  
  
# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu  
# hom combina totes les coordenades que apareixen al gràfic  
all_coords <- rbind(coords3Dn_diag_corr1,  
                      p0_flux, p1_flux)  
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000  
offset2 <- apply(all_coords, 2, min)
```

```

# hom aplica el desplaçament
coords3Dn_diag_corr2 <- sweep(coords3Dn_diag_corr1, 2, offset2, "-")
p0_flux <- p0_flux - offset2
p1_flux <- p1_flux - offset2

# [2] _____
```

---

```

open3d()

## wgl
## 40

plot3d(coords3Dn_diag_corr2, col = "orangered", size = 1, type = "s")

edges <- as_edgelist(g_bosc)
edges1 <- edges[-c(1,18),]

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retrolateral (retroflux i Lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr2[v1, ], coords3Dn_diag_corr2[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr2))
text3d(coords3Dn_diag_corr2[, "X"]+0.02,
       coords3Dn_diag_corr2[, "Y"],
       coords3Dn_diag_corr2[, "Z"]+offset,
       texts = rownames(coords3Dn_diag_corr2),
       adj = c(0, 0.2),
       cex=0.9)

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr2[7, ], coords3Dn_diag_corr2[1, ]),
           col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr2[1, ], coords3Dn_diag_corr2[8, ]),
           col = "purple", lwd = 1.5)
```

```

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] + 4,
       texts = "Flux causal local", color = "black")

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr2) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

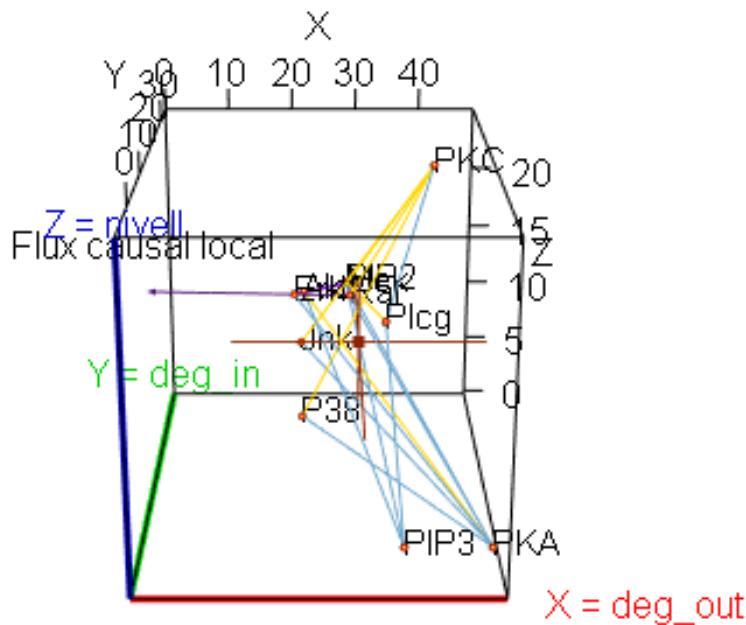
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



### 5.7.2 FLUX CAUSAL VERTICAL GENERAL del model geomètric topoempíric general

Essent fidel a l'arbre vertical sols n'hi han tres, de vectors directors verticals purs, és a dir, que són estrictament verticals sense contraflux (si hom considerés els dos col·lididors-fulla verticals, PIP3->Akt->PKA i Plcg->PKC<-PIP2, els resultats foren molt semblants i no canviarien gaire els angles; hom n'agafa sols l'aresta de cadascuna que té la tendència a esser més significativa en el conjunt del model, aspecte ja analitzat abans. Aquest dilema no hi és en PKA->Erk<-Mek perquè és un col·lididor fulla laterovertical, cosa que sols et deixa l'opció d'agafar l'aresta vertical). →Un vector director PIP3-Plcg-PKC →Tres vectors directors d'una aresta: PIP3-PIP2, PIP3-Akt i PKA-Erk (un vector director d'una aresta és la mateixa aresta)

```
# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:
# el p1_flux té valors negatius i cal fer el recalc de les coordenades abans de fer el plot3d
nodes_cadena_PIP3_Plcg_PKC <- c("PIP3", "Plcg", "PKC")
cadena3n_PIP3_Plcg_PKC <- induced_subgraph(g_bosc, vids = nodes_cadena_PIP3_Plcg_PKC)
coords_cadena3n_PIP3_Plcg_PKC <- coords3Dn_diag_corr1[c("PIP3", "Plcg", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux0 <- longitud_cami_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)

res_flux0 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)
u_bar0 <- as.numeric(res_flux0$u_bar)
u_dir0 <- u_bar0 / sqrt(sum(u_bar0^2))

p0_flux0 <- coords_cadena3n_PIP3_Plcg_PKC[ "PIP3",c("X","Y","Z")]
```

```

p1_flux0 <- p0_flux0 + len_flux0 * u_dir0

# [1]_____
# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux0, p1_flux0)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset3 <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
coords3Dn_diag_corr3 <- sweep(coords3Dn_diag_corr1, 2, offset3, "-")
p0_flux0 <- p0_flux0 - offset3
p1_flux0 <- p1_flux0 - offset3

# [2]_____

```

```

open3d()

## wgl
## 43

plot3d(coords3Dn_diag_corr3, col = "orangered", size = 1, type = "s")

edges <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retrolateral (retroflux i lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr3[v1, ], coords3Dn_diag_corr3[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr3))
text3d(coords3Dn_diag_corr3[, "X"]+0.02,

```

```

coords3Dn_diag_corr3[, "Y"],
coords3Dn_diag_corr3[, "Z"]+2,
texts = rownames(coords3Dn_diag_corr3),
adj = c(0, 0.2),
cex=0.9)

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr3[7, ], coords3Dn_diag_corr3[1, ]),col = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr3[1, ], coords3Dn_diag_corr3[8, ]),col = "blue", lwd = 1.5)

# segeta blava1
arrow3d(
  p0 = p0_flux0,
  p1 = p1_flux0,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux0[1], p1_flux0[2], p1_flux0[3] + 4,
       texts = "Flux vertical 1", color = "black")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr3[3,] ,
  p1 = coords3Dn_diag_corr3[2,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 2", color = "black")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr3[3,] ,
  p1 = coords3Dn_diag_corr3[9,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 3", color = "black")

```

```

# segeta blava3
arrow3d(
  p0 = coords3Dn_diag_corr3[4] ,
  p1 = coords3Dn_diag_corr3[8] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 4", color = "black")

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr3) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

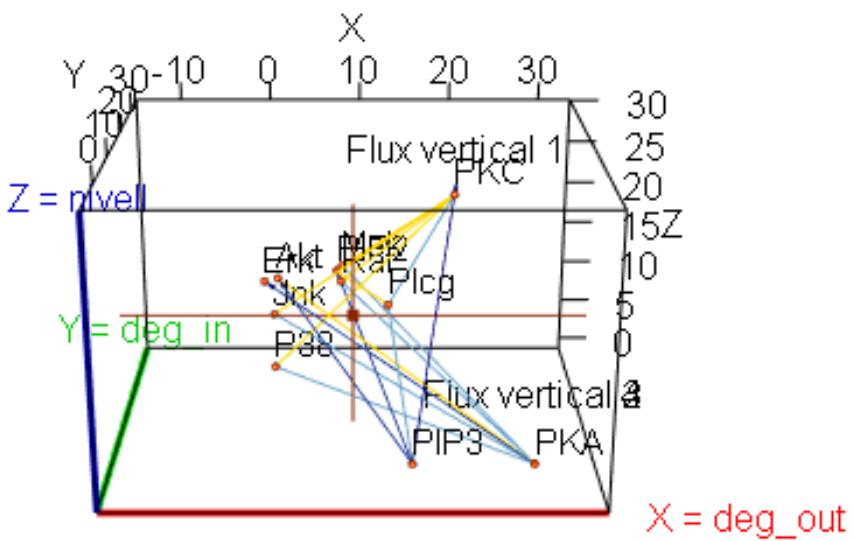
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



### 5.7.3 FLUX CAUSAL RETROVERTICAL del model geomètric topoempíric general

Hi han 4 col·lididors retroverticals

```
# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:
# el p1_flux té valors negatius i cal fer el recalc dels valors abans de fer el plot3d
nodes_collididor_PKA_P38_PKC <- c("PKA", "P38", "PKC")
collididor3n_PKA_P38_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_P38_PKC)
coords_collididor_PKA_P38_PKC <- coords3Dn_diag_corr1[c("PKA", "P38", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux1 <- longitud_cami_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1)

res_flux1 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
r_in1 <- as.numeric(res_flux1$r_in)
u_dir1 <- r_in1 / sqrt(sum(r_in1^2))

p0_flux1 <- coords_collididor_PKA_P38_PKC["P38",c("X","Y","Z")]
p1_flux1 <- p0_flux1 + len_flux1 * u_dir1
#[1]

nodes_collididor_PKA_Jnk_PKC <- c("PKA", "Jnk", "PKC")
collididor3n_PKA_Jnk_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Jnk_PKC)
coords_collididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X", "Y", "Z")]
```

```

# Longitud camí
len_flux2 <- longitud_cami_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

res_flux2 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)
r_in2 <- as.numeric(res_flux2$r_in)
u_dir2 <- r_in2 / sqrt(sum(r_in2^2))

p0_flux2 <- coords_collididor_PKA_Jnk_PKC["Jnk",c("X","Y","Z")]
p1_flux2 <- p0_flux2 + len_flux2 * u_dir2

# _____
# Longitud camí
nodes_collididor_PKA_Mek_PKC <- c("PKA", "Mek", "PKC")
collididor3n_PKA_Mek_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Mek_PKC)
coords_collididor_PKA_Mek_PKC <- coords3Dn_diag_corr1[c("PKA", "Mek", "PKC"), c("X",
"Y", "Z")]

# Longitud camí
len_flux3 <- longitud_cami_fisic(collididor3n_PKA_Mek_PKC, coords3Dn_diag_corr1)

res_flux3 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Mek_PKC,
coords3Dn_diag_corr1)
r_in3 <- as.numeric(res_flux3$r_in)
u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

p0_flux3 <- coords_collididor_PKA_Mek_PKC["Mek",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3

# _____
# Longitud camí
nodes_collididor_PKA_Raf_PKC <- c("PKA", "Raf", "PKC")
collididor3n_PKA_Raf_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Raf_PKC)
coords_collididor_PKA_Raf_PKC <- coords3Dn_diag_corr1[c("PKA", "Raf", "PKC"), c("X",
"Y", "Z")]

# Longitud camí
len_flux4 <- longitud_cami_fisic(collididor3n_PKA_Raf_PKC, coords3Dn_diag_corr1)

res_flux4 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Raf_PKC,
coords3Dn_diag_corr1)
r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_collididor_PKA_Raf_PKC["Raf",c("X","Y","Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

# _____
# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic

```

```

all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux1, p1_flux1,
                      p0_flux2, p1_flux2,
                      p0_flux3, p1_flux3,
                      p0_flux4, p1_flux4)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset4 <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
coords3Dn_diag_corr4 <- sweep(coords3Dn_diag_corr1, 2, offset4, "-")
p0_flux1 <- p0_flux1 - offset4
p1_flux1 <- p1_flux1 - offset4
p0_flux2 <- p0_flux2 - offset4
p1_flux2 <- p1_flux2 - offset4
p0_flux3 <- p0_flux3 - offset4
p1_flux3 <- p1_flux3 - offset4
p0_flux4 <- p0_flux4 - offset4
p1_flux4 <- p1_flux4 - offset4

```

# [2]

---

```
open3d()
```

```
## wgl
## 46
```

```
plot3d(coords3Dn_diag_corr4, col = "orangered", size = 1, type = "s")
```

```
edges <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]
```

# hom converteix els subgrafs en vectors d'arestes (com "A\_B")

```
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")
```

# assigna color segons pertinença

```
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retroLateral (retroflux i Lateral)
  }
})
```

```
for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr4[v1, ], coords3Dn_diag_corr4[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}
```

```

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr4))
text3d(coords3Dn_diag_corr4[, "X"]+0.02,
       coords3Dn_diag_corr4[, "Y"],
       coords3Dn_diag_corr4[, "Z"]+2,
       texts = rownames(coords3Dn_diag_corr4),
       adj = c(0, 0.2),
       cex=0.9)

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr4[7, ], coords3Dn_diag_corr4[1, ]), col = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr4[1, ], coords3Dn_diag_corr4[8, ]), col = "blue", lwd = 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux1[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical 1", color = "black")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 2", color = "black")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 3", color = "black")

```

```

# se geta bLava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 4", color = "black")

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr4) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

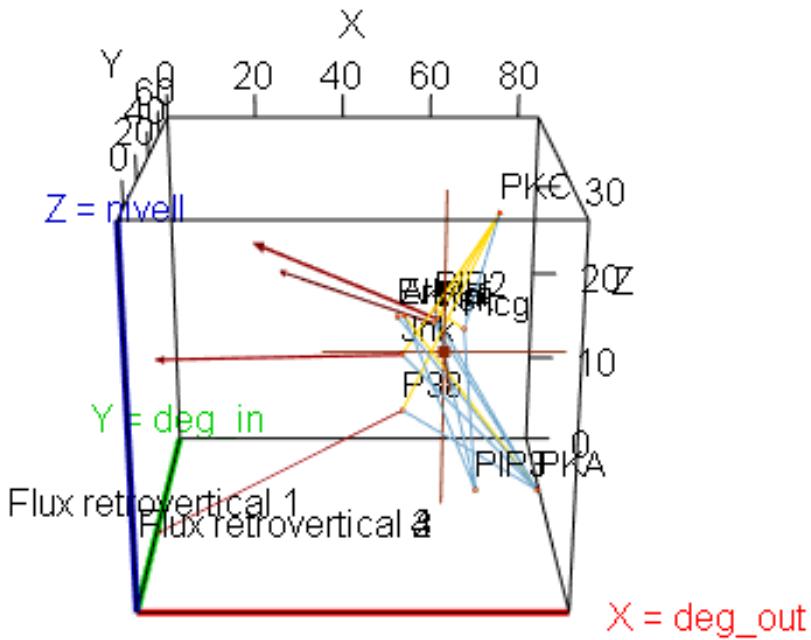
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



#### 5.7.4 FLUX CAUSAL CORREGIT VERTICAL del model geomètric topoempíric general

```

nodes_cadena_PIP3_Plcg_PKC <- c("PIP3", "Plcg", "PKC")
cadena3n_PIP3_Plcg_PKC <- induced_subgraph(g_bosc, vids = nodes_cadena_PIP3_Plcg_PKC)
coords_cadena3n_PIP3_Plcg_PKC <- coords3Dn_diag_corr1[c("PIP3", "Plcg", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux0 <- longitud_cami_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)

res_flux0 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)
u_bar0 <- as.numeric(res_flux0$u_bar)
u_dir0 <- u_bar0 / sqrt(sum(u_bar0^2))

p0_flux0 <- coords_cadena3n_PIP3_Plcg_PKC["PIP3",c("X","Y","Z")]
p1_flux0 <- p0_flux0 + len_flux0 * u_dir0

#
# _____
nodes_collididor_PKA_P38_PKC <- c("PKA", "P38", "PKC")
collididor3n_PKA_P38_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_P38_PKC)
coords_collididor_PKA_P38_PKC <- coords3Dn_diag_corr1[c("PKA", "P38", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux1 <- longitud_cami_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1)

```

```

res_flux1 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
r_in1 <- as.numeric(res_flux1$r_in)
u_dir1 <- r_in1 / sqrt(sum(r_in1^2))

p0_flux1 <- coords_collididor_PKA_P38_PKC[ "P38",c("X","Y","Z")]
p1_flux1 <- p0_flux1 + len_flux1 * u_dir1
#[1] _____
```

---

```

nodes_collididor_PKA_Jnk_PKC <- c("PKA", "Jnk", "PKC")
collididor3n_PKA_Jnk_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Jnk_PKC)
coords_collididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X",
"Y", "Z")]

# Longitud camí
len_flux2 <- longitud_cami_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

res_flux2 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)
r_in2 <- as.numeric(res_flux2$r_in)
u_dir2 <- r_in2 / sqrt(sum(r_in2^2))

p0_flux2 <- coords_collididor_PKA_Jnk_PKC[ "Jnk",c("X","Y","Z")]
p1_flux2 <- p0_flux2 + len_flux2 * u_dir2
```

---

```

# _____
```

---

```

nodes_collididor_PKA_Mek_PKC <- c("PKA", "Mek", "PKC")
collididor3n_PKA_Mek_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Mek_PKC)
coords_collididor_PKA_Mek_PKC <- coords3Dn_diag_corr1[c("PKA", "Mek", "PKC"), c("X",
"Y", "Z")]

# Longitud camí
len_flux3 <- longitud_cami_fisic(collididor3n_PKA_Mek_PKC, coords3Dn_diag_corr1)

res_flux3 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Mek_PKC,
coords3Dn_diag_corr1)
r_in3 <- as.numeric(res_flux3$r_in)
u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

p0_flux3 <- coords_collididor_PKA_Mek_PKC[ "Mek",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3
```

---

```

# _____
```

---

```

nodes_collididor_PKA_Raf_PKC <- c("PKA", "Raf", "PKC")
collididor3n_PKA_Raf_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Raf_PKC)
coords_collididor_PKA_Raf_PKC <- coords3Dn_diag_corr1[c("PKA", "Raf", "PKC"), c("X",
"Y", "Z")]

# Longitud camí
len_flux4 <- longitud_cami_fisic(collididor3n_PKA_Raf_PKC, coords3Dn_diag_corr1)
```

```

res_flux4 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Raf_PKC,
coords3Dn_diag_corr1)
r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_collididor_PKA_Raf_PKC[ "Raf",c("X","Y","Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

# _____
# nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
# cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)
# coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y",
# "Z")]

# Longitud camí
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

p0_flux <- coords_cadena3n_Raf_Mek_Erk[ "Raf",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux * u_dir

# _____
# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux, p1_flux,
                      p0_flux0, p1_flux0,
                      p0_flux1, p1_flux1,
                      p0_flux2, p1_flux2,
                      p0_flux3, p1_flux3,
                      p0_flux4, p1_flux4
                     )
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset5 <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
coords3Dn_diag_corr5 <- sweep(coords3Dn_diag_corr1, 2, offset5, "-")
p0_flux   <- p0_flux - offset5
p1_flux   <- p1_flux - offset5
p0_flux0  <- p0_flux0 - offset5
p1_flux0  <- p1_flux0 - offset5
p0_flux1  <- p0_flux1 - offset5
p1_flux1  <- p1_flux1 - offset5
p0_flux2  <- p0_flux2 - offset5
p1_flux2  <- p1_flux2 - offset5
p0_flux3  <- p0_flux3 - offset5
p1_flux3  <- p1_flux3 - offset5
p0_flux4  <- p0_flux4 - offset5

```

```

p1_flux4 <- p1_flux4 - offset5
# [2] _____


open3d()

## wgl
## 49

plot3d(coords3Dn_diag_corr5, col = "orangered", size = 1, type = "s")

edges <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retrolateral (retroflux i Lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr5[v1, ], coords3Dn_diag_corr5[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr5))
text3d(coords3Dn_diag_corr5[, "X"]+0.02,
       coords3Dn_diag_corr5[, "Y"],
       coords3Dn_diag_corr5[, "Z"]+2,
       texts = rownames(coords3Dn_diag_corr5),
       adj = c(0, 0.2),
       cex=0.9)

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr5[7, ], coords3Dn_diag_corr5[1, ]), col = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr5[1, ], coords3Dn_diag_corr5[8, ]), col = "blue", lwd = 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,

```

```

col = "red",
type = "rotation",
lwd = 2,
width = 0.3,
barrlen = 0.02
)
text3d(p1_flux1[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical 1", color = "black")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 2", color = "black")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 3", color = "black")

# segeta blava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 4", color = "black")

# segeta blava1
arrow3d(

```

```

p0 = p0_flux0,
p1 = p1_flux0,
col = "blue",
type = "rotation",
lwd = 2,
width = 0.3,
barrlen = 0.02
)
text3d(p1_flux1[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux vertical 1", color = "black")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr5[3] ,
  p1 = coords3Dn_diag_corr5[2] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 2", color = "black")

# segeta blava3
arrow3d(
  p0 = coords3Dn_diag_corr5[3] ,
  p1 = coords3Dn_diag_corr5[9] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 3", color = "black")

# segeta blava4
arrow3d(
  p0 = coords3Dn_diag_corr5[4] ,
  p1 = coords3Dn_diag_corr5[8] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 4", color = "black")

```

```

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr5[7, ], coords3Dn_diag_corr5[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr5[1, ], coords3Dn_diag_corr5[8, ]),
            col = "purple", lwd = 1.5)

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] + 1,
       texts = "FC local", color = "purple")

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr5) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)

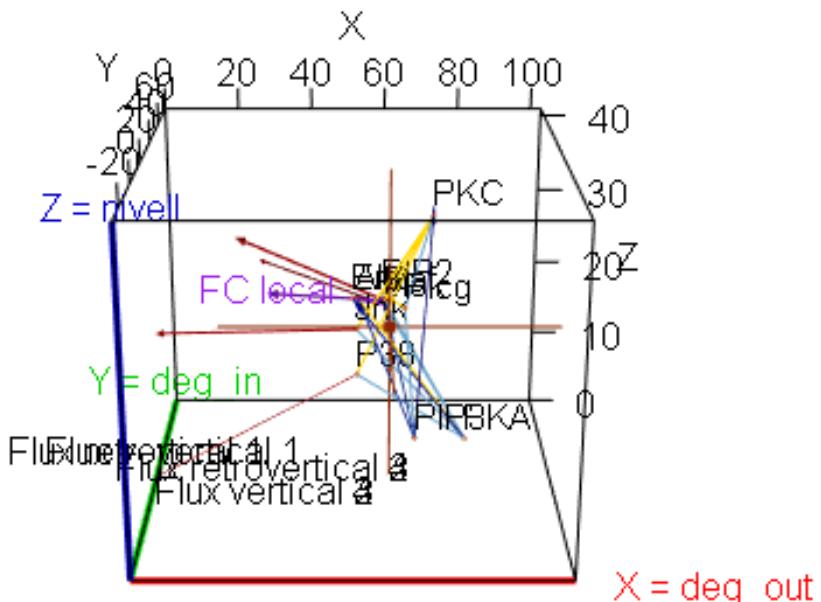
```

```

text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



## 5.8 Càcul de l'angle de flux causal vertical (Z) del model topoempíric: theta\_flux\_vertical\_corregit

```

# Flux causal local cadena
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr2)

# Fluxos verticals (1 cadena i 3 arestes verticals)
flux_vertical1 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Plcg_PKC,
coords3Dn_diag_corr2)

nodes_cadena_PIP3_PIP2 <- c("PIP3", "PIP2")
cadena3n_PIP3_PIP2 <- induced_subgraph(g_bosc, vids = nodes_cadena_PIP3_PIP2)
flux_vertical2 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_PIP2, coords3Dn_diag_corr2)

nodes_cadena_PIP3_Akt <- c("PIP3", "Akt")
cadena3n_PIP3_Akt <- induced_subgraph(g_bosc, vids = nodes_cadena_PIP3_Akt)
flux_vertical3 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Akt, coords3Dn_diag_corr2)

```

```

nodes_cadena_PKA_Erk <- c("PKA", "Erk")
cadena3n_PKA_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_PKA_Erk)
flux_vertical4 <- theta_flux_vectorial_Z_fisic(cadena3n_PKA_Erk, coords3Dn_diag_corr2)

# Fluxos retroverticals (4 col·lididors)

flux_retro_vertical1 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC,
coords3Dn_diag_corr2)
flux_retro_vertical2 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr2)
flux_retro_vertical3 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Mek_PKC,
coords3Dn_diag_corr2)
flux_retro_vertical4 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Raf_PKC,
coords3Dn_diag_corr2)

taula_fluxos_general <- function(fluxos) {
  extreu <- function(f) {

    if ("u_bar" %in% names(f$res)) {
      # vertical
      v <- f$res$u_bar
      modul <- f$res$len_u
    } else {
      # col·Lidor
      v <- f$res$r_in
      modul <- f$res$len_r
    }

    tibble::tibble(
      flux = f$nom,
      node_1 = f$nodes[1],
      node_2 = f$nodes[2],
      node_3 = ifelse(length(f$nodes) == 3, f$nodes[3], NA),
      u_bar_X = v[1],
      u_bar_Y = v[2],
      u_bar_Z = v[3],
      modul = modul,
      flux_z = f$res$flux_z,
      theta_z = f$res$theta_z,
      signe = f$res$signe
    )
  }
}

dplyr::bind_rows(lapply(fluxos, extreu))
}

fluxos <- list(
  list(
    nom = "flux_objectiu",
    res = flux_objectiu,

```

```

    nodes = c("Raf", "Mek", "Erk")
),
list(
  nom = "flux_vertical1",
  res = flux_vertical1,
  nodes = c("PIP3", "Plcg", "PKC")
),
list(
  nom = "flux_vertical2",
  res = flux_vertical2,
  nodes = c("PIP3", "PIP2")
),
list(
  nom = "flux_vertical3",
  res = flux_vertical3,
  nodes = c("PIP3", "Akt")
),
list(
  nom = "flux_vertical4",
  res = flux_vertical4,
  nodes = c("PKA", "Erk")
),

list(
  nom = "flux_retro_vertical1",
  res = flux_retro_vertical1,
  nodes = c("PKA", "P38", "PKC")
),
list(
  nom = "flux_retro_vertical2",
  res = flux_retro_vertical2,
  nodes = c("PKA", "Jnk", "PKC")
),
list(
  nom = "flux_retro_vertical3",
  res = flux_retro_vertical3,
  nodes = c("PKA", "Mek", "PKC")
),
list(
  nom = "flux_retro_vertical4",
  res = flux_retro_vertical4,
  nodes = c("PKA", "Raf", "PKC")
)
)

taula_general_fluxos <- taula_fluxos_general(fluxos)
taula_general_fluxos

## # A tibble: 9 × 11
##   flux node_1 node_2 node_3 u_bar_X u_bar_Y u_bar_Z modul flux_z theta_z signe
##   <chr> <chr> <chr> <chr>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl> <dbl>
## 1 flux... Raf     Mek     Erk      -8.02   -1.15    0.414  8.11    0.414   87.1    1
## 2 flux... PIP3    Plcg    PKC      4.49    14.5    24.7    29.0    24.7    31.5    1
## 3 flux... PIP3    PIP2    <NA>     -7.61   18.2    14.7    24.6    14.7    53.3    1

```

```

## 4 flux... PIP3    Akt     <NA>    -13.8    16.8    14.4    26.1    14.4    56.5    1
## 5 flux... PKA     Erk     <NA>    -27.2    15.5    14.5    34.5    14.5    65.2    1
## 6 flux... PKA     P38     PKC     -44.5    11.4    -13.8    48.0    -13.8    73.3    -1
## 7 flux... PKA     Jnk     PKC     -44.6    16.4    -3.35    47.7    -3.35    86.0    -1
## 8 flux... PKA     Mek     PKC     -30.7    43.9    -1.68    53.5    -1.68    88.2    -1
## 9 flux... PKA     Raf     PKC     -30.0    18.5    3.33    35.4    3.33    84.6    1

# --- Funció: calcula la mitjana vectorial global ----

mitjana_vectorial_global <- function(taula_fluxos) {
  # 1. Excloem el flux objectiu: NO entra en la mitjana
  taula2 <- taula_fluxos %>%
    dplyr::filter(flux != "flux_objectiu")

  # 2. Matriu dels vectors (u_bar_X, u_bar_Y, u_bar_Z)
  V <- as.matrix(taula2[, c("u_bar_X", "u_bar_Y", "u_bar_Z")])

  # 3. Mòduls = pesos
  w <- sqrt(rowSums(V^2))

  # 4. Normalitzem cada vector
  U_norm <- V / w

  # 5. Suma vectorial ponderada
  v_sum <- colSums(w * U_norm)

  # 6. Normalitzem la direcció resultant
  v_global <- v_sum / sqrt(sum(v_sum^2))

  # 7. Angle vertical RESULTANT (respecte Z)
  # → No s'utilitza cap node, només la direcció general
  cos_theta <- v_global[3] / sqrt(sum(v_global^2)) # component Z
  angle_graus <- acos(cos_theta) * 180 / pi

  # 8. Angle del flux objectiu (ja està a la taula!)
  angle_objectiu <- taula_fluxos$theta_z[taula_fluxos$flux == "flux_objectiu"]

  # 9. Diferència d'angles
  diferencia_angles <- abs(angle_objectiu - angle_graus)

  # 10. Retornem tot
  list(
    vectors_utilitzats = taula2$flux,
    v_sum = v_sum,
    v_global = v_global,
    angle_vertical = angle_graus,
    angle_objectiu = angle_objectiu,
    diferencia_angles = diferencia_angles
  )
}

(resultat <- mitjana_vectorial_global(taula_general_fluxos))

```

```

## $vectors_utilitzats
## [1] "flux_vertical1"      "flux_vertical2"      "flux_vertical3"
## [4] "flux_vertical4"      "flux_retro_vertical1" "flux_retro_vertical2"
## [7] "flux_retro_vertical3" "flux_retro_vertical4"
##
## $v_sum
##   u_bar_X   u_bar_Y   u_bar_Z
## -194.04447 155.14495 52.76348
##
## $v_global
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.7640067 0.6108486 0.2077444
##
## $angle_vertical
## u_bar_Z
## 78.0098
##
## $angle_objectiu
## [1] 87.0712
##
## $diferencia_angles
## u_bar_Z
## 9.061406

```

Matriu de distàncies dels punts de destinació dels vectors directors i a la diagonal la seva distància amb el centre de masses segons aquests 8 punts

```

## 0. Preparem taula de puntes dels vectors directors

# Ens quedem només amb els fluxos generals (sense objectiu)
fluxos_sense_obj <- subset(taula_general_fluxos) #, flux != "flux_objectiu")

# Coordenades dels nodes
coords <- coords3Dn_diag_corr1

# Calculem els punts finals dels vectors directors
puntes <- data.frame(
  Xend = coords[fluxos_sense_obj$node_2, "X"] + fluxos_sense_obj$u_bar_X,
  Yend = coords[fluxos_sense_obj$node_2, "Y"] + fluxos_sense_obj$u_bar_Y,
  Zend = coords[fluxos_sense_obj$node_2, "Z"] + fluxos_sense_obj$u_bar_Z,
  row.names = fluxos_sense_obj$flux
)

# 1. Puntes dels fluxos verticals i retroverticals (sense objectiu)
puntes_cm <- puntes

# 2. Centre de masses
centre <- colMeans(puntes_cm)

# 3. Distància al centre
dist_centre <- apply(puntes, 1, function(p)
  sqrt(sum((p - centre)^2)))

# 4. Matriu de distàncies

```

```

n <- nrow(puntes)
D <- matrix(0, n, n)
rownames(D) <- colnames(D) <- rownames(puntes)

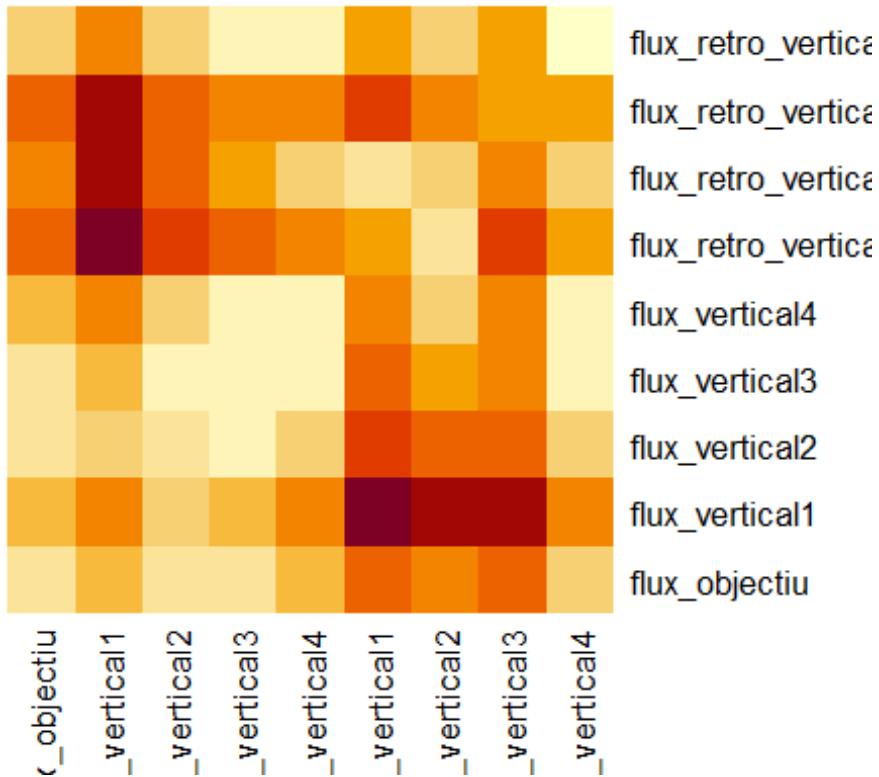
for (i in 1:n) {
  for (j in 1:n) {
    D[i, j] <- sqrt(sum((puntes[i, ] - puntes[j, ]) ^ 2))
  }
}

# 5. Substituem la diagonal per distància al centre
diag(D) <- dist_centre

# Resultat final
matriu_distancies <- D

heatmap(as.matrix(matriu_distancies), Rowv = NA, Colv = NA, scale = "none")

```



```
knitr:::kable(matriu_distancies)
```

	flux_objectiu	flux_vertical1	flux_vertical2	flux_vertical3	flux_vertical4	flux_retro_vertical1	flux_retro_vertical2	flux_retro_vertical3	flux_retro_vertical4
flux_objectiu	19.56	32.65	19.39	21.62	32.22	48.01474	44.02211	50.42988	23.73404
flux_vertical1	186	858	109	284	451				2
flux_vertical2	32.65	42.08	24.04	33.23	46.40	77.03674	69.31054	70.58422	46.56467
flux_vertical3	858	075	664	507	157				2
flux_vertical4	19.39	24.04	20.43	12.69	27.66	58.71860	49.07335	47.38882	25.07453
flux_retro_vertical1	109	664	137	367	749				2
flux_retro_vertical2	21.62	33.23	12.69	11.62	15.00	49.18087	37.90876	45.07954	14.97209

	flux_objecti u	flux_vere tical1	flux_vere tical2	flux_vere tical3	flux_vere tical4	flux_retro _vertical1	flux_retro _vertical2	flux_retro _vertical3	flux_retro _vertical4
cal3	284	507	367	388	100				8
flux_verti cal4	32.22	46.40	27.66	15.00	15.24	41.15570	27.14627	46.44092	13.29726
flux_retro _vertical1	451	157	749	100	587				5
flux_retro _vertical2	48.01	77.03	58.71	49.18	41.15	38.89949	17.33421	56.02190	35.12134
flux_retro _vertical3	474	674	860	087	570				0
flux_retro _vertical4	44.02	69.31	49.07	37.90	27.14	17.33421	28.87303	46.40274	24.07279
flux_retro _vertical1	211	054	335	876	627				9
flux_retro _vertical2	50.42	70.58	47.38	45.07	46.44	56.02190	46.40274	39.29908	38.76293
flux_retro _vertical3	988	422	882	954	092				4
flux_retro _vertical4	23.73	46.56	25.07	14.97	13.29	35.12134	24.07280	38.76293	4.971639

Hom hi afegeix la setgea del flux causal vertical (a efectes il·lustratius hom trasllada l'origen del vector director c(0,0,0) al centre de masses)

```

nodes_cadena_PIP3_Plcg_PKC <- c("PIP3", "Plcg", "PKC")
cadena3n_PIP3_Plcg_PKC <- induced_subgraph(g_bosc, vids = nodes_cadena_PIP3_Plcg_PKC)
coords_cadena3n_PIP3_Plcg_PKC <- coords3Dn_diag_corr1[c("PIP3", "Plcg", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux0 <- longitud_cami_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)

res_flux0 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)
u_bar0 <- as.numeric(res_flux0$u_bar)
u_dir0 <- u_bar0 / sqrt(sum(u_bar0^2))

p0_flux0 <- coords_cadena3n_PIP3_Plcg_PKC["PIP3",c("X","Y","Z")]
p1_flux0 <- p0_flux0 + len_flux0 * u_dir0

# _____
# Longitud camí
len_flux1 <- longitud_cami_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1)

res_flux1 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
r_in1 <- as.numeric(res_flux1$r_in)
u_dir1 <- r_in1 / sqrt(sum(r_in1^2))

p0_flux1 <- coords_collididor_PKA_P38_PKC["P38",c("X","Y","Z")]
p1_flux1 <- p0_flux1 + len_flux1 * u_dir1
#[1]

```

```

nodes_collididor_PKA_Jnk_PKC <- c("PKA", "Jnk", "PKC")
collididor3n_PKA_Jnk_PKC <- induced_subgraph(g_bosc, vids = nodes_collididor_PKA_Jnk_PKC)
coords_collididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X",
"Y", "Z")]

# Longitud camí
len_flux2 <- longitud_cami_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

res_flux2 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)
r_in2 <- as.numeric(res_flux2$r_in)
u_dir2 <- r_in2 / sqrt(sum(r_in2^2))

p0_flux2 <- coords_collididor_PKA_Jnk_PKC["Jnk",c("X","Y","Z")]
p1_flux2 <- p0_flux2 + len_flux2 * u_dir2

# _____
# Longitud camí
len_flux3 <- longitud_cami_fisic(collididor3n_PKA_Mek_PKC, coords3Dn_diag_corr1)

res_flux3 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Mek_PKC,
coords3Dn_diag_corr1)
r_in3 <- as.numeric(res_flux3$r_in)
u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

p0_flux3 <- coords_collididor_PKA_Mek_PKC["Mek",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3

# _____
# Longitud camí
len_flux4 <- longitud_cami_fisic(collididor3n_PKA_Raf_PKC, coords3Dn_diag_corr1)

res_flux4 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Raf_PKC,
coords3Dn_diag_corr1)
r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_collididor_PKA_Raf_PKC["Raf",c("X","Y","Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

```

```

# _____
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)
coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y", "Z")]

```

# Longitud camí

```

len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

p0_flux <- coords_cadena3n_Raf_Mek_Erk["Raf",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux * u_dir

```

# \_\_\_\_\_

# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu

# hom combina totes les coordenades que apareixen al gràfic

```

all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux, p1_flux,
                      p0_flux0, p1_flux0,
                      p0_flux1, p1_flux1,
                      p0_flux2, p1_flux2,
                      p0_flux3, p1_flux3,
                      p0_flux4, p1_flux4
)

```

# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000

```

offset5 <- apply(all_coords1, 2, min)

```

# hom aplica el desplaçament

```

coords3Dn_diag_corr5 <- sweep(coords3Dn_diag_corr1, 2, offset5, "-")
p0_flux <- p0_flux - offset5
p1_flux <- p1_flux - offset5
p0_flux0 <- p0_flux0 - offset5
p1_flux0 <- p1_flux0 - offset5
p0_flux1 <- p0_flux1 - offset5
p1_flux1 <- p1_flux1 - offset5
p0_flux2 <- p0_flux2 - offset5
p1_flux2 <- p1_flux2 - offset5
p0_flux3 <- p0_flux3 - offset5
p1_flux3 <- p1_flux3 - offset5
p0_flux4 <- p0_flux4 - offset5
p1_flux4 <- p1_flux4 - offset5

```

# [2]

```

open3d()

```

```

## wg1
## 52

plot3d(coords3Dn_diag_corr5, col = "orangered", size = 1, type = "s")

edges <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retrolateral (retroflux i Lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr5[v1, ], coords3Dn_diag_corr5[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr5))
text3d(coords3Dn_diag_corr5[, "X"]+0.02,
       coords3Dn_diag_corr5[, "Y"],
       coords3Dn_diag_corr5[, "Z"]+2,
       texts = rownames(coords3Dn_diag_corr5),
       adj = c(0, 0.2),
       cex=0.9)

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr5[7, ], coords3Dn_diag_corr5[1, ]), col = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr5[1, ], coords3Dn_diag_corr5[8, ]), col = "blue", lwd = 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barlen = 0.02
)
text3d(p1_flux1[1], p1_flux1[2], p1_flux1[3] + 4,

```

```

texts = "Flux retrovertical 1", color = "black")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 2", color = "black")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 3", color = "black")

# segeta bLava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 4,
       texts = "Flux retrovertical 4", color = "black")

# segeta blava1
arrow3d(
  p0 = p0_flux0,
  p1 = p1_flux0,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)

```

```

)
text3d(p1_flux1[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux vertical 1", color = "black")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr5[3,] ,
  p1 = coords3Dn_diag_corr5[2,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 2", color = "black")

# segeta blava3
arrow3d(
  p0 = coords3Dn_diag_corr5[3,] ,
  p1 = coords3Dn_diag_corr5[9,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 3", color = "black")

# segeta blava4
arrow3d(
  p0 = coords3Dn_diag_corr5[4,] ,
  p1 = coords3Dn_diag_corr5[8,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1[1], p1[2], p1[3] + 1,
       texts = "Flux vertical 4", color = "black")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr5[7, ], coords3Dn_diag_corr5[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr5[1, ], coords3Dn_diag_corr5[8, ]),
            col = "purple", lwd = 1.5)

```

```

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple1",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux[1], p1_flux[2], p1_flux[3] + 1,
       texts = "FC local", color = "purple1")

# el flux causal global
res_global <- mitjana_vectorial_global(taula_general_fluxos)

v_sum <- res_global$v_sum # components reals
v_global <- res_global$v_global # direcció normalitzada

# hom escala la segeta perquè sigui comparable amb el model
escala <- 20
p0 <- coords3Dn_diag_corr5["Erk",]
p1 <- p0 + v_global * escala # direcció

arrow3d(
  p0 = p0,
  p1 = p1,
  type = "rotation",
  col = "purple4",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)

text3d(p1,
       texts = "Flux causal vertical",
       col = "purple4")

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr5) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

```

```

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

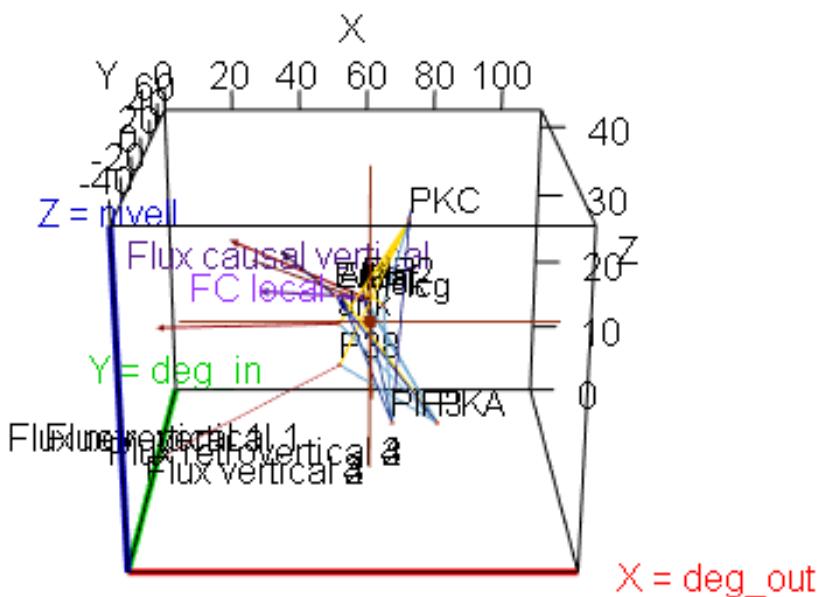
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



## 5.8 Càlcul de l'angle de flux causal holístic (ZYX) del model topoempíric: flux\_fulla\_corregit

```
collididor3n_PKA_P38_PKC <- crea_colididor_artificial(g_bosc, chain_nodes =
c("PKA", "P38", "PKC"))
collididor3n_PKA_Jnk_PKC <- crea_colididor_artificial(g_bosc, chain_nodes =
c("PKA", "Jnk", "PKC"))
collididor3n_PIP3_Akt_PKA <- crea_colididor_artificial(g_bosc, chain_nodes =
c("PIP3", "Akt", "PKA"))
collididor3n_PKA_Erk_Mek <- crea_colididor_artificial(g_bosc, chain_nodes =
c("PKA", "Erk", "Mek"))

#
# nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
# cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)
# Flux objectiu
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

# Flux fulla vertical (1 col·lididor fulla vertical)
flux1_coll_ver <- theta_flux_colididor_Z_fisic(collididor3n_PIP3_Akt_PKA,
coords3Dn_diag_corr1)

# Fluxos fulles retroverticals (2 col·lididors fulla retro-verticals)
flux2_coll_retrover <- theta_flux_colididor_Z_fisic(collididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
flux3_coll_retrover <- theta_flux_colididor_Z_fisic(collididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)

# Flux fulla laterovertical (1 col·lididor fulla latero-vertical)
flux4_coll_laterover <- theta_flux_colididor_Z_fisic(collididor3n_PKA_Erk_Mek,
coords3Dn_diag_corr1)

taula_fluxos_general <- function(fluxos) {
  extreu <- function(f) {

    if ("u_bar" %in% names(f$res)) {
      # vertical
      v <- f$res$u_bar
      modul <- f$res$len_u
    } else {
      # col·lididor
      v <- f$res$r_in
      modul <- f$res$len_r
    }

    tibble::tibble(
      flux = f$nom,
      node_1 = f$nodes[1],
      node_2 = f$nodes[2],
      node_3 = ifelse(length(f$nodes) == 3, f$nodes[3], NA),
      u_bar_X = v[1],
      u_bar_Y = v[2],
      u_bar_Z = v[3],
      modul = modul,
    )
  }
}
```

```

    flux_z = f$res$flux_z,
    theta_z = f$res$theta_z,
    signe = f$res$signe
  )
}

dplyr::bind_rows(lapply(fluxos, extreu))
}

fluxos1 <- list(
  list(
    nom = "flux_objectiu",
    res = flux_objectiu,
    nodes = c("Raf", "Mek", "Erk")
  ),
  list(
    nom = "flux1_coll_ver",
    res = flux1_coll_ver,
    nodes = c("PIP3", "Akt", "PKA")
  ),
  list(
    nom = "flux2_coll_retrover",
    res = flux2_coll_retrover,
    nodes = c("PKA", "P38", "PKC")
  ),
  list(
    nom = "flux3_coll_retrover",
    res = flux3_coll_retrover,
    nodes = c("PKA", "Jnk", "PKC")
  ),
  list(
    nom = "flux4_coll_laterover",
    res = flux4_coll_laterover,
    nodes = c("PKA", "Erk", "Mek")
  )
)

# --- Funció: calcula la mitjana vectorial global -----
mitjana_vectorial_global <- function(taula_fluxos) {
  # 1. Excloem el flux objectiu: NO entra en la mitjana
  taula2 <- taula_fluxos %>%
    dplyr::filter(flux != "flux_objectiu")

  # 2. Matriu dels vectors ( $\bar{u}_X$ ,  $\bar{u}_Y$ ,  $\bar{u}_Z$ )
  V <- as.matrix(taula2[, c("u_bar_X", "u_bar_Y", "u_bar_Z")])

  # 3. Mòduls = pesos
  w <- sqrt(rowSums(V^2))

  # 4. Normalitzem cada vector
  U_norm <- V / w
}

```

```

# 5. Suma vectorial ponderada
v_sum <- colSums(w * U_norm)

# 6. Normalitzem la direcció resultant
v_global <- v_sum / sqrt(sum(v_sum^2))

# 7. Angle vertical RESULTANT (respecte Z)
# → No s'utilitza cap node, només la direcció general
cos_theta <- v_global[3] / sqrt(sum(v_global^2)) # component Z
angle_graus <- acos(cos_theta) * 180 / pi

# 8. Angle del flux objectiu (ja està a la taula!)
angle_objectiu <- taula_fluxos$theta_z[taula_fluxos$flux == "flux_objectiu"]

# 9. Diferència d'angles
diferencia_angles <- abs(angle_objectiu - angle_graus)

# 10. Retornem tot
list(
  vectors_utilitzats = taula2$flux,
  v_sum = v_sum,
  v_global = v_global,
  angle_vertical = angle_graus,
  angle_objectiu = angle_objectiu,
  diferencia_angles = diferencia_angles
)
}

## ---- Diferències angulars detallades ----
taula_general_fluxos_fulla <- taula_fluxos_general(fluxos1)
resultat1 <- mitjana_vectorial_global(taula_general_fluxos_fulla)

v_obj <- as.numeric(t(taula_general_fluxos_fulla[taula_general_fluxos_fulla$flux == "flux_objectiu", c("u_bar_X", "u_bar_Y", "u_bar_Z")])) 

# normalitza vectors
ng <- resultat1$v_global / sqrt(sum(resultat1$v_global^2))
no <- v_obj / sqrt(sum(v_obj^2))

# angles per eix
ang_X <- acos(ng[1] * no[1]) * 180 / pi
ang_Y <- acos(ng[2] * no[2]) * 180 / pi
ang_Z <- acos(ng[3] * no[3]) * 180 / pi

# angle total 3D
ang_total <- acos(sum(ng * no)) * 180 / pi

resultat_angles <- list(
  angle_X = ang_X,
  angle_Y = ang_Y,
  angle_Z = ang_Z,
  angle_total_3D = ang_total
)

```

```

)
resultat_angles

## $angle_X
## u_bar_X
## 24.55658
##
## $angle_Y
## u_bar_Y
## 92.91845
##
## $angle_Z
## u_bar_Z
## 89.53264
##
## $angle_total_3D
## [1] 29.91185

## ---- Diferències angulars detallades ----
taula_general_fluxos_fulla <- taula_fluxos_general(fluxos1)
resultat1 <- mitjana_vectorial_global(taula_general_fluxos_fulla)

```

## 5.8.1 FLUX CAUSAL holístic del model geomètric topoempíric general: FLUX CAUSAL ÚNIC EN L'ESFERA DE DIRECCIONS

Cada model causal geomètric té un únic flux causal holístic perquè hom treballa amb les 3 dimensions de manera òptima gràcies als nodes fulla que sintetitzen alhora màxima verticalitat, màxim graus\_IN i mínim graus\_OUT.

Hi han 4 col·lididors fulla: 1 vertical, 1 laterovertical i 2 retroverticals

```

# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:
# el p1_flux té valors negatius i cal fer el recalc de les coordenades abans de fer el plot3d
#nodes_colididor_PKA_P38_PKC <- c("PKA", "P38", "PKC")
collididor3n_PKA_P38_PKC <- crea_colididor_artificial(g_bosc, chain_nodes = c("PKA", "P38", "PKC"))
coords_colididor_PKA_P38_PKC <- coords3Dn_diag_corr1[c("PKA", "P38", "PKC"), c("X", "Y", "Z")]

flux2_coll_retrover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
res_flux1 <- flux2_coll_retrover
# Longitud camí
len_flux1 <- longitud_cami_fisic(colididor3n_PKA_P38_PKC, coords3Dn_diag_corr1)

r_in1 <- as.numeric(res_flux1$r_in)
u_dir1 <- r_in1 / sqrt(sum(r_in1^2))

p0_flux1 <- coords_colididor_PKA_P38_PKC["P38",c("X","Y","Z")]
p1_flux1 <- p0_flux1 + len_flux1 * u_dir1
#[1]

```

```

#nodes_colididor_PKA_Jnk_PKC <- c("PKA", "Jnk", "PKC")
collididor3n_PKA_Jnk_PKC <- crea_colididor_artificial(g_bosc, chain_nodes = c("PKA",
"Jnk", "PKC"))
coords_colididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X",
"Y", "Z")]

flux3_coll_retrover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)
res_flux2 <- flux3_coll_retrover
# Longitud camí
len_flux2 <- longitud_cami_fisic(colididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

r_in2 <- as.numeric(res_flux2$r_in)
u_dir2 <- r_in2 / sqrt(sum(r_in2^2))

p0_flux2 <- coords_colididor_PKA_Jnk_PKC["Jnk",c("X","Y","Z")]
p1_flux2 <- p0_flux2 + len_flux2 * u_dir2

# _____
#nodes_colididor_PKA_Erk_Mek <- c("PKA", "Erk", "Mek")
collididor3n_PKA_Erk_Mek <- crea_colididor_artificial(g_bosc, chain_nodes = c("PKA",
"Erk", "Mek"))

coords_colididor_PKA_Erk_Mek <- coords3Dn_diag_corr1[c("PKA", "Erk", "Mek"), c("X",
"Y", "Z")]

flux4_coll_laterover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_Erk_Mek,
coords3Dn_diag_corr1)
res_flux3 <- flux4_coll_laterover
# Longitud camí
len_flux3 <- longitud_cami_fisic(colididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1)

r_in3 <- as.numeric(res_flux3$r_in)
u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

p0_flux3 <- coords_colididor_PKA_Erk_Mek["Erk",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3

# _____
#nodes_colididor_PIP3_Akt_PKA <- c("PIP3", "Akt", "PKA")
collididor3n_PIP3_Akt_PKA <- crea_colididor_artificial(g_bosc, chain_nodes = c("PIP3",
"Akt", "PKA"))

coords_colididor_PIP3_Akt_PKA <- coords3Dn_diag_corr1[c("PIP3", "Akt", "PKA"), c("X",
"Y", "Z")]

flux1_coll_ver <- theta_flux_colididor_Z_fisic(colididor3n_PIP3_Akt_PKA,
coords3Dn_diag_corr1)
res_flux4 <- flux1_coll_ver
# Longitud camí

```

```

len_flux4 <- longitud_cami_fisic(collididor3n_PIP3_Akt_PKA, coords3Dn_diag_corr1)

r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_collididor_PIP3_Akt_PKA["Akt",c("X","Y","Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

#
#cadena3n_Raf_Mek_Erk <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk,
coords3Dn_diag_corr1)
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)

# Longitud camí
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- flux_objectiu
u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y",
"Z")]
p0_flux <- coords_cadena3n_Raf_Mek_Erk["Raf",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux * u_dir
#
# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux, p1_flux,
                      p0_flux1, p1_flux1,
                      p0_flux2, p1_flux2,
                      p0_flux3, p1_flux3,
                      p0_flux4, p1_flux4)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset6 <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
coords3Dn_diag_corr6 <- sweep(coords3Dn_diag_corr1, 2, offset6, "-")
p0_flux <- p0_flux - offset6
p1_flux <- p1_flux - offset6
p0_flux1 <- p0_flux1 - offset6

```

```

p1_flux1 <- p1_flux1 - offset6
p0_flux2 <- p0_flux2 - offset6
p1_flux2 <- p1_flux2 - offset6
p0_flux3 <- p0_flux3 - offset6
p1_flux3 <- p1_flux3 - offset6
p0_flux4 <- p0_flux4 - offset6
p1_flux4 <- p1_flux4 - offset6

# [2] _____
```

**open3d()**

```

## wgl
## 55

plot3d(coords3Dn_diag_corr6, col = "orangered", size = 1, type = "s")

edges <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retroLateral (retroflux i Lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr6[v1, ], coords3Dn_diag_corr6[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr6))
text3d(coords3Dn_diag_corr6[, "X"],
       coords3Dn_diag_corr6[, "Y"],
       coords3Dn_diag_corr6[, "Z"],
       texts = rownames(coords3Dn_diag_corr6),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr6[7, ], coords3Dn_diag_corr6[1, ]), col = "blue", lwd
```

```

= 1.5)
#segments3d(rbind(coords3Dn_diag_corr6[1, ], coords3Dn_diag_corr6[8, ]), col = "blue", lwd
= 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux1,#[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical ", color = "green4",cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux2,,
       texts = "Flux retrovertical ", color = "green4",cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux3,
       texts = "Flux laterovertical ", color = "green4",cex = 0.75,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "green",

```

```

type = "rotation",
lwd = 2,
width = 0.3,
barrlen = 0.02
)
text3d(p1_flux4,
       texts = "Flux vertical", color = "green4",cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr6[7, ], coords3Dn_diag_corr6[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr6[1, ], coords3Dn_diag_corr6[8, ]),
            col = "purple", lwd = 1.5)

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux, #[1], p1_flux[2], p1_flux[3] + 4,
       texts = "Flux causal local",
       color = "purple4",
       cex = 0.75,
       depth_test="always")

# el flux causal global
res_holistic <- mitjana_vectorial_global(taula_general_fluxos_fulla)

v_sum <- res_holistic$v_sum # components reals
v_holistic <- res_holistic$v_global # direcció normalitzada

# hom escala la segeta perquè sigui comparable amb el model
escala <- 20
p0 <- coords3Dn_diag_corr6["Erk",]
p1 <- p0 + v_holistic * escala # direcció

arrow3d(
  p0 = p0,
  p1 = p1,
  type = "rotation",
  col = "red",
  lwd = 3,
  width = 0.3,
  barrlen = 0.02
)

```

```

text3d(p1,
  texts = "Flux causal holistic",
  col = "darkred",
  cex = 0.75,
  depth_test="always")

## CON CAUSAL HOLÍSTIC 3D

library(rgl)

# Paràmetres del con
angle <- resultat_angles$angle_total_3D * pi / 180 # en radians
h <- 40 # alçada del con (ajustable)
rad <- h * tan(angle) # radi = h*tan(θ)

# Base d'un con alineat amb l'eix Z
n <- 60
theta <- seq(0, 2 * pi, length.out = n)

x1 <- rad * cos(theta)
y1 <- rad * sin(theta)
z1 <- rep(h, n)

# 1) hom gira el con perquè apunti com el flux objectiu
v_obj <- v_obj/sqrt(sum(v_obj^2)) # vector normalitzat objectiu que ja és calculat

# Vector unitari Z
z_axis <- c(0, 0, 1)

# Producte vectorial per trobar eix de rotació
rot_angle <- acos(sum(v_obj * z_axis))

# Matriu de rotació (Rodrigues)
if (rot_angle != 0) {
  rot_axis <- pracma::cross(z_axis, v_obj)
  rot_axis <- rot_axis / sqrt(sum(rot_axis^2))

  K <- matrix(c(0,-rot_axis[3],rot_axis[2],rot_axis[3],
               0,-rot_axis[1],-rot_axis[2],rot_axis[1],0),3,3, byrow = TRUE)

  R <- diag(3) + sin(rot_angle) * K + (1 - cos(rot_angle)) * (K %*% K)
} else {
  R <- diag(3)
}

cone_pts <- t(R %*% rbind(x1, y1, z1))

# baricentre nodes fulla θ-

```

```

origins_fulles <- rbind(p0_flux1, p0_flux2, p0_flux3, p0_flux4)
origin <- colMeans(origins_fulles)

cone_pts <- sweep(cone_pts, 2, origin, "+")

# 3) hom renderitza el con
triangles <- c()
for (i in 1:(n - 1))
  triangles <- rbind(triangles, cbind(i, i + 1, n + 1))
cone_mesh <- tmesh3d(
  vertices = t(rbind(cone_pts, origin)),
  indices = t(triangles),
  homogeneous = FALSE
)
shade3d(cone_mesh, alpha=0.18, col="purple") # <<< Con causal holístic!
text3d(origin, texts="Con causal holistic", col="purple", cex=0.9)

```

```
aspect3d(1,1,1)
```

```

centre <- colMeans(coords3Dn_diag_corr6) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

```

```

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

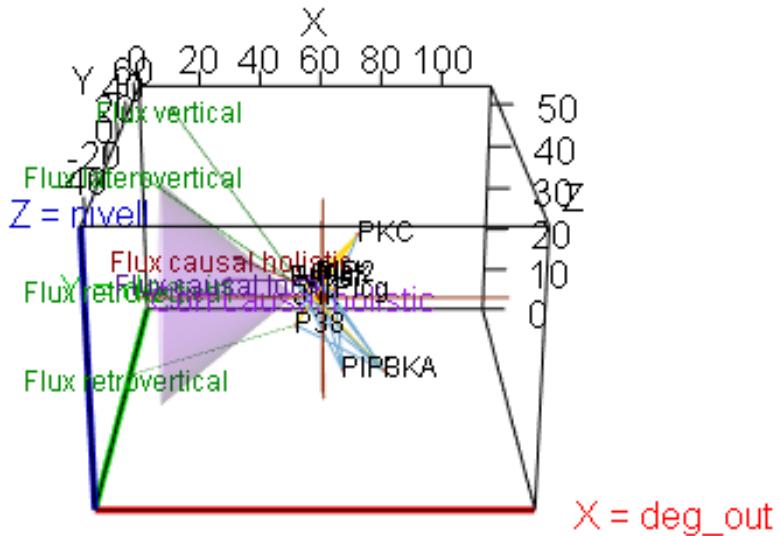
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

```

```
## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead
```



### 5.8.2 Càlcul de les coordenades del flux causal holístic ( $Z-Y-X$ ) del model topoempíric: `flux_fulla_corregit`

# bloc de codi fins a [1] per tal de dibuixar el vector director al `plot3d` sense valors negatius:

# el `p1_flux` té valors negatius i cal fer el recalc dels valors abans de fer el `plot3d`

```
#nodes_colididor_PKA_P38_PKC <- c("PKA", "P38", "PKC")
collididor3n_PKA_P38_PKC <- crea_colididor_artificial(g_bosc, chain_nodes = c("PKA",
"P38", "PKC"))
coords_colididor_PKA_P38_PKC <- coords3Dn_diag_corr1[c("PKA", "P38", "PKC"), c("X",
"Y", "Z")]
```

```
flux2_coll_retrover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
res_flux1 <- flux2_coll_retrover
# Longitud camí
len_flux1 <- longitud_cami_fisic(colididor3n_PKA_P38_PKC, coords3Dn_diag_corr1)
```

```
r_in1 <- as.numeric(res_flux1$r_in)
u_dir1 <- r_in1 / sqrt(sum(r_in1^2))

p0_flux1 <- coords_colididor_PKA_P38_PKC["P38",c("X","Y","Z")]
```

```

p1_flux1 <- p0_flux1 + len_flux1 * u_dir1
#[1] _____
```

---

```

#nodes_colidador_PKA_Jnk_PKC <- c("PKA", "Jnk", "PKC")
collididor3n_PKA_Jnk_PKC <- crea_colidador_artificial(g_bosc, chain_nodes = c("PKA",
"Jnk", "PKC"))
coords_colidador_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X",
"Y", "Z")]

flux3_coll_retrover <- theta_flux_colidador_Z_fisic(colididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)
res_flux2 <- flux3_coll_retrover
# Longitud camí
len_flux2 <- longitud_cami_fisic(colididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

r_in2 <- as.numeric(res_flux2$r_in)
u_dir2 <- r_in2 / sqrt(sum(r_in2^2))

p0_flux2 <- coords_colidador_PKA_Jnk_PKC["Jnk",c("X","Y","Z")]
p1_flux2 <- p0_flux2 + len_flux2 * u_dir2
```

---

```

# _____
```

---

```

#nodes_colidador_PKA_Erk_Mek <- c("PKA", "Erk", "Mek")
collididor3n_PKA_Erk_Mek <- crea_colidador_artificial(g_bosc, chain_nodes = c("PKA",
"Erk", "Mek"))

coords_colidador_PKA_Erk_Mek <- coords3Dn_diag_corr1[c("PKA", "Erk", "Mek"), c("X",
"Y", "Z")]

flux4_coll_laterover <- theta_flux_colidador_Z_fisic(colididor3n_PKA_Erk_Mek,
coords3Dn_diag_corr1)
res_flux3 <- flux4_coll_laterover
# Longitud camí
len_flux3 <- longitud_cami_fisic(colididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1)

r_in3 <- as.numeric(res_flux3$r_in)
u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

p0_flux3 <- coords_colidador_PKA_Erk_Mek["Erk",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3
```

---

```

# _____
```

---

```

#nodes_colidador_PIP3_Akt_PKA <- c("PIP3", "Akt", "PKA")
collididor3n_PIP3_Akt_PKA <- crea_colidador_artificial(g_bosc, chain_nodes = c("PIP3",
"Akt", "PKA"))

coords_colidador_PIP3_Akt_PKA <- coords3Dn_diag_corr1[c("PIP3", "Akt", "PKA"), c("X",
"Y", "Z")]

flux1_coll_ver <- theta_flux_colidador_Z_fisic(colididor3n_PIP3_Akt_PKA,
coords3Dn_diag_corr1)
```

```

res_flux4 <- flux1_coll_ver
# Longitud camí
len_flux4 <- longitud_cami_fisic(collididor3n_PIP3_Akt_PKA, coords3Dn_diag_corr1)

r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_collididor_PIP3_Akt_PKA["Akt", c("X", "Y", "Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

#
#cadena3n_Raf_Mek_Erk <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk,
#coords3Dn_diag_corr1)
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)

# Longitud camí
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- flux_objectiu
u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y",
"Z")]
p0_flux <- coords_cadena3n_Raf_Mek_Erk["Raf", c("X", "Y", "Z")]
p1_flux <- p0_flux + len_flux * u_dir
#
#

```

```

# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux, p1_flux,
                      p0_flux1, p1_flux1,
                      p0_flux2, p1_flux2,
                      p0_flux3, p1_flux3,
                      p0_flux4, p1_flux4)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset6 <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
coords3Dn_diag_corr6 <- sweep(coords3Dn_diag_corr1, 2, offset6, "-")
p0_flux <- p0_flux - offset6
p1_flux <- p1_flux - offset6

```

```

p0_flux1 <- p0_flux1 - offset6
p1_flux1 <- p1_flux1 - offset6
p0_flux2 <- p0_flux2 - offset6
p1_flux2 <- p1_flux2 - offset6
p0_flux3 <- p0_flux3 - offset6
p1_flux3 <- p1_flux3 - offset6
p0_flux4 <- p0_flux4 - offset6
p1_flux4 <- p1_flux4 - offset6

```

# [2]

---

```

open3d()

## wgl
## 58

plot3d(coords3Dn_diag_corr6, col = "orangered", size = 1, type = "s")

edges1 <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retrolateral (retroflux i lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr6[v1, ], coords3Dn_diag_corr6[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr6))
text3d(coords3Dn_diag_corr6[, "X"],
       coords3Dn_diag_corr6[, "Y"],
       coords3Dn_diag_corr6[, "Z"],
       texts = rownames(coords3Dn_diag_corr6),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]

```

```

#segments3d(rbind(coords3Dn_diag_corr6[7, ], coords3Dn_diag_corr6[1, ]),col = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr6[1, ], coords3Dn_diag_corr6[8, ]),col = "blue", lwd = 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux1,#[1], p1_flux1[2], p1_flux1[3] + 4,
  texts = "Flux retrovertical ", color = "green4",cex = 0.75,
  depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux2,,
  texts = "Flux retrovertical ", color = "green4",cex = 0.75,
  depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux3,
  texts = "Flux laterovertical ", color = "green4",cex = 0.75,
  depth_test="always")

# segeta blava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,

```

```

    col = "green",
    type = "rotation",
    lwd = 2,
    width = 0.3,
    barblen = 0.02
)
text3d(p1_flux4,
       texts = "Flux vertical", color = "green4",cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr6[7, ], coords3Dn_diag_corr6[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr6[1, ], coords3Dn_diag_corr6[8, ]),
            col = "purple", lwd = 1.5)

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux, #[1], p1_flux[2], p1_flux[3] + 4,
       texts = "Flux causal local",
       color = "purple4",
       cex = 0.75,
       depth_test="always")

# el flux causal global
res_holistic <- mitjana_vectorial_global(taula_general_fluxos_fulla)

v_sum <- res_holistic$v_sum # components reals
v_holistic <- res_holistic$v_global # direcció normalitzada

# hom escala la segeta perquè sigui comparable amb el model
escala <- 40

nodes_arrel <-c("PIP3", "PKA")
p0 <- colMeans(coords3Dn_diag_corr6[nodes_arrel, , drop = FALSE])
#p0 <- coords3Dn_diag_corr5["Erk",]
p1 <- p0 + v_holistic * escala # direcció

arrow3d(
  p0 = p0,
  p1 = p1,
  type = "rotation",
  col = "red",
  lwd = 3,

```

```

width = 0.3,
barblen = 0.02
)

text3d(p1,
  texts = "Flux causal holistic",
  col = "darkred",
  cex = 0.75,
  depth_test="always")

## VECTOR HOLÍSTIC
v_h <- resultat1$v_global
v_h_norm <- v_h / sqrt(sum(v_h^2))# 1. Vector holistic normalitzat

## ANGLES DE LES 4 FULLES RESPECTE AL VECTOR HOLÍSTIC
# 2. Extreiem els vectors de les 4 fulles
fulles <- taula_general_fluxos_fulla %>%
  filter(flux != "flux_objectiu") %>%
  select(u_bar_X, u_bar_Y, u_bar_Z) %>%
  as.matrix()

fulles_norm <- fulles / sqrt(rowSums(fulles^2))

angles <- acos(fulles_norm %*% v_h_norm) * 180/pi
theta_con <- max(angles)# 5. Amplada del con = màxima variació angular

theta_con# aquest ÉS l'angle correcte del con causal
## [1] 26.73612

## CON CAUSAL HOLISTIC 3D
library(rgl)

# paràmetres del con
angle <- theta_con * pi / 180 # en radians
h <- 40 # alçada del con (ajustable)
rad <- h * tan(angle) # radi base

# base d'un con alineat amb l'eix Z
n <- 60
theta <- seq(0, 2 * pi, length.out = n)
x <- rad * cos(theta)
y <- rad * sin(theta)
z <- rep(h, n)

# hom gira el con perquè apunyi com el flux objectiu
v_obj <- v_h_norm # <<<< el vector direcció normalitzat

```

```

z_axis <- c(0, 0, 1) # vector unitari Z

# Producte vectorial per trobar eix de rotació
rot_axis <- pracma::cross(z_axis, v_obj)
rot_angle <- acos(sum(z_axis * v_obj))

# Matriu de rotació (Rodrigues)
if (sqrt(sum(rot_axis^2)) > 0) {
  k <- rot_axis / sqrt(sum(rot_axis^2))

  K <- matrix(c(0, -k[3], k[2], k[3], 0, -k[1], -k[2], k[1], 0),
               nrow = 3,
               byrow = TRUE)

  R <- diag(3) * cos(rot_angle) +
    sin(rot_angle) * K +
    (1 - cos(rot_angle)) * (k %*% t(k))
} else {
  R <- diag(3)
}

# generem el con en format 3xN
cone_base <- rbind(x, y, z) # 3xN (base)
vertex <- matrix(c(0,0,0), nrow = 3, ncol = 1) # 3x1

# Apliquem la rotació correcta
cone_rot <- R %*% cone_base

# ara el con apunta EXACTAMENT en la direcció del flux holístic

# hom desplaça el con al punt de mitjanes nodes arrels
nodes_arrel <- c("PIP3", "PKA")
origin <- colMeans(coords3Dn_diag_corr6[nodes_arrel, , drop = FALSE])

cone_pts <- sweep(t(cone_rot), 2, origin, "+")

# hom renderitza el con
triangles <- c()
for (i in 1:(n - 1)) {
  triangles <- rbind(triangles, cbind(i, i + 1, n + 1))
}

cone_mesh <- tmesh3d(
  vertices = t(rbind(cone_pts, origin)),
  indices = t(triangles),
  homogeneous = FALSE
)

shade3d(cone_mesh, alpha = 0.18, col = "red")
text3d(origin, texts = "Con causal holistic", col = "red", cex = 0.9)

```

```

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr6) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

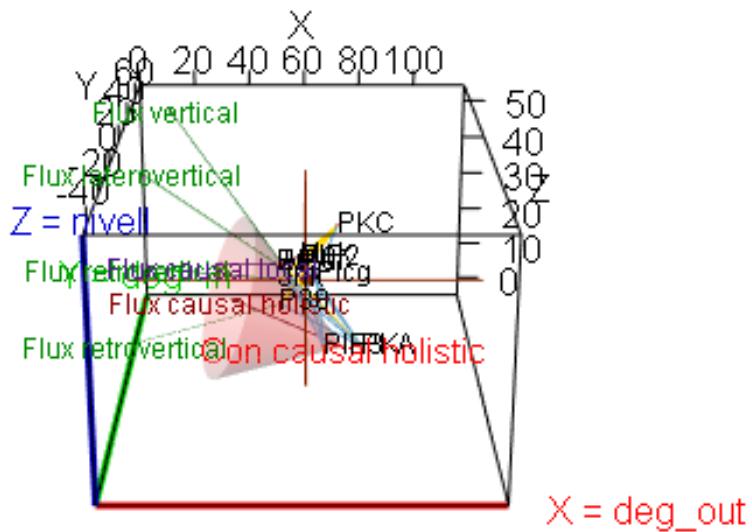
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



```

diag_con_flux <- list()

#vector director holistic
v_h <- resultat1$v_global
modul_vh <- sqrt(sum(v_h^2))
v_h_norm <- v_h / modul_vh

diag_con_flux$flux_holistic <- list(vector = v_h,
                                         modul = modul_vh,
                                         normalitzat = v_h_norm)

#dos angles en 3D
# elevació (respecte pla XY)
theta_vertical <- acos(v_h_norm[3]) * 180 / pi

# azimut (al pla XY)
phi_azimut <- atan2(v_h_norm[2], v_h_norm[1]) * 180 / pi

diag_con_flux$angles_holistic <- list(elevacio_Z = theta_vertical, azimut_XY =
phi_azimut)

#angles de les 4 fulles
fulles <- taula_general_fluxos_fulla %>%
  filter(flux != "flux_objectiu") %>%
  select(u_bar_X, u_bar_Y, u_bar_Z) %>%
  as.matrix()

fulles_norm <- fulles / sqrt(rowSums(fulles^2))

```

```

angles_fulles <- acos(fulles_norm %*% v_h_norm) * 180 / pi

diag_con_flux$angles_fulles <- angles_fulles
names(diag_con_flux$angles_fulles) <-
taula_general_fluxos_fulla$flux[taula_general_fluxos_fulla$flux != "flux_objectiu"]

# obertura del con
theta_con <- max(angles_fulles)

diag_con_flux$angle_con <- theta_con

#dimensions del con
h <- 40 # alçada del rang dels eixos
rad <- h * tan(theta_con * pi / 180) # radi base

diag_con_flux$dim_con <- list(alcada = h,
                                radi_base = rad,
                                angle_obertura = theta_con)

## distància angular entre flux holístic i flux objectiu
v_obj <- as.numeric(t(taula_general_fluxos_fulla[taula_general_fluxos_fulla$flux ==
"flux_objectiu", c("u_bar_X", "u_bar_Y", "u_bar_Z")])))

v_obj_norm <- v_obj / sqrt(sum(v_obj^2))

angle_obj_hol <- acos(sum(v_obj_norm * v_h_norm)) * 180 / pi

diag_con_flux$angle_flux_objectiu_vs_holistic <- angle_obj_hol

diag_table <- tibble::tibble(
  Parametre = c(
    "Mòdul vector holístic",
    "Elevació holística (Z)",
    "Azimut holístic (XY)",
    "Angles fulles (mitjana)",
    "Angle màxim fulles ( $\theta_{\text{con}}$ )",
    "Radi del con (h=40)",
    "Angle flux_objectiu vs holístic"
  ),
  Valor = c(
    modul_vh,
    theta_vertical,
    phi_azimut,
    mean(angles_fulles),
    theta_con,
    rad,
    angle_obj_hol
  )
)

diag_con_flux$taula_resum <- diag_table

```

```

diag_con_flux

## $flux_holistic
## $flux_holistic$vector
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.9201414  0.3575679  0.1596404
##
## $flux_holistic$modul
## [1] 1
##
## $flux_holistic$normalitzat
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.9201414  0.3575679  0.1596404
##
## $angles_holistic
## $angles_holistic$elevacio_Z
##   u_bar_Z
## 80.81398
##
## $angles_holistic$azimut_XY
##   u_bar_Y
## 158.7638
##
## $angles_fulles
##   [,1]
## [1,] 26.61945
## [2,] 26.73612
## [3,] 13.25770
## [4,] 24.38631
## attr(,"names")
## [1] "flux1_coll_ver"      "flux2_coll_retrover"  "flux3_coll_retrover"
## [4] "flux4_coll_laterover"
##
## $angle_con
## [1] 26.73612
##
## $dim_con
## $dim_con$alcada
## [1] 40
##
## $dim_con$radi_base
## [1] 20.14951
##
## $dim_con$angle_obertura
## [1] 26.73612
##
## $angle_flux_objectiu_vs_holistic
## [1] 29.91185
##
## $taula_resum
## # A tibble: 7 × 2
##   Parametre          Valor

```

```

##  <chr>                <dbl>
## 1 Mòdul vector holístic      1
## 2 Elevació holística (Z)    80.8
## 3 Azimut holístic (XY)     159.
## 4 Angles fulles (mitjana)   22.7
## 5 Angle màxim fulles ( $\theta_{\text{con}}$ ) 26.7
## 6 Radi del con ( $h=40$ )     20.1
## 7 Angle flux_objectiu vs holístic 29.9

```

Amb la bisectriu dels 4 vectors col·lididors fulla com a origen del vector director del flux causal holístic

```

# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense valors negatius:
# el p1_flux té valors negatius i cal fer el recalcül de les coordenades abans de fer el plot3d
#nodes_collididor_PKA_P38_PKC <- c("PKA", "P38", "PKC")
collididor3n_PKA_P38_PKC <- crea_collididor_artificial(g_bosc, chain_nodes = c("PKA", "P38", "PKC"))
coords_collididor_PKA_P38_PKC <- coords3Dn_diag_corr1[c("PKA", "P38", "PKC"), c("X", "Y", "Z")]

flux2_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC,
coords3Dn_diag_corr1)
res_flux1 <- flux2_coll_retrover
# longitud camí
len_flux1 <- longitud_cami_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1)

r_in1 <- as.numeric(res_flux1$r_in)
u_dir1 <- r_in1 / sqrt(sum(r_in1^2))

p0_flux1 <- coords_collididor_PKA_P38_PKC["P38",c("X","Y","Z")]
p1_flux1 <- p0_flux1 + len_flux1 * u_dir1
#[1] _____

#nodes_collididor_PKA_Jnk_PKC <- c("PKA", "Jnk", "PKC")
collididor3n_PKA_Jnk_PKC <- crea_collididor_artificial(g_bosc, chain_nodes = c("PKA", "Jnk", "PKC"))
coords_collididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X", "Y", "Z")]

flux3_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC,
coords3Dn_diag_corr1)
res_flux2 <- flux3_coll_retrover
# longitud camí
len_flux2 <- longitud_cami_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

r_in2 <- as.numeric(res_flux2$r_in)
u_dir2 <- r_in2 / sqrt(sum(r_in2^2))

p0_flux2 <- coords_collididor_PKA_Jnk_PKC["Jnk",c("X","Y","Z")]
p1_flux2 <- p0_flux2 + len_flux2 * u_dir2

```

```

#_____
#nodes_colididor_PKA_Erk_Mek <- c("PKA", "Erk", "Mek")
collididor3n_PKA_Erk_Mek <- crea_colididor_artificial(g_bosc, chain_nodes = c("PKA",
"Erk", "Mek"))

coords_colididor_PKA_Erk_Mek <- coords3Dn_diag_corr1[c("PKA", "Erk", "Mek"), c("X",
"Y", "Z")]

flux4_coll_laterover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_Erk_Mek,
coords3Dn_diag_corr1)
res_flux3 <- flux4_coll_laterover
# longitud camí
len_flux3 <- longitud_cami_fisic(colididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1)

r_in3 <- as.numeric(res_flux3$r_in)
u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

p0_flux3 <- coords_colididor_PKA_Erk_Mek["Erk",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3

#_____
#nodes_colididor_PIP3_Akt_PKA <- c("PIP3", "Akt", "PKA")
collididor3n_PIP3_Akt_PKA <- crea_colididor_artificial(g_bosc, chain_nodes = c("PIP3",
"Akt", "PKA"))

coords_colididor_PIP3_Akt_PKA <- coords3Dn_diag_corr1[c("PIP3", "Akt", "PKA"), c("X",
"Y", "Z")]

flux1_coll_ver <- theta_flux_colididor_Z_fisic(colididor3n_PIP3_Akt_PKA,
coords3Dn_diag_corr1)
res_flux4 <- flux1_coll_ver
# longitud camí
len_flux4 <- longitud_cami_fisic(colididor3n_PIP3_Akt_PKA, coords3Dn_diag_corr1)

r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_colididor_PIP3_Akt_PKA["Akt",c("X","Y","Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

#_____
#cadena3n_Raf_Mek_Erk <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk,
coords3Dn_diag_corr1)
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc, vids = nodes_cadena_Raf_Mek_Erk)

# longitud camí
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

res_flux <- flux_objectiu
```

```

u_bar <- as.numeric(res_flux$u_bar)
u_dir <- u_bar / sqrt(sum(u_bar^2))

coords_cadena3n_Raf_Mek_Erk <- coords3Dn_diag_corr1[c("Raf", "Mek", "Erk"), c("X", "Y",
"Z")]
p0_flux <- coords_cadena3n_Raf_Mek_Erk["Raf",c("X","Y","Z")]
p1_flux <- p0_flux + len_flux * u_dir
#
# _____
# bloc de codi fins a [2] per tal de desplaçar el model a un espai positiu
# hom combina totes les coordenades que apareixen al gràfic
all_coords1 <- rbind(coords3Dn_diag_corr1,
                      p0_flux, p1_flux,
                      p0_flux1, p1_flux1,
                      p0_flux2, p1_flux2,
                      p0_flux3, p1_flux3,
                      p0_flux4, p1_flux4)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
offset6 <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
coords3Dn_diag_corr6 <- sweep(coords3Dn_diag_corr1, 2, offset6, "-")
p0_flux <- p0_flux - offset6
p1_flux <- p1_flux - offset6
p0_flux1 <- p0_flux1 - offset6
p1_flux1 <- p1_flux1 - offset6
p0_flux2 <- p0_flux2 - offset6
p1_flux2 <- p1_flux2 - offset6
p0_flux3 <- p0_flux3 - offset6
p1_flux3 <- p1_flux3 - offset6
p0_flux4 <- p0_flux4 - offset6
p1_flux4 <- p1_flux4 - offset6

# [2] _____

```

**open3d()**

```

## wgl
## 61

plot3d(coords3Dn_diag_corr6, col = "orangered", size = 1, type = "s")

edges1 <- as_edgelist(g_bosc)
#edges1 <- edges[-c(1,18),]

```

```

# hom converteix els subgrafs en vectors d'arestes (com "A_B")
edges1_vert <- apply(as_edgelist(g_bosc_vertical1), 1, paste, collapse = "_")
edges1_lat <- apply(as_edgelist(g_retrolateral1), 1, paste, collapse = "_")

# assigna color segons pertinença
edge_colors <- sapply(1:nrow(edges1), function(i) {
  e <- paste(edges1[i, 1], edges1[i, 2], sep = "_")
  if (e %in% edges1_vert) {
    return("skyblue3") # flux vertical (causal)
  } else if (e %in% edges1_lat) {
    return("gold") # connexió retrolateral (retroflux i Lateral)
  }
})

for (i in 1:nrow(edges1)) {
  v1 <- which(V(g_bosc)$name == edges1[i, 1])
  v2 <- which(V(g_bosc)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr6[v1, ], coords3Dn_diag_corr6[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr6))
text3d(coords3Dn_diag_corr6[, "X"],
       coords3Dn_diag_corr6[, "Y"],
       coords3Dn_diag_corr6[, "Z"],
       texts = rownames(coords3Dn_diag_corr6),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr6[7, ], coords3Dn_diag_corr6[1, ]),col = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr6[1, ], coords3Dn_diag_corr6[8, ]),col = "blue", lwd = 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barlen = 0.02
)
text3d(p1_flux1,#[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical ", color = "green4",cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux2,

```

```

p1 = p1_flux2,
col = "green",
type = "rotation",
lwd = 2,
width = 0.3,
barrlen = 0.02
)
text3d(p1_flux2,
       texts = "Flux retrovertical ", color = "green4",cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux3,
       texts = "Flux laterovertical ", color = "green4",cex = 0.75,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux4,
       texts = "Flux vertical", color = "green4",cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr6[7, ], coords3Dn_diag_corr6[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr6[1, ], coords3Dn_diag_corr6[8, ]),
            col = "purple", lwd = 1.5)

# segeta blava
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",

```

```

type = "rotation",
lwd = 3,
width = 0.3,
barrlen = 0.02
)
text3d(p1_flux, #[1], p1_fLux[2], p1_fLux[3] + 4,
       texts = "Flux causal local",
       color = "purple4",
       cex = 0.75,
       depth_test="always")

# el flux causal holistic
res_holistic <- mitjana_vectorial_global(taula_general_fluxos_fulla)

v_sum <- res_holistic$v_sum # components reals
v_holistic <- res_holistic$v_global # direcció normalitzada

# hom escala la segeta perquè sigui comparable amb el model
escala <- 40

nodes_arrel <-c("PIP3", "PKA")

#p0 <- colMeans(coords3Dn_diag_corr6[nodes_arrel, , drop = FALSE])
#p0 <- coords3Dn_diag_corr5["Erk",]

coords_fulles <- coords3Dn_diag_corr6[c("P38","Jnk","Erk","Akt"), c("X","Y","Z")]
p0 <- colMeans(coords_fulles) # barycentre complet
p1 <- p0 + v_holistic * escala # direcció

arrow3d(
  p0 = p0,
  p1 = p1,
  type = "rotation",
  col = "red",
  lwd = 3,
  width = 0.3,
  barrlen = 0.02
)

text3d(p1,
       texts = "Flux causal holistic",
       col = "darkred",
       cex = 0.75,
       depth_test="always")

```

```

## VECTOR HOLÍSTIC
v_h <- resultat1$v_global
v_h_norm <- v_h / sqrt(sum(v_h^2))# 1. Vector holistic normalitzat

## ANGLES DE LES 4 FULLES RESPECTE AL VECTOR HOLISTIC
# 2. Extreiem els vectors de les 4 fulles
fulles <- taula_general_fluxos_fulla %>%
  filter(flux != "flux_objectiu") %>%
  select(u_bar_X, u_bar_Y, u_bar_Z) %>%
  as.matrix()

fulles_norm <- fulles / sqrt(rowSums(fulles^2))

angles <- acos(fulles_norm %*% v_h_norm) * 180/pi
theta_con <- max(angles)# 5. Amplada del con = màxima variació angular

theta_con# aquest ÉS l'angle correcte del con causal

## [1] 26.73612

## CON CAUSAL HOLISTIC 3D
library(rgl)

# paràmetres del con
angle <- theta_con * pi / 180 # en radians
h <- 40 # alçada del con (ajustable)
rad <- h * tan(angle) # radi base

# base d'un con alineat amb l'eix Z
n <- 60
theta <- seq(0, 2 * pi, length.out = n)
x <- rad * cos(theta)
y <- rad * sin(theta)
z <- rep(h, n)

# hom gira el con perquè apunyi com el flux objectiu
v_obj <- v_h_norm # <<<< el vector direcció normalitzat

z_axis <- c(0, 0, 1) # vector unitari Z

# Producte vectorial per trobar eix de rotació
rot_axis <- pracma::cross(z_axis, v_obj)
rot_angle <- acos(sum(z_axis * v_obj))

# Matriu de rotació (Rodrigues)
if (sqrt(sum(rot_axis^2)) > 0) {
  k <- rot_axis / sqrt(sum(rot_axis^2))

  K <- matrix(c(0, -k[3], k[2], k[3], 0, -k[1], -k[2], k[1], 0),
               nrow = 3,
               byrow = TRUE)

  R <- diag(3) * cos(rot_angle) +
    sin(rot_angle) * K +

```

```

    (1 - cos(rot_angle)) * (k %*% t(k))
} else {
  R <- diag(3)
}

# generem el con en format 3xN
cone_base <- rbind(x, y, z) # 3xN (base)
vertex <- matrix(c(0,0,0), nrow = 3, ncol = 1) # 3x1

# Apliquem la rotació correcta
cone_rot <- R %*% cone_base

# ara el con apunta EXACTAMENT en la direcció del flux holístic

# hom desplaça el con a un origen híbrid: Z segons els nodes arrels XY segons les 4 fulles
## --- 1) barycentre angular XY dels 4 col·lididors fulla ---
coords_fulles <- coords3Dn_diag_corr6[c("P38","Jnk","Erk","Akt"), c("X","Y","Z")]
B <- colMeans(coords_fulles) # barycentre complet

## --- 2) mitjana del nivell Z dels nodes arrel ---
#coords_arrel <- coords3Dn_diag_corr6[c("PIP3","PKA"), ]
#A <- colMeans(coords_arrel)

## --- 3) punt híbrid ---
#origin <- c(B["X"], B["Y"], A["Z"])
origin <- B
cone_pts <- sweep(t(cone_rot), 2, origin, "+")

# hom renderitza el con
triangles <- c()
for (i in 1:(n - 1)) {
  triangles <- rbind(triangles, cbind(i, i + 1, n + 1))
}

cone_mesh <- tmesh3d(
  vertices = t(rbind(cone_pts, origin)),
  indices = t(triangles),
  homogeneous = FALSE
)

shade3d(cone_mesh, alpha = 0.18, col = "red")
text3d(origin, texts = "Con causal holistic", col = "red", cex = 0.9)

```

```

aspect3d(1,1,1)

centre <- colMeans(coords3Dn_diag_corr6) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orangered4", size = 5)
# eixos centrats al centre de masses
dx <- (xmax - xmin) / 2
dy <- (ymax - ymin) / 2
dz <- (zmax - zmin) / 2

segments3d(rbind(centre - c(dx, 0, 0), centre + c(dx, 0, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, dy, 0), centre + c(0, dy, 0)), col="orangered4", lwd=0.9)
segments3d(rbind(centre - c(0, 0, dz), centre + c(0, 0, dz)), col="orangered4", lwd=0.9)

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]

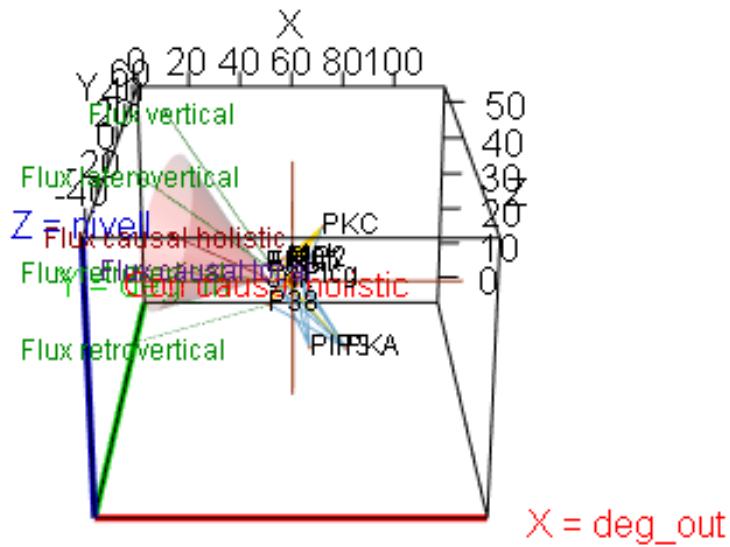
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymin, zmax)), col="blue3", lwd=3)
text3d(xmin, ymin, zmax, texts="Z = nivell", color="blue3", adj=c(0.5,0))
rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



## 6. MODEL CAUSAL GEOMÈTRIC COM A TOR FUNCIONAL/EMPÍRIC DEFORMABLE

Aquí tindríem el model tor funcional/empíric deformable:

Un model en cadena o arbre són modelables en 3D, no són encara una superfície sinó un aplec de punts i arestes (una constel·lació espacial), però no són modelables com a tor perquè no hi ha laterals a cap nivell que es puguin plegar damunt si mateixos, per tant un model d'estructura en cadena o arbòria no és modelable com a tor (model topològic tancat), geomètricament són espais oberts a dins de  $R^3$ . En canvi, sols que hi hagi un node lateral (mitjancer) que tingui o bé ( $\text{in} >= 1$  i  $\text{out} >= 2$ ) o bé ( $\text{in} >= 2$  i  $\text{out} >= 1$ ) i que hi hagi com a mínim un col·lididor-fulla ( $\text{in} >= 2$ ,  $\text{out} = 0$ ), el DAG ja és toritzable, és a dir, representable a dins d'un espai topològicament tancat (tor funcional deformable).

El producte cartesià tancat, *el tor funcional deformable*, ja no és un espai vectorial obert sinó una superfície compacta i orientable, on les dues dimensions laterals són periòdiques i la vertical manté el flux causal. La lectura geomètrica d'un DAG causal equival a una superfície causal plegada (el flux vertical és el gradient d'informació i el lateral és la difusió funcional). Cada subgraf (vertical i lateral) és una superfície parcialment compartida i el sistema sencer és el producte vectorial de totes dues dimensions.

El model en tor plega funcionalment primer (no pas causalment) els laterals de cada nivell (cada nivell genera un anell horitzontal: cilindre sols tancat lateralment però obert per cap i cua) i segon tanca funcionalment (compactació topològica del domini funcional) arrels i fulles (arrels i fulles són

funcionalment equivalents: el sistema pot regenerar les mateixes funcions d'entrada i sortida a dins d'un espai tancat); l'arbre fóra un con obert sols vertical sense redundància lateral, no hi ha cap manera de tancar l'espai, el flux sempre baixa o convergeix. Per exemple, un pathway metabòlic: l'enzim inicial (arrel) comença una cadena de reaccions, diversos enzims mitjancers (laterals) comparteixen substrats (aquí apareix el ciliaríndre funcional); si el producte final activa una funció equivalent a la inicial, no pas causalment, sinó per feedback funcional (per exemple, activació de la mateixa via a través d'un altre mecanisme), aleshores el sistema es torifica.

Al model 3D empíric previ, l'anàlisi és més aviat local, ara en tancar l'espai primer horizontal (hi ha lateralitat, no és una cadena ni tampoc un arbre) i segon vertical (amb un únic col·lididor-fulla ja és possible), s'obté el model tor empíric deformable, el qual permet analitzar el model de forma holística. Les arestes també han d'esser toritzades.

Un altre aspecte a tenir en compte és poder establir el volum de l'espai tancat com a 1 i poder treballar així amb probabilitats.

```
# graus i nivells
deg_in <- degree(g_bosc, mode = "in")
deg_out <- degree(g_bosc, mode = "out")
nivell <- V(g_bosc)$nivell

# paràmetres del tor
R <- 3 # radi gran (separació principal)
r <- 1.2 # radi petit (deformació local)

# angles toroidals reals (corregits)
# atès que els graus són petits, hom evita una fracció parcial del tor i són repartits
# uniformement
n_nodes <- length(V(g_bosc))
theta <- seq(0, 2*pi, length.out = n_nodes) # eix grau_in → angle θ
phi <- seq(0, 2*pi, length.out = n_nodes) # eix grau_out → angle φ

# Barregem una mica per evitar simetries perfectes
set.seed(777)
theta <- sample(theta)
phi <- sample(phi)

# coordenades toroidals
coords3Dn_diag_corr1_tor <- cbind(
  X = (R + r * cos(theta+0.25*nivell)) * cos(phi),
  Y = (R + r * cos(theta+0.25*nivell)) * sin(phi),
  Z = r * sin(theta+0.25*nivell) # "nivell" = ascensió causal vertical
)

# colors segons rol causal
cols <- ifelse(deg_out == 0, "gold",
ifelse(deg_in == 0, "red",
ifelse(deg_in >= 2 & deg_out >= 2, "orange", "skyblue")))

open3d()
## wgl
## 64
```

```

plot3d(coords3Dn_diag_corr1_tor, col = cols, size = 1.2, type = "s", box = FALSE)
axes3d(col = "gray60")

# arestes direccioades amb curvatura toroidal
edges <- as_edgelist(g_bosc)

n_steps <- 30 # resolució de la corba

for (i in 1:nrow(edges)) {
  v1 <- which(V(g_bosc)$name == edges[i,1])
  v2 <- which(V(g_bosc)$name == edges[i,2])

  # angles corresponents
  t1 <- theta[v1]; t2 <- theta[v2]
  p1 <- phi[v1]; p2 <- phi[v2]
  n1 <- nivell[v1]; n2 <- nivell[v2]

  # correcció angular (travessa pel costat curt del tor)
  if (abs(p2 - p1) > pi) {
    if (p2 > p1) p2 <- p2 - 2*pi else p2 <- p2 + 2*pi
  }

  # seqüència suau entre els dos nodes
  t_seq <- seq(t1, t2, length.out = n_steps)
  p_seq <- seq(p1, p2, length.out = n_steps)
  n_seq <- seq(n1, n2, length.out = n_steps)

  # coordenades toroidals intermitges
  X <- (R + r * cos(t_seq + 0.25 * n_seq)) * cos(p_seq)
  Y <- (R + r * cos(t_seq + 0.25 * n_seq)) * sin(p_seq)
  Z <- r * sin(t_seq + 0.25 * n_seq)

  # corba com a línia suau
  lines3d(X, Y, Z, col = "gray40", lwd = 1.2)
}

text3d(coords3Dn_diag_corr1_tor, texts = V(g_bosc)$name, adj = c(0.6, -0.4), color = "black")

# dimensions
text3d(R+2, 0, 0, text = "deg_out", color = "red")
text3d(0, R+2, 0, text = "deg_in", color = "green")
text3d(0, 0, max(coords3Dn_diag_corr1_tor[,3])*1.2, text = "nivell causal (Z)", color = "blue")

```

```

# superfície toroidal transparenta
u <- seq(0, 2*pi, length.out = 40)
v <- seq(0, 2*pi, length.out = 40)

U <- outer(u, v*0, FUN="+")
V <- outer(u*0, v, FUN="+")

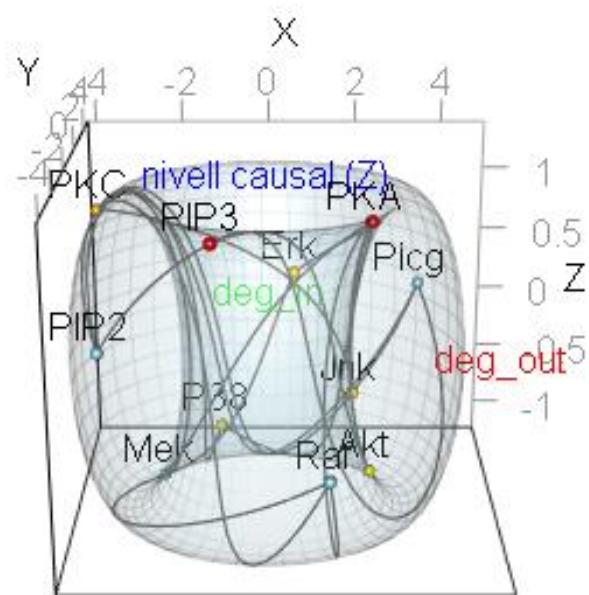
x_surf <- (R + r * cos(V)) * cos(U)
y_surf <- (R + r * cos(V)) * sin(U)
z_surf <- r * sin(v)

surface3d(x_surf, y_surf, z_surf, color = "lightblue", alpha = 0.2, front = "lines")

rglwidget()

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

```



1. Una altra natura de qüestions emanen d'aquesta geometria causal:  
hom es pot demanar quin comportament té el tor si té un nombre fix de nodes però in->inf i out->inf... o si al DAG toroïdal hi han cicles locals.

## ANNEX2 CONCEPTES COMPLEMENTARIS DEL MODEL CAUSAL GEOMÈTRIC EN 3D

### A1. conversió del pathway a DAG i la seva descomposició

La conversió del pathway de Sachs et al. a un DAG 2D acíclic és resolta per una lectura atenta de l'article de treball. Des del meu punt de vista, els autors del pathway són els únics qui poden fer canvis en el disseny del pathway teòric, una altra cosa és quan hom completa i valida el model amb les dades empíriques. Així, per tal de fer la conversió del pathway a DAG 2D, hom resol dues arestes bidireccionals a fi que passin a esser unidireccionals i evitar així cicles a dins de l'estructura. El pathway ja és fet amb arestes dirigides, per tant, no calia resoldre arestes sense direcció de flux.

La segona fase ha estat la descomposició del DAG 2D en dos subgrafs (cadascun dels quals ha sortit non-connex): el vertical és fet de dos arbres sense unió, per tant, un bosc dirigit non-conex, el primer té tres nivells i el segon, dos. Els components del graf lateral no són definits a dins d'un únic nivell, per exemple, el node PKC, qui uneix els dos arbres del bosc vertical, té un sentit contrari al flux vertical, d'aquí que calgui anomenar el subraf lateral com a retrolateral. El retrolateral és compost d'una cadena mínima de dos nodes (un única aresta que inclou un node arrel) i un segon component lateral divers en termes de tipus causals (confonedor, col·lididor, forca). Tots dos components del retrolateral són de dos nivells, cap de tres.

La descomposició d'un DAG 2D ja no és sols una eina de perfeccionament i/o validació del mateix DAG 2D sinó que també és el primer pas de constitució d'un model causal geomètric, el qual podria potser anar més enllà del mateix DAG en dues dimensions. Per exemple, en biologia estructural, tots els seus models són en 3D, *hom converteix la xarxa bioquímica en un model causal geomètric 3D, en què l'estructura d'arestes i nodes és viva gràcies a la seqüència del temps* que és: → deformable segons l'input de dades observades → reversible en termes del sentit de les arestes segons el node condicionat (per exemple en condicionar un node que és entre dos nodes non-adjacents a fi d'avaluar-ne la seva independència, és a dir, en aplicar un test geomètric d'independència entre dos nodes non-adjacents, el sentit de l'aresta del node condicionat en qüestió es veu afectat). → dinàmic si hi ha un contínum de dades observades, si s'hi inclou la quarta dimensió temps (fóra un DAG 3D en moviment, per exemple, abans i després d'un estímul, el núvol de nodes i arestes prenen configuracions diferents entre dos dues unitats de temps consecutives). L'espai-temps emergeix com a projecció geomètrica del flux causal.

Hom pot comparar l'estructura geomètrica causal d'una via senyalitzadora amb la seva estructura tridimensional molecular; veure quins canvis topològics (en in/out) corresponen a canvis físics (plegament, interacció...); o detectar punts de transició on la geometria causal es trenca (indicadors de canvi funcional, mutacions, inhibicions...). Hom creu que és del tot justificat entendre ara la descomposició del DAG 2D com una primera fase per a la modelització del pathway mèdic o biològic en un model d'inferència causal geomètric en 3D.

### A2. Grau d'entrada (dependència), grau de sortida (efecte) i nivell (jerarquia de flux causal)

Aquesta descomposició permet de veure les tres dimensions essencials i estructurals d'un DAG en 2D: nivells i dues menes de graus (grau\_IN i grau\_OUT, entrant i sortint), és a dir, cada node pertany

a un únic nivell del bosc i té un nombre determinat d'ins i outs (d'arestes entrants i sortints de cada node).

Aquestes tres dimensions suposen un salt intuït però no previst, perquè ara hom formalitza un model d'inferència causal geomètric (el primer segons la informació que hom té a l'abast), en què aquest model geomètric recull primer la informació topològica del DAG 2D (amb escalatge inclòs dels eixos per tal de no deformar la distribució espacial dels nodes) i després recull la informació empírica de les dades registrades per mitjà d'una normalització dels nodes segons diverses projeccions (la mitjana i la desviació tipus de les onze variables-columna i després les correlacions); la distribució espacial topològica dels nodes passa a esser-ne després una de topoempírica: hom pot veure que la distribució topològica de nodes es reconfigura a posteriori, si hi han nodes encavalcats en coordenades topològiques, deixen de sobreposar-se i ocupen el seu lloc a l'espai).

### *A3. Test d'independència condicionada: reversió del sentit de l'aresta\_IN o aresta\_OUT (a fi de comparar els dos vectors directors locals i condicionament)*

Si hom treballa amb geometria algebraica lineal les definicions de condicionament sobre un node tenen restriccions perquè, si bé obereixen a simetries que responen a relacions purament algebraiques ( $AB=B-A$ ,  $BA=-(B-A)$ ,  $|AB|=|BA|$ ,  $\text{angle}(AB,Z)=-\text{angle}(BA,Z)\dots$ ), no responen a relacions causals ( $BA$  no és  $-AB$ , el pendent vertical és causal no pas geomètric, hi han contrafluxos...els vectors directors no són lineals algebraics sinó físics perquè tenen intensitat, verticalitat causal...); llavors això podria esser un problema perquè limita les possibilitats d'expressar geomètricament les susdites relacions causals. La geometria algebraica lineal és euclidiana, la geometria algebraica causal és non-euclidiana (el mateix eix privilegiat de flux causal vertical Z ja trenca Euclides)

Per exemple: si hom condiciona pel node medial d'una forca ( $A <- B -> C$ ) sota la definició que totes les arestes\_OUT esdeven arestes\_IN ( $A \rightarrow B <- C$ ) i aplica una pura geometria algebraica, llavors tant condicionament ( $A <- B -> C$ ) com condicionament ( $A \rightarrow B <- C$ ) l'angle respecte l'eix vertical Z és el mateix (amb canvi de signe) i el mòdul és exactament el mateix. Cosa que en termes causals o físics no és correcte perquè són fenòmens opositius (un és el xoc de fluxos, es frenen, l'energia vectorial es dissipa, per tant el mòdul ha d'escurçar-se); en canvi, a una forca el flux es divideix però no topa, el mòdul ha d'ésser més llarg que el d'una topada de forces). D'aquí que calgui recórrer a la cadena per poder comparar la forca condicionament ( $A <- B -> C$ ) i la mateixa condicionament ( $A \rightarrow B -> C$ ), però pagant-ne un preu, la independència dels nodes extrems és parcial, sols un n'és respecte de l'altre, el node A respecte del C o a l'inrevés (hi torna haver simetria) però no tots dos alhora. El mateix problema hi ha amb la figura causal del col·lididor, hom s'és a la situació inversa; cal recórrer una altra volta a la solució parcial de la cadena. Una segona solució és fer la comparança condicionament versus el flux causal vertical general corregit.

Ara bé, si hom treballa amb geometria física o causal, no hi ha cap restricció, sigui quina sigui la definició que hom fa de condicionament en cap dels tres tipus causals essencials: cadena, forca i col·lididor. Ara hom pot definir geomètricament el col·lididor com a invers causal de la seva forca i la forca com a invers causal del seu col·lididor (ara hom pot invertir el sentit de les arestes de tots dos vectors), hom no cau en l'artifici algebraic (la simetria) que no descriu el seu significat causal; igual com en física: la forca/divergència allarga el mòdul (té un guany de nivell energètic) i el col·lididor/convergència escurça el mòdul (pèrdua de força) per saturació.

Per exemple, de la forca  $A <- B -> C$ ,  $B \rightarrow A$  és molt vertical i  $B \rightarrow C$  és molt horitzontal, les dues no poden esser perfectament opositades perquè són arrelades a la geometria espacial dels nodes:  la direcció sortint no és simètrica (el vectors directors no tenen la mateixa inclinació)  de la mateixa manera,

quan hom inverteix el sentit d'una aresta, el mòdul/intensitat del flux tampoc no és la mateixa (no hi pot haver simetria inversa quan en canvia el sentit, perquè ara A->B suposa anar en contra la verticalitat i B->C té un pendent diferent; és a dir, la inversió del flux no inverteix la geometria del camí).

Cal afegir tres notes addicionals: hom preveu que la millor manera de desenvolupar el model causal geomètric és assumir la geometria física perquè hom suposa que aquesta dóna millors resultats respecte als resultats estadístics com a punt de referència de validació. I una segona observació, les dades biomèdiques són treballades per mitjà de la geometria física, no pas amb àlgebra lineal. I en darrer lloc, dir que els vectors directors són definits de manera diferent segons sigui geometria algebraica lineal o física. En resum, no sembla que la simetria algebraica existeixi en la geometria real en l'àmbit biomèdic on hom mesura les dades d'éssers vivents.

En condicionar per un node —resultat d'aplicar un test local geomètric d'independència entre dos nodes non-adjacents— en un model geomètric 3D vol dir teòricament (les dades poden contradir el model si detecten camins espuris):

### *EN GEOMETRIA ALGEBRAICA LINEAL: geometria euclidiana*

~> per un cadena (vector director cadena versus vector director col·lididor): que una de les arestes\_OUT revertexi el seu sentit cap al mateix node condicionat; o just a l'inrevés, que una de les arestes\_IN revertexi el seu sentit cap al node adjacent (la cadena esdevé o bé un col·lididor o bé una forca). ~> per una forca (vector director forca versus vector director cadena): que una de les dues arestes\_OUT revertexi el seu sentit cap al mateix node condicionat, és a dir, que el node condicionat sols tingui una aresta\_IN i la resta romanguin arestes\_OUT (la forca esdevé cadena). Hom no pot revertir totes les arestes\_OUT a arestes\_IN perquè són geomètricament simètriques; a més, hom vol veure si hi ha un camí espuri entre els nodes fills, cosa que implica un vector director en configuració de cadena, no pas de forca.

~> per un col·lididor (vector director col·lididor versus vector director cadena): que una de les dues arestes\_IN sotmeses a test d'independència condicionada revertexi el seu sentit cap al mateix node condicionat, és a dir, que el node condicionat sols tingui una aresta\_OUT i la resta romanguin com a arestes\_IN (el col·lididor esdevé també cadena). El mateix fenomen de simetria susdit al cas d'una forca.

### *EN GEOMETRIA FÍSICA O CAUSAL: geometria vectorial non-euclidiana a dins d'un espai causalment estratificat*

~> per un cadena (vector director cadena versus vector director col·lididor): que una de les arestes\_OUT revertexi el seu sentit cap al mateix node condicionat (també és plantejable just a l'inrevés, que una de les arestes\_IN revertexi el seu sentit cap al node adjacent: aquí la cadena esdevé una forca). ~> per una forca (vector director forca versus vector director col·lididor): que les dues arestes\_OUT sotmeses a test d'independència condicionada revertixin el seu sentit cap al mateix node condicionat. ~> per un col·lididor (vector director col·lididor versus vector director forca): que les dues arestes\_IN sotmeses a test d'independència condicionada revertixin el seu sentit cap al node condicionat.

### *A4. validació empírica multiescala del DAG resultant*

Hom podria validar el DAG 2D, el model causal geomètric 3D o la seva versió toroïdal per mitjà de la tríada de tècniques d'aprenentatge estadístic (machine learning), amb diferents nivells d'anàlisi: nodes o variables (Random Forests): triatge empíric de les variables més rellevants; validació local (Suport Vector Machine, d'ara endavant SVM): comprovació de relacions causals puntuals; i a la fi —si és possible— validació global (Xarxes Neuronals, d'ara endavant ANN o CNN).

## *A5. trasllat del concepte d'independència condicionada del DAG 2D al model geomètric 3D*

Una forma possible de traslladar-hi el concepte és emprant el sentit de les arestes, és a dir, el node condicionat a la independència de dos nodes non-adjacents és justament passar el grau\_out com a grau\_in, altrement dit: a una cadena de tres nodes, és condiciona pel medial i aquest té una aresta\_IN i una aresta\_OUT, doncs ara el node medial passa a tenir dues arestes\_IN i cap aresta\_OUT (o just a l'inrevés: el node medial passa a tenir cap aresta\_IN i dues arestes\_OUT); i sembla lògic perquè en totes dues situacions l'angle del vector director és el mateix però amb el signe canviat (forca i col·lididor són geomètricament simètrics respecte d'una configuració de 3 nodes).

## *A6. Trencament d'un col·lididor-fulla (una aresta cap al subgraf vertical i una altra, al retro-lateral)*

en fer la descomposició del DAG 2D general al model geomètric 3D emergeixen dos col·lididors-fulla (dos nodes pares envien una aresta cadascun a un únic node fill, el qual no té néts, d'aquí que en digui col·lididor-fulla) en l'aplicació de la definició de graf arbre (una única grau\_IN en cada node, llevat dels dos nodes arrels que sols tenen graus\_OUT). En aplicació de la definició del graf arbre o bé hom treuca el node col·lididor-fulla (una aresta al vertical i la segona o bé al lateral o bé rebutjada, la tria de quina roman al vertical és feta per mitjà de test estadístic, la més informativa en termes de flux d'informació) o bé rebutges el node col·lididor-fulla. Hom prioritza no perdre gens d'informació, per tant, no perdre cap node ni cap aresta; d'aquí que hom opta per trencar el col·lididor-fulla i enviar l'aresta més informativa al vertical i la segona, al lateral. Aquesta decisió té una conseqüència: per als components del subgraf vertical, l'única conseqüència és que el bosc deixa d'esser connex.

## *A7. Flux causal vertical corregit (mitjana vectorial vertical global o bé segons l'eix vertical Z o bé segons tots tres eixos ZYX)*

el flux causal vertical sorgeix de fer la mitjana vectorial vertical global (fer de forma robusta la bisectriu de forma sistemàtica) entre tots els vectors directors verticals (les arestes verticals del subgraf bosc), però aquí cal fer una puntualització perquè en el subgraf retrolateral hi han vectors directors que van en sentit contrari als verticals. Cal doncs calcular un flux vertical i un altre diguem-ne de retroflux (aquí no s'hi inclouen pas les arestes laterals, les que es lliguen dins d'un mateix nivell, sols les que van en contra del flux arbori vertical, és a dir, que el sentit de les arestes van dels nodes-fulla cap als nodes-arrels) i calcular un flux causal vertical corregit.

Si el flux i el retroflux tenen la mateixa direcció encara que sentit contrari, sols cal corregir el mòdul del vector de flux vertical (en aquest cas és més curt i, per tant, assenyala que minva el flux de força causal de nodes-arrel a nodes-fulla); però si el flux i retroflux no són col-lineals, cal fer també una correcció d'angle del vector de flux vertical, és a dir, cal corregir el mòdul i l'angle (i un mòdul més curt i un angle menys vertical). És essencial que el vector general de flux vertical tingui l'angle respecte Z i la llargària del mòdul òptims d'acord amb les arestes del subgraf vertical (el subgraf bosc) i les arestes de contraflux del subgraf retrolateral (altrement dit, triar del subraf retrolateral sols les arestes de contraflux).

En aquest estudi, el contraflux prové del node PKC, abans node-fulla del primer arbre al subgraf de flux causal vertical, esdevé emissor retroactiu a dins del graf complet, és a dir, esdevé un node medial de retroalimentació.

La correcció del flux general és just aquí: de les arestes dels col·lididors-fulla hom n'obté un mòdul més llarg (més força) i un angle molt vertical (pels volts de 90°) però, per contra, de les arestes dels col·lididors-retroverticals hom n'obté un mòdul més curt (pot arribar a anul·lar-se o esser negatiu en

termes de flux causal) i un angle lateralitzat que influeix en l'angle de flux vertical general. Per tant, per tal de calcular el flux causal general vertical caldrà fer la correcció per cada node-fulla del subgraf vertical que sigui col·lididor-fulla o col·lididor-retrovertical a l'espai de transició entre subgrafs.

## A8. Interval de flux (IF)

Sorgeix d'agafar l'angle mínim i màxim de les arestes que participen del càlcul del flux vertical corregit. Aquest interval assenyala els límits de convergència possibles per a qualsevol via retrolateral; com més a prop de la mitjana de l'IF més forta és la convergència d'una via retrolateral al flux causal vertical.

## A9. Proximitat nodes al model causal geomètric

Com més a prop són dos nodes a l'espai causal més probable que hi hagi una aresta no detectada fins al punt que hom pugui agrupar-les com a una única variable atès que en el cas degenerat les dues variables se sobreposarien del tot. Fet que encara es reforça més si totes dues són molt a prop del centre de masses.

## A10. Espai de transició entre els subgrafs vertical i retrolateral: els nodes-frontera

Els nodes-fulla del subgraf vertical són també els nodes-frontera de l'espai de transició que uneixen els dos subgrafs i són tots col·lididors (retro)verticals; els col·lididors que surten al subgraf retrolateral són o bé retrolaterals o bé retrolateroverticals (llevat de l'aresta que prové de la descomposició del col·lididor fulla vertical), tenen sempre com a mínim una aresta lateral, però tota la resta són col·lididors sense lateralitat i sols són visibles si hom té en compte aquest espai de transició entre els dos subgrafs de la descomposició, és a dir, els nodes comuns a tots dos subgrafs (d'aquí que hom en dedueix que el mínim de nodes que pot tenir el subgraf retrolateral és el nombre de fulles del subgraf vertical). D'aquí que la descomposició hagi trencat tots els col·lididors retroverticals i sols hagin romàs els que tenen com a mínim una aresta lateral.

## A11. Mitjana vectorial global

La bisectriu simple deixa d'esser representativa i cal obtenir una direcció mitjana que reflecteixi la resultant geomètrica real del camp causal 3D. La versió robusta de la bisectriu sistemàtica és la mitjana vectorial global, quan hi ha més de dues arestes que defineixen el flux (diverses direccions locals verticals, quan hi ha 3 o més direccions locals o angles dispersos) i els angles entre aquests poden anar més enllà dels 45°, altrament dit, és invariant a l'ordre de combinació dels vectors i geomètricament equivalent a una direcció sortint mitjana (reflecteix la direcció resultant real del camp causal 3D).

És la direcció de màxim consens funcional entre totes les arestes verticals del subgraf  $i$ , així, obtenir l'angle òptim de flux causal vertical per a un model geomètric concret. Formalment, correspon a la suma vectorial ponderada dels vectors directors  $v_{\square_i}$  de cada aresta, on  $w_i = |v_{\square_i}|$  és el mòdul de cada vector (la seva força en termes de flux causal local) que actua com a pes. Aquesta ponderació assegura que les arestes amb flux causal més intens tinguin més de pes en la definició de la direcció resultant. Geomètricament, és la mitjana vectorial global és una direcció sortint mitjana.

$$V_{\square\_global} = \left( \sum_i [w_i v_{\square_i}] \right) / \left( \left\| \sum_i [w_i v_{\square_i}] \right\| \right)$$

Reversió del sentit de l'aresta\_IN o aresta\_OUT d'un node condicionat:

en condicionar per un node —resultat d'aplicar un test local geomètric d'independència entre dos nodes non-adjacents— en un model geomètric 3D vol dir teòricament (les dades poden contradir el

model si detecten camins espuris): □ per un cadena: que una de les arestes\_OUT reverteixi el seu sentit cap al mateix node condicionat; o just a l'inrevés, que una de les arestes\_IN reverteixi el seu sentit cap al node adjacent (la cadena esdevé o bé un col·lididor o bé una forca). □ per una forca: que una de les dues arestes\_OUT reverteixi el seu sentit cap al mateix node condicionat, és a dir, que el node condicionat sols tingui una aresta\_IN i la resta romanguin arestes\_OUT (la forca esdevé cadena). Hom no pot revertir totes les arestes\_OUT a arestes\_IN perquè són geomètricament simètriques; a més, hom vol veure si hi ha un camí espuri entre els nodes fills, cosa que implica un vector director en configuració de cadena, no pas de forca.

□ per un col·lididor: que una de les dues arestes\_IN sotmeses a test d'Independència condicionada reverteixi el seu sentit cap al mateix node condicionat, és a dir, que el node condicionat sols tingui una aresta\_OUT i la resta romanguin com a arestes\_IN (el col·lididor esdevé també cadena). El mateix fenomen de simetria susdit al cas d'una forca.

Nova significança per als tipus causals bàsics: cadena, forca, col·lididor...

Aquests tipus causals al DAG causal 3D deixen d'esser una estructura en si mateixa, és a dir, per exemple, una cadena de 3 nodes al pla pot semblar trivial amb relació al DAG 2D per la seva mateixa linealitat si no hi han connexions espúries (per exemple un confonedor); però la mateixa cadena a dins d'un model geomètric 3D fa que la seva posició relativa a l'espai geomètric sigui molt més significativa segons el flux causal que no pas la seva estructura en si mateixa; és a dir, no són el mateix en termes de causalitat una cadena vertical, una de lateral a dins d'un mateix nivell o a dins de nivells diversos... o fins i tot una de retrovertical. Una senzilla cadena a dins de l'espai geomètric pot tenir un impacte causal molt divers, fins al punt que una pot esser antagònica a una altra, per exemple, els casos de cadena vertical versus lateral. Aquestes distincions són invisibles al DAG 2D, a dins del pla totes són cadenes.

Així que ara el marc geomètric imposa una lectura de cada tipus causal no sols per l'estructura interna d'ells mateixos sinó —i encara més important— la posició relativa que ocupa a dins de l'espai geomètric, ara cadascun és un tipus polièdric en termes de causalitat: l'estructura i la seva posició relativa a dins dels eixos de coordenades 3D.

## A. Model de tor empíric deformable

Un model en cadena o arbre són modelables en 3D, però no són encara una superfície sinó un aplec de punts i arestes (una constel·lació espacial), però no són modelables com a tor perquè no hi ha laterals a cap nivell que es puguin plegar damunt si mateixos, per tant un model d'estructura en cadena o arbòria no és modelable com a tor (model topològic o empíric tancat), geomètricament són espais oberts a dins de  $R^3$ . L'arbre fóra un con obert sols vertical sense redundància lateral, no hi ha cap manera de tancar l'espai, el flux sempre baixa o divergeix.

En canvi, quan hi han nodes laterals (mitjancers) amb  $\text{arestes\_OUT} >= 2$  i amb col·lididors-fulla ( $\text{in} >= 2$ ,  $\text{out} = 0$ ), el DAG ja comença a esser toritzable, és a dir, representable a dins d'un espai topològicament tancat (tor funcional deformable, de fet tor empíric deformable perquè abans ja és fet el canvi de model topològic geomètric 3D a topoempíric). ##. ### PRIMER PLEGAMENT

**HORIZONTAL** El model en tor es plega funcionalment primer (encara no pas causalment) pels laterals de cada nivell: cada nivell genera un anell funcional horizontal (un cilindre obert per cap i cua, però ja tancat lateralment). Aquest primer plegament funcional implica un tancament de funcions equivalents funcionalment (compactació topològica del flux local). Aquests anells horitzontals estableixen isomorfismes funcionals entre els punts d'entrada i sortida del nivell: el flux ja no és lineal ni jeràrquic sinó recorrent i autoreferencial a dins de cada domini horitzontal. Quan aquesta estructura d'anells es projecta empíricament en la geomatria de fluxos, altrettant dit, quan les direccions de flux i contraflux mesurables defineixen un circuit tancat a dins de l'espai tridimensional, el sistema passa d'esser funcional a empíric.

## SEGON PLEGAMENT VERTICAL

Un DAG 3D acíclic manté la direccionalitat del flux causal (d'arrels a fulles), tanmateix la retroalimentació local estableix una transferència funcional inversa sense violar la causalitat (sempre que no reentri a un node arrel, sols connecta localment dos nivells funcionals). És a dir: el plegament o simetria arrel – fulla no implica causalitat invertida sinó ressonància d'informació a dins del mateix flux. En introduir retroalimentació local, el flux no s'inverteix pas sinó que es corba, el flux segueix essent ascendent (acíclic). El flux no torna enrere sinó que connecta els extrems del gradient causal per la corbatura de l'espai de flux; l'arrel i fulla ara són dins d'una mateixa classe funcional sense deixar de pertànyer a un sistema acíclic.

El segon plegament entre arrels i fulles (arrels i fulles són funcionalment equivalents: el sistema pot regenerar les mateixes funcions d'entrada i sortida a dins d'un espai tancat); arrels i fulles segueixen essent causalment diferents però funcionalment simètriques a dins del tor perquè el domini ara és tancat i tota la informació es preserva. Per exemple, un pathway metabòlic: l'enzim inicial (arrel) comença una cadena de reaccions, diversos enzims mitjancers (laterals) comparteixen substrats (aquí apareix el cilinídre funcional); si el producte final activa una funció equivalent a la inicial, no pas causalment, sinó per feedback funcional (per exemple, activació de la mateixa via a través d'un altre mecanisme), aleshores el sistema es torifica.

La possibilitat de toritzar el model causal geomètric permet fer una anàlisi holística del model, com un tot, i no tant de forma local. Però encara més important, la torització del model geomètric pot facilitar la seva validació amb machine learning o aprenentatge estadístic.

### A. Model toroïdal acíclic amb retrofluxos

El model toroïdal accepta retrofluxos locals sense comprometre l'aciclicitat total del model