

# **INFERÈNCIA CAUSAL GEOMÈTRICA COM A ESTRUCTURA DE L'ESPAI-TEMPS**

**De pathway a un model causal geomètric 3D**



**Universitat  
Oberta  
de Catalunya**



**UNIVERSITAT  
DE  
BARCELONA**

**David Graupere Villà**

MU Bioinformàtica i  
Bioestadística  
Àrea de treball final

**Tutora de TF**  
Agnès Pérez Millan

**Professors responsables  
de l'assignatura**  
Ferran Reverter Comas  
Joan Carles Lario Loyo

7 de gener de 2026

**«...δεν πιστεύουμε ότι γνωρίζουμε ένα πράγμα πριν συλλάβουμε το γιατί , ήτοι το “διὰ τί” του (πριν δηλαδή συλλάβουμε την πρώτη αιτία)...» (Φυσικά, 194 b 17 – 20)**  
**Αριστοτέλης (350 π.Χ.)**

**«Hom pensa que no té coneixement d'una cosa fins que no se n'ha copsat el perquè, és a dir, la seva causa»**  
**Aristòtil (350 aC)**

**«El desenvolupament de la ciència occidental es basa en dos grans èxits: la invenció del sistema lògic formal (en geometria euclidiana) pels filòsofs grecs, i el descobriment de la possibilitat d'esbrinar relacions causals mitjançant experiments sistemàtics (durant el Renaixement)»**

**Albert Einstein (1953)**

**A la meva mare, Teresa**  
**Als pacients codirectors del TFG, Erik Cobo i Xavier Puig**  
**Als amics i professors Antonio Lubary i Natalia Sadovskia**



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](#)

## B) GNU Free Documentation License (GNU FDL)

Copyright © ANY EL-TEU-NOM.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## C) Copyright

© (l'autor/a)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

# INFERÈNCIA CAUSAL GEOMÈTRICA COM A ESTRUCTURA DE L'ESPAI-TEMPS

## De pathways a un model causal geomètric 3D

david graupere villà

2025-09-25

### Table of Contents

<b>TESI DE MÀSTER EN BIOESTADÍSTICA I BIOINFORMÀTICA UOC-UB .....</b>	<b>14</b>
<b>Abstract / Extracte .....</b>	<b>14</b>
<i>Objectiu general.....</i>	16
<i>Objectiu nuclear de validació.....</i>	16
<i>Objectius específics .....</i>	17
1.3. <i>Impacte en sostenibilitat, ètic-social i de diversitat.....</i>	17
<i>Impacte en sostenibilitat.....</i>	17
<i>Impacte ètic-social i responsabilitat social .....</i>	18
<i>Impacte en diversitat i drets humans .....</i>	18
1.4. <i>Enfocament i mètode seguit .....</i>	18
1.5. <i>Planificació del Treball .....</i>	19
1.6. <i>Breu sumari de productes obtinguts .....</i>	23
1.7. <i>Breu descripció dels altres capítols de la memòria .....</i>	23
2. <i>Estat de l'art .....</i>	24
3. <i>Materials i mètode: projecció i transformació geomètrica (topològica i empírica) .....</i>	25
LA METÀFORA BAYESIANA.....	30
CADENA DE TRANSFORMACIONS GEOMÈTRIQUES A PARTIR DE LES DADES: .....	31
NORMALITZACIÓ RADIAL: PRIMERA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI.	
Mitjana/desviació tipus global .....	32
NORMALITZACIÓ INDIVIDUAL: SEGONA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI. Correlació .....	33
JUSTIFICACIÓ D'ÚS DE LA CORRELACIÓ A LA SEGONA TRANSFORMACIÓ GEOMÈTRICA: l'ús de la funció localTest() .....	33

<i>JUSTIFICACIÓ D'ÚS DE LA CORRELACIÓ A LA SEGONA TRANSFORMACIÓ GEOMÈTRICA: l'ús de la funció localTest()</i> .....	35
3.6.1 CONCLUSIONS EN VALIDACIÓ TEÒRICA PER TIPUS CAUSALS MÍNIMS .....	36
3.6.2 ASSAIG DE VALIDACIÓ SIMULADA A ESCALA LOCAL EN CAMINS DE MÉS DE TRES NODES .....	37
3.6.3 FLUX CAUSAL GENERAL del model geomètric topoempíric general .....	38
ASSUMPCIÓ ESFERA DE DIRECCIONS.....	38
IMPLICACIONS D'UN MODEL GEOMÈTRIC AMB O SENSE ARESTA EN QÜESTIÓ.....	39
4. Resultats.....	41
4.1 VALIDACIÓ MULTIESCALA DEL DAG 2D COMBINATORI PER MACHINE LEARNING O APRENENTATGE ESTADÍSTIC .....	41
Random Forests (variables), SVM (validació local) i ANN/CNN (validació global) .....	41
4.2 VALIDACIÓ LOCAL DEL MODEL CAUSAL GEOMÈTRIC .....	42
4.1.1 FLUX CAUSAL LOCAL de la cadena lateral en el marc del model geomètric topoempíric general .....	42
4.1.2 FLUX CAUSAL VERTICAL GENERAL del model geomètric topoempíric general.....	42
ASSUMPCIÓ ESFERA DE DIRECCIONS:.....	42
4.1.3 FLUX CAUSAL VERTICAL (8 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1 (SENSE ARESTA Erk->Akt) .....	42
4.1.4 FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1 (SENSE ARESTA Erk->Akt) .....	43
Hom força la inclusió de l'aresta absent Erk -> Akt com a hipòtesi estructural alternativa .....	44
4.1.5 FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1_MOD (AMB ARESTA Erk->Akt) .....	45
4.1.6 FLUX CAUSAL RETROLATERAL (3 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1_MOD (AMB ARESTA Erk->Akt) .....	46
VALIDACIÓ DE L'ARESTA ABSENT DE L'ARTICLE DE SACHS ET AL. COM A HIPÒTESI ESTRUCTURAL ALTERNATIVA .....	47
VALIDACIÓ GLOBAL DEL MODEL CAUSAL GEOMÈTRIC: subgraf retrolateral versus torització del model .....	47
4.3. Objectius: anàlisi crítica dels resultats.....	48
5. Discussió sobre l'aplicabilitat i l'abast del nou mètode.....	50
6. Conclusions i treballs futurs: línies de continuació de desenvolupament del mètode.....	51
7. GLOSSARI DEL MÈTODE CAUSAL GEOMÈTRIC EN 3D.....	53
A1. conversió del pathway a DAG i la seva descomposició .....	53
A2. Grau d'entrada (dependència), grau de sortida (efecte) i nivell (jerarquia de flux causal) .....	54

<i>A3. Test d'independència condicionada a fi de comparar els dos vectors directors locals i condicionament.....</i>	54
<i>EN GEOMETRIA ALGEBRAICA LINEAL AMB TRES NODES: geometria euclidiana .....</i>	55
<i>EN GEOMETRIA FÍSICA O CAUSAL AMB TRES NODES: geometria vectorial non-euclidiana a dins d'un espai causalment estratificat.....</i>	56
<i>A4. validació empírica multiescala del DAG resultant .....</i>	56
<i>A5. trasllat del concepte d'independència condicionada del DAG_combinatori al model geomètric 3D .....</i>	57
<i>A6. Trencament d'un col·lididor-fulla (una aresta cap al subgraf vertical i una altra, al retro-lateral).....</i>	57
<i>A7. Flux causal vertical corregit (mitjana vectorial vertical global o bé segons l'eix vertical Z o bé segons tots tres eixos ZYX) .....</i>	57
<i>A8. Interval de flux (IF) .....</i>	58
<i>A9. Proximitat nodes al model causal geomètric .....</i>	58
<i>A10. Espai de transició entre els subgrafs vertical i retrolateral: els nodes-frontera.....</i>	58
<i>A11. Mitjana vectorial global.....</i>	58
<i>A12. Nova significança per als tipus causals bàsics: cadena, forca, col·lididor .....</i>	59
<i>A13. DEFINICIONS CLAU FINALS.....</i>	59
<b>8. Referències .....</b>	60
<i>8.1 Llibres constitutius .....</i>	60
<i>8.2 Llibres consultius .....</i>	61
<i>8.3 Treballs acadèmics: TFG, TFM, Tesis, Seminaris .....</i>	61
<i>8.4 Articles.....</i>	61
<b>9. Annexos .....</b>	62
<b>ANNEX1 CODI R COMENTAT .....</b>	62
<b>1. REDUCCIÓ D'UN PATHWAY SENSE CICLES A ARBRE .....</b>	64
<i>1.1 Representació del pathway o network biològic (xarxa de dependències direccionals) .....</i>	64
Figura 1.....	64
<i>1.2 Garantir l'aciclicitat general abans de reduir el pathway a arbre/bosc .....</i>	66
Figura 2.....	67
Taules 1 <impliedConditionalIndependencies & localTests> .....	68
<b>2. SUBGRAF VERTICAL .....</b>	72
<i>2.1 Generació de l'arbre o bosc (segons sigui un únic node arrel o més d'un) .....</i>	72
Figura 4.....	74
Figura5 <subgraf bosc vertical dirigit sense col·lididors-fulla>.....	75

<b>3. SUBGRAF RETROLATERAL .....</b>	76
Figura 7.....	79
Taula 2.....	81
Taula 3.....	83
<b>4. ANÀLISI DELS SUBGRAFS VERTICAL I RETROLATERAL.....</b>	84
<b>Introducció.....</b>	84
4.1 Vertical amb tots dos col·lididors-fulla inclosos.....	85
Taula 5 <Nombre de parelles localTest()> .....	85
Taula 6 <impliedConditionalIndependencies()> .....	86
4.2 Vertical sense cap col·lididors-fulla .....	90
4.3 Retrolateral .....	93
4.4 Retrovertical (contraflux vertical) .....	96
4.5 General.....	97
<b>5. MODEL CAUSAL GEOMÈTRIC 3D .....</b>	100
<b>5.1 SUBGRAF VERTICAL TOPOLOGIC.....</b>	101
5.1.1 Model d'arbre sense els eixos normalitzats .....	101
Taula 7 .....	101
Figura 6.....	102
5.1.2 Model d'arbre amb els eixos normalitzats.....	103
Taula 3 .....	103
<b>5.2 SUBGRAF RETROLATERAL TOPOLOGIC .....</b>	106
<b>5.3 MODEL GENERAL TOPOLOGIC .....</b>	112
5.3.1 Model general topològic sense els eixos normalitzats .....	112
5.3.2 Model general topològic amb els eixos normalitzats .....	115
Taula 11.....	115
<b>CADENA DE TRANSFORMACIONS GEOMÈTRIQUES A PARTIR DE LES DADES .....</b>	120
Normalització de la distribució espacial causal topològica (escalatge de coordenades).....	120
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....	121
==>NORMALITZACIÓ RADIAL: PRIMERA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI.	
Mitjana/desviació tipus global .....	121
==>NORMALITZACIÓ INDIVIDUAL: SEGONA OPERACIÓ DE REUBICACIÓ DE NODES A L'ESPAI.	
Correlació .....	121

<i>JUSTIFICACIÓ D'ÚS DE LA CORRELACIÓ A LA SEGONA TRANSFORMACIÓ GEOMÈTRICA: l'ús de la funció localTest()</i> .....	122
<i>Normalització vectorial: no hi és inclosa</i> .....	123
<i>LA METÀFORA BAYESIANA ESTRUCTURAL: empírica estructural a priori i empírica quantitativa a posteriori</i> .....	123
<i>5.4.1 Model general topoempíric amb la primera transformació geomètrica node-radial de mu/sigma_global</i> .....	124
Figura 14.....	126
<i>5.4.2 Model general topoempíric amb la segona projecció aresta-local de correlacions</i> .....	127
Taula 14.....	129
Taula 15 <coords3Dn_diag_corr1: reajust de la llargària de cada aresta segons la correlació>	129
Figura 15.....	131
<i>5.5 FLUX CAUSAL DEL MODEL GENERAL TOPOEMPÍRIC AMB DUES TRANSFORMACIONS GEOMÈTRIQUES: LA MITJANA VECTORIAL GLOBAL</i> .....	133
<i>Introducció</i> .....	133
<i>A. MITJANA DELS VECTORS DIRECTORS SOLS RESPECTE DE Z</i> .....	133
<i>B. MITJANA DELS VECTORS DIRECTORS RESPECTE DE Z, X i Y: fer la mitjana vectorial global de tots els vectors directors provinents de tots els nodes fulla del subgraf retrolateral: siguin pseudoverticals (les arestes rebutjades en descompondre els col·lididors fulla per aplicació de la definició d'arbre/bosc), siguin retroverticals o siguin lateroverticals al model geomètric 3D.</i> .....	137
<i>C. D'ALTRES PER ARA REBUTJATS</i> .....	137
<i>DUES VIES DE VALIDACIÓ LOCAL DEL MODEL CAUSAL GEOMÈTRIC</i> .....	137
<i>5.6 ASSAIG DE VALIDACIÓ SIMULADA A ESCALA LOCAL A PARTIR DEL TIPUS CAUSALS MÍNIMS PROVINTS DEL MODEL GENERAL TOPOEMPÍRIC (per mitjà de LocalTest() del DAG combinatori versus la diferència d'angles dels fluxos locals θ_flux_SENSE_CONDICIONAMENT i θ_flux_AMB_CONDICIONAMENT</i> .....	138
<i>IMPORTANT: És una pràctica simulada no real perquè les dades empíriques aquí emprades no responen a un model de DAG combinatori de 3 nodes i dues arestes, hom tria els tipus causals simples per tal d'assajar casos triviais abans de fer la validació local real del model causal geomètric 3D. És talment un assaig per tal de verificar i retocar el procediment per mitjà de tipus causals mínims (cadena, col·lididor...).</i> .....	138
<i>5.6.1 INCLINACIÓ RESPECTE L'EIX Z: FUNCIONS DE CÀLCUL D'ANGLES DELS TIPUS CADENA i COL·LIDIDOR i TESTS D'ASSAIGS</i> .....	138
<i>càcul del vector director segons sigui quin sigui el tipus causal: cadena, col·lididor o forca</i> .....	138
<i>5.6.1 INCLINACIÓ RESPECTE L'EIX XY: FUNCIONS DE CÀLCUL D'ANGLES DELS TIPUS CADENA i COL·LIDIDOR</i> .....	145
<i>5.6.2 SUBGRAF AÏLLAT: CADENA VERTICAL DE N=3 NODES AMB TEST NEGATIU</i> .....	147

<b>5.6.3 SUBGRAF AïLLAT: CADENA LATERAL DE N=3 NODES AMB TEST NEGATIU .....</b>	159
<b>5.6.4 BATERIA DE SUBGRAFS AïLLATS de 3 NODES AMB TEST POSITIU .....</b>	170
P38    PIP3   PKC (plantejable com a cadena PIP3->...>PKC->P38) .....	171
Erk    PIP2   PKC (plantejable com a cadena (?) PIP2->PKC[>]Erk) .....	176
Akt    PKC   PIP3 (plantejable com a cadena (?) PIP3->...>PKC[>]Akt).....	178
Mek    PIP3   PKC (plantejable com a cadena PIP3->...>PKC->Mek) .....	180
<b>5.6.5 CONCLUSIONS EN VALIDACIÓ TEÒRICA PER TIPUS CAUSALS MÍNIMS (no encara sobre el model geomètric 3D).....</b>	183
<b>CONCEPTE DE CONVERGÈNCIA.....</b>	184
<b>5.6.6 ASSAIG DE VALIDACIÓ SIMULADA A ESCALA LOCAL EN CAMINS DE MÉS DE TRES NODES .....</b>	184
<b>5.7 ASSAIG DE VALIDACIÓ LOCAL (l'aresta absent entre Akt i Erk segons reporten SACHS et al.) A PARTIR DEL FLUX CAUSAL GENERAL .....</b>	184
<b>Introducció .....</b>	184
<b>5.7.1 FLUX CAUSAL LOCAL de la cadena lateral en el marc del model geomètric topoempíric general .....</b>	186
Figura 17 Mek->Erk> .....	186
<b>5.7.2 FLUX CAUSAL VERTICAL GENERAL del model geomètric topoempíric general.....</b>	190
Figura 18.....	190
<b>5.7.3 FLUX CAUSAL RETROVERTICAL del model geomètric topoempíric general.....</b>	195
<b>5.7.4 FLUX CAUSAL CORREGIT VERTICAL del model geomètric topoempíric general .....</b>	201
Figura 20.....	201
<b>5.8 Càlcul de l'angle de flux causal vertical (Z amb XY lliures) del model topoempíric:</b>	
theta_flux_vertical_corregit .....	210
FLUX CAUSAL VERTICAL (8 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1 (SENSE ARESTA Erk->Akt) .....	210
taula_fluxos verticals (i local).....	210
Taula 16.....	210
Taula 17 .....	212
taula_general_fluxos_fulla8 amb coords3Dn_diag_corr1 .....	216
Taula 18.....	216
interseccio_fluxos8 amb coords3Dn_diag_corr1 .....	217
Taula 19.....	217
DEFINICIONS CLAU .....	220
ASSUMPCIÓ ESFERA DE DIRECCIONS:.....	221

Figura 21 <Plot coords3Dn_diag_corr1 amb 8 vectors directors de flux SENSE ARESTA Erk->Akt>	225
.....	.....
<b>5.8 Càlcul de l'angle de flux causal holístic (ZYX) del model topoempíric: flux_fulla_corregit.....</b>	<b>236</b>
taula_general_fluxos_fulla4 amb coords3Dn_diag_corr1 .....	236
Taula 20.....	236
<b>5.8.1 FLUX CAUSAL holístic del model geomètric topoempíric general: FLUX CAUSAL ÚNIC EN L'ESFERA DE DIRECCIONS .....</b>	<b>240</b>
taula_fluxos_general.....	240
Taula 21.....	243
<b>FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1 (SENSE ARESTA Erk-&gt;Akt).....</b>	<b>244</b>
Hom força la inclusió de l'aresta absent Erk -> Akt .....	254
Taula 23 <matriu d'adjacència (dirigida)> .....	257
Taula 25.....	259
Figura 24.....	261
<b>5.8.2 Càlcul de les coordenades del flux causal holístic (ZYX) del model topoempíric: flux_fulla_corregit.....</b>	<b>263</b>
FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1_MOD (AMB ARESTA Erk->Akt).....	263
taula_general_fluxos_fulla4MOD amb coords3Dn_diag_corr1_MOD .....	263
Taula 26.....	263
interseccio_fluxos4MOD amb coords3Dn_diag_corr1_MOD .....	268
Taula 27 .....	268
Plot casquet esfèric causal MODEL *FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) .....	269
Figura 26 <Plot coords3Dn_diag_corr1_MOD amb 4 vectors directors de flux AMB ARESTA Erk->Akt> .....	269
FLUX CAUSAL RETROLATERAL (3 col·lididors fulla) AMB COORDENADES coords3Dn_diag_corr1_MOD (AMB ARESTA Erk->Akt).....	279
taula_general_fluxos_fulla3MOD amb coords3Dn_diag_corr1_MOD .....	279
Taula 28.....	283
interseccio_fluxos amb coords3Dn_diag_corr1_MOD.....	285
Taula 29.....	285
Plot casquet esfèric causal MODEL *FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) .....	286
Figura 27 <Plot coords3Dn_diag_corr1_MOD amb 3 vectors directors de flux AMB ARESTA Erk->Akt> .....	286

<b>ANNEX2 CODI R COMENTAT PER A LA VALIDACIÓ ESTRUCTURAL DEL MODEL CAUSAL GEOMÈTRIC: TOR FUNCIONAL DEFORMABLE .....</b>	296
<i>Model de tor empíric deformable .....</i>	297
<i>PRIMER PLEGAMENT HORITZONTAL .....</i>	297
<i>SEGON PLEGAMENT VERTICAL .....</i>	298
<i>Model toroïdal acíclic amb retrofluxos .....</i>	298
Figura 28.....	298
<b>ANNEX3 CODI R COMENTAT SOBRE LA VALIDACIÓ DEL DAG_combinatori: PER MITJÀ D'APRENENTATGE ESTADÍSTIC O MACHINE LEARNING .....</b>	301
<i>VALIDACIÓ LOCAL .....</i>	310
l'antecedent: DAGSVM .....	310
<i>VALIDACIÓ GLOBAL: CNN com a validació empírica complementària d'un DAG.....</i>	317

## ÍNDEX DE FIGURES

<i>Figura 1 .....</i>	<i>Error! Bookmark not defined.</i>
<i>Figura 2 .....</i>	<i>Error! Bookmark not defined.</i>

Taules 1 <impliedConditionalIndependencies & localTests>**Error! Bookmark not defined.**

Figura 4 ..... **Error! Bookmark not defined.**

Figura5 <subgraf bosc vertical dirigit sense col·lididors-fulla>**Error! Bookmark not defined.**

Figura 7 ..... **Error! Bookmark not defined.**

Taula 2 ..... **Error! Bookmark not defined.**

Taula 3 ..... **Error! Bookmark not defined.**

Taula 5 <Nombre de parelles localTest()>..... **Error! Bookmark not defined.**

Taula 6 <impliedConditionalIndependencies()>**Error! Bookmark not defined.**

Taula 7 ..... **Error! Bookmark not defined.**

Figura 6 ..... **Error! Bookmark not defined.**

Taula 3 ..... **Error! Bookmark not defined.**

Taula 11 ..... **Error! Bookmark not defined.**

Figura 14 ..... **Error! Bookmark not defined.**

Taula 14 ..... **Error! Bookmark not defined.**

Taula 15 <coords3Dn\_diag\_corr1: reajust de la llargària de cada aresta segons la correlació> ..... **Error! Bookmark not defined.**

Figura 15 ..... **Error! Bookmark not defined.**

Figura 17 Mek->Erk>..... **Error! Bookmark not defined.**

Figura 20 ..... **Error! Bookmark not defined.**

coords3Dn\_diag\_corr1 (SENSE ARESTA Erk->Akt)**Error! Bookmark not defined.**

taula\_fluxos\_verticals (i local)..... **Error! Bookmark not defined.**

Taula 16 ..... **Error! Bookmark not defined.**

Taula 17 ..... **Error! Bookmark not defined.**

taula\_general\_fluxos\_fulla8 amb coords3Dn\_diag\_corr1 **Error! Bookmark not defined.**

Taula 18 ..... **Error! Bookmark not defined.**

interseccio\_fluxos8 amb coords3Dn\_diag\_corr1 **Error! Bookmark not defined.**

Taula 19 ..... **Error! Bookmark not defined.**

*Figura 21 <Plot coords3Dn\_diag\_corr1 amb 8 vectors directors de flux SENSE ARESTA Erk->Akt> ..... Error! Bookmark not defined.*

*taula\_general\_fluxos\_fulla4 amb coords3Dn\_diag\_corr1 Error! Bookmark not defined.*

*Taula 20 ..... Error! Bookmark not defined.*

*taula\_fluxos\_general..... Error! Bookmark not defined.*

*Taula 21 ..... Error! Bookmark not defined.*

*coords3Dn\_diag\_corr1 (SENSE ARESTA Erk->Akt)Error! Bookmark not defined.*

*Taula 23 <matriu d'adjacència (dirigida)> ..... Error! Bookmark not defined.*

*Taula 25 ..... Error! Bookmark not defined.*

*Figura 24 ..... Error! Bookmark not defined.*

*coords3Dn\_diag\_corr1\_MOD (AMB ARESTA Erk->Akt)Error! Bookmark not defined.*

*taula\_general\_fluxos\_fulla4MOD amb coords3Dn\_diag\_corr1\_MOD .... Error!*

*Bookmark not defined.*

*Taula 26 ..... Error! Bookmark not defined.*

*interseccio\_fluxos4MOD amb coords3Dn\_diag\_corr1\_MODError! Bookmark not*

*defined.*

*Taula 27 ..... Error! Bookmark not defined.*

*Plot casquet esfèric causal MODEL \*FLUX CAUSAL RETROLATERAL (4 col·lididors*

*fulla) ..... Error! Bookmark not defined.*

*Figura 26 <Plot coords3Dn\_diag\_corr1\_MOD amb 4 vectors directors de flux AMB*

*ARESTA Erk->Akt> ..... Error! Bookmark not defined.*

*coords3Dn\_diag\_corr1\_MOD (AMB ARESTA Erk->Akt)Error! Bookmark not defined.*

*taula\_general\_fluxos\_fulla3MOD amb coords3Dn\_diag\_corr1\_MOD .... Error!*

*Bookmark not defined.*

*Taula 28 ..... Error! Bookmark not defined.*

*interseccio\_fluxos amb coords3Dn\_diag\_corr1\_MODError! Bookmark not*

*defined.*

*Taula 29 ..... Error! Bookmark not defined.*

*Plot casquet esfèric causal MODEL \*FLUX CAUSAL RETROLATERAL (4 col·lididors*

*fulla) ..... Error! Bookmark not defined.*

*Figura 27 <Plot coords3Dn\_diag\_corr1\_MOD amb 3 vectors directors de flux AMB ARESTA Erk->Akt> ..... Error! Bookmark not defined.*

*Figura 28 ..... Error! Bookmark not defined.*

# TESI DE MÀSTER EN BIOESTADÍSTICA I BIOINFORMÀTICA

## UOC-UB

### Abstract / Extracte

In this work, we present a Geometric Causal Model, a framework for the causal analysis of biological pathways through their systematic conversion into a combinatorial DAG and subsequent embedding in a three-dimensional geometric space. Within this framework, nodes are represented by coordinates defined by causal dependency, causal effect, and hierarchical causal depth, enabling causal relationships to be analyzed in terms of directions, angles, and magnitudes of causal flow.

The proposed methodology starts from a signaling pathway and proceeds with a graph-theoretical decomposition of the resulting Directed Acyclic Graph (hereinafter DAG) into vertical (hierarchical) and retrolateral (cross-edge) subgraphs. These structures are then transformed into a three-dimensional topological representation and into a topoempirical geometric model by incorporating statistical information derived from empirical data through controlled geometric transformations.

Causal validation is performed through direct comparison with the gold-standard study of Sachs et al. (2005), focusing on two competing structural scenarios: one in which a previously reported absent edge is enforced and another in which it remains excluded. The impact of this choice is assessed by analyzing changes in local conditional independence patterns and in the structural coherence of the inferred causal graph.

The geometric embedding further enables a topological closure of causal directions through a toroidal representation (torification), which provides a criterion for evaluating global causal coherence. This closure is not a trivial property, but emerges from the joint configuration of the graph, particularly the organization of the retrolateral subgraph. The causal flow remains globally directed and confined within the modeled functional domain, without introducing external dependencies or causal loops.

---

En aquest treball hom presenta un model causal geomètric, un marc per analitzar causalment pathways mitjançant la seva reconversió en un DAG combinatori i la seva inclusió en un espai geomètric tridimensional. En aquest espai, els nodes es representen mitjançant coordenades definides per la dependència causal, l'efecte causal i la profunditat causal jeràrquica, cosa que permet analitzar les relacions causals en termes de direccions, angles i magnituds del flux causal.

La metodologia proposada parteix d'un pathway de senyalització biològica i continua amb una descomposició segons la teoria de grafs del Graf Acíclic Dirigit resultant (d'ara endavant DAG) en subgrafs vertical (jeràrquic) i retrolateral (arestes transversals). Aquestes estructures són transformades després en una representació topològica tridimensional i, a la fi, en un model geomètric topoempíric mitjançant la incorporació d'estadístics derivats de les dades empíriques de l'assaig mitjançant transformacions geomètriques controlades.

La validació causal es duu a terme mitjançant una comparació directa amb els resultats de l'estudi de referència (gold standard) de Sachs et al. (2005), considerant dos escenaris estructurals competidors: un en què s'imposa una aresta prèviament considerada absent i un altre en què aquesta aresta roman exclosa. L'impacte d'aquesta decisió s'avalua mitjançant l'anàlisi dels patrons locals d'independència condicional i de la coherència estructural del graf causal inferit.

La inclusió geomètrica permet, a més, el tancament topològic de les direccions causals mitjançant una representació toroidal (torització), que actua com a criteri geomètric per avaluar la coherència causal global del model. El flux causal roman globalment dirigit i contingut dins del domini funcional modelitzat, sense generar dependències causals externes ni bucles causals.

## #1. Introducció

### ##1.1. Context i justificació del Treball

Els pathways de senyalització biològica constitueixen una eina fonamental per descriure i sintetitzar el coneixement existent sobre els mecanismes moleculars que governen els sistemes biològics. Aquests esquemes integren informació experimental acumulada al llarg del temps i permeten representar, de manera visual i estructurada, les interaccions entre proteïnes, gens o metabòlits implicats en processos cel·lulars complexos. Tanmateix, els pathways tenen un caràcter essencialment descriptiu i no són concebuts, en el seu format original, com a models formals per a la inferència causal.

En particular, els pathways soLEN incloure cicles, dependències implícites i relacions non-explicades que dificulten l'avaluació rigorosa de la causalitat a partir de dades empíriques. Això limita la seva utilitat quan l'objectiu no és sols descriure un sistema, sinó analitzar-ne la coherència causal, contrastar hipòtesis estructurals o avaluar la compatibilitat entre una estructura proposada i les dades observades. En aquest context, els Grafs Acíclics Dirigits (DAGs) forneixen un marc formal sólid per a la representació causal, però la seva aplicació directa a pathways biològics no és trivial i requereix un procés de conversió estructurada i justificable.

D'altra banda, les aproximacions causals clàssiques basades en DAGs tendeixen a centrar-se en propietats combinatòries o estadístiques locals, i sovint ofereixen una visió limitada de la coherència global de l'estructura causal. Això fa difícil captar propietats emergents del sistema o interpretar el comportament causal com un conjunt integrat, especialment en xarxes complexes amb múltiples nivells jeràrquics i dependències transversals.

En aquest treball hom parteix de la necessitat de disposar d'un marc que permeti formalitzar causalment els pathways i, alhora, oferir una lectura global i interpretable de la seva estructura. Amb aquest objectiu, es proposa un model causal geomètric que reconverteix un pathway en un DAG combinatori i el representa a dins d'un espai geomètric tridimensional, on la causalitat pot analitzar-se en termes de direccions, angles i fluxos. Aquesta aproximació permet integrar informació estructural i empírica, així com explorar criteris de coherència causal que van més enllà de l'anàlisi local, aportant una nova perspectiva per a l'estudi causal de sistemes biològics complexos.

### ##1.2. Objectius del Treball

## *Objectiu general*

Desenvolupar i validar un mètode causal geomètric per a l'anàlisi de pathways (en aquest cas biològic), basat en la seva reconversió en un DAG combinatori i la seva inclusió a dins d'un espai geomètric tridimensional, amb l'objectiu d'avaluar la coherència causal tant local com global de l'estructura geomètrica resultant, tot assegurant-ne la reproductibilitat i reutilització mitjançant la publicació del codi i la documentació associada en format obert.

## *Objectiu nuclear de validació*

Validar empíricament i estructuralment el mètode causal geomètric proposat mitjançant la seva aplicació controlada al cas d'estudi de Sachs et al. (2005), emprat com a gold standard, comparant escenaris estructurals competidors i analitzant-ne la coherència causal mitjançant criteris estadístics, geomètrics i topològics.

## *Objectius específics*

1. Reconstruir un pathway de senyalització biològica com un DAG combinatori, garantint l'aciclicitat mitjançant la reducció sistemàtica de cicles i la definició d'estructures d'arbre o bosc segons la teoria de grafs.
2. Descompondre el DAG combinatori en subgrafts vertical i retrolateral, tot identificant el paper causal dels col·lididors-fulla i de les arestes transversals en l'organització del graf.
3. Analitzar estructuralment els subgrafts vertical i retrolateral, tant de manera independent com conjunta, abans de la incorporació d'informació empírica.
4. Construir una representació topològica tridimensional del DAG, assignant coordenades causals associades al grau d'entrada (dependència), grau de sortida (efecte) i nivell jeràrquic del flux causal.
5. Incloure informació empírica al model topològic mitjançant transformacions geomètriques controlades, incloent-hi la normalització de la distribució espacial causal i la integració d'estadístics derivats de les dades experimentals.
6. Definir i calcular vectors de flux causal locals i globals, així com els angles i mòduls associats, per caracteritzar quantitativament la direccionalitat causal del sistema.
7. Avaluar la coherència causal local del model mitjançant tests d'independència condicionada aplicats tant als subgrafts com al DAG global del pathway.
8. Comparar escenaris estructurals competidors, imposant o excloent una aresta considerada absent a l'estudi de referència, i analitzar-ne l'impacte causal, estructural i geomètric.
9. Analitzar el tancament topològic del flux causal mitjançant una representació toroidal, estudiant el paper del subgraf retrolateral en la coherència causal global del model.
10. Validar el DAG\_combinatori mitjançant tècniques d'aprenentatge estadístic i machine learning, diferenciant explícitament la validació del model de la interpretació causal dels resultats.

### *1.3. Impacte en sostenibilitat, ètic-social i de diversitat*

Aquest Treball Final és centrat en el desenvolupament i la validació d'un mètode formal per a l'anàlisi causal de pathways mitjançant una aproximació geomètrica i estructural. Es tracta, per tant, d'un treball de natura metodològica i conceptual, sense una aplicació directa immediata sobre persones o col·lectius específics. Tot i això, és pertinent analitzar-ne l'impacte potencial en les tres dimensions de la competència transversal de .

#### *Impacte en sostenibilitat*

El treball no comporta un impacte ambiental directe significatiu perquè es basa en essència en l'anàlisi computacional de dades ja existents i en el desenvolupament de codi. L'ús de recursos computacionals és moderat i és limitat a entorns estàndard de càlcul, sense requeriments d'infraestructura intensiva ni generació de residus físics. En aquest sentit, el treball s'alinea amb pràctiques de consum responsable de recursos computacionals.

De manera indirecta, el mètode proposat pot tenir un impacte positiu en sostenibilitat científica, en tant que fomenta la reutilització de dades, la formalització rigorosa del coneixement existent i la reducció de processos experimentals redundants mitjançant una millor validació causal prèvia. Aquest enfocament contribueix a una recerca més eficiente i alineada amb principis de sostenibilitat en la producció de coneixement.

#### *Impacte ètic-social i responsabilitat social*

Des del punt de vista ètic i de responsabilitat social, el treball respecta els principis de la pràctica científica responsable. Hom no empra dades personals ni sensibles, i totes les dades emprades provenen d'estudis publicats i àmpliament acceptats com a referència (gold standard). El mètode proposat no pretén substituir el judici científic ni clínic, sinó oferir una eina formal per avaluar la coherència causal de models, tot evitant interpretacions acrítiques o conclusions non-justificades per les dades.

A més a més, la distinció explícita entre validació del model i interpretació dels resultats causals constitueix un element clau de responsabilitat metodològica, ja que redueix el risc d'ús indegit o de sobreinterpretació dels resultats, especialment en contextos aplicats.

#### *Impacte en diversitat i drets humans*

El treball no introduceix impactes directes sobre col·lectius específics ni inclou mecanismes que puguin generar discriminació per raó de gènere, origen, orientació o altres dimensions de la diversitat. En tant que proposta metodològica general, el mètode és neutral respecte als subjectes d'estudi i aplicable a diferents dominis sense biaixos inherents.

En el disseny i la redacció del treball hom vetlla per un ús acurat del llenguatge acadèmic i inclusiu, així com per una citació rigorosa de les aportacions científiques, reconeixent l'autoria dels treballs previs. La disponibilitat del codi en obert afavoreix, a més, la democratització de l'accés a les eines metodològiques, contribuint a reduir desigualtats en l'accés al coneixement científic.

## *1.4. Enfocament i mètode seguit*

L'enfocament d'aquest treball és metodològic i exploratori, amb l'objectiu de desenvolupar i validar un nou marc formal d'anàlisi causal aplicable a pathways complexos. El treball no consisteix en l'aplicació d'un mètode existent, sinó en la construcció progressiva d'un producte metodològic nou, el model causal geomètric, a partir de fonaments de teoria de grafs, inferència causal i representació geomètrica.

Com a estratègia general, hom opta per reconvertir un pathway biològic existent en un objecte causal formal (DAG combinatori) i, a partir d'aquest, desenvolupar un procés sistemàtic d'anàlisi i validació en diverses etapes. Aquesta estratègia permet aprofitar el coneixement empíric acumulat en la literatura biomèdica —codificat en forma de pathway— i transformar-lo en una estructura adient per a la inferència causal rigorosa, tot respectant les restriccions formals dels DAGs, en particular l'aciclicitat.

El mètode és articulat en quatre grans fases:

- I. Conversió del pathway a un DAG combinatori, mitjançant l'eliminació sistemàtica de cicles i la imposició d'un ordre causal funcional coherent amb les dades disponibles. De manera complementària, el DAG obtingut —de naturalesa estrictament combinatòria— és sotmès a una validació multiescala mitjançant tècniques d'aprenentatge estadístic, emprades no pas per inferir relacions causals, sinó per avaluar la consistència global i la coherència estructural del graf.
- II. Descomposició del DAG combinatori segons la teoria de grafs en un subgraf vertical (jeràrquic) i un subgraf retrolateral (arestes transversals), amb l'objectiu d'analitzar de forma aïllada les diferents dimensions de la causalitat.
- III. Validació causal local, basada en tests d'independència condicionada, per avaluar la coherència empírica de les estructures causals proposades.
- IV. Construcció del model causal geomètric tridimensional, a través d'una immersió topoempírica en què la informació estructural heretada del pathway (DAG combinatori) i la informació empírica derivada de la validació causal s'integren de manera coherent, culminant en una assignació de coordenades topoempíriques com a expressió geomètrica final del model.

Aquest enfocament és en especial adient a fi d'assolir els objectius del treball, atès que permet connectar rigor formal, interpretabilitat causal i validació empírica, tot mantenint una clara separació entre coneixement causal estructural i informació estadística observacional.

## *1.5. Planificació del Treball*

Hom planifica aquest treball de manera estructurada al llarg del semestre, seguint les fites definides a les diferents PACs de l'assignatura; la planificació temporal és feta mitjançant un diagrama de Gantt, que permet de veure la seqüència de tasques, la seva durada i les dependències entre elles, així com fitar clarament els punts de lliurament parcials.

Els recursos emprats són l'entorn de programació R, llibreries estadístiques i de teoria de grafs, i eines de visualització geomètrica. Les dades són públiques, procedents de l'estudi de referència de Sachs et

al. (2005), emprat com a gold standard. I en darrer lloc la literatura científica sobre inferència causal, DAGs, teoria de grafs, pathways de senyalització i mètodes d'aprenentatge estadístic. Tot emmarcat en el seguiment acadèmic per part dels co-tutors del TFM.

La planificació de tasques i fites és desenvolupada en les fases següents, d'acord amb les PACs de l'assignatura, i reflectides explícitament en el diagrama de Gantt. La redacció de la memòria és duta a terme de manera transversal i en paral·lel a totes les PACs, amb l'objectiu de garantir coherència entre el desenvolupament metodològic, la validació del model i la presentació final dels resultats.

**PAC1:** definició del problema, contextualització i justificació del treball, formulació dels objectius generals i específics, definició de l'enfocament metodològic i planificació temporal del projecte mitjançant un diagrama de Gantt.

**PAC2:** reconversió del pathway de senyalització biològica en un DAG combinatori, eliminació sistemàtica de cicles, definició de l'estructura causal bàsica i descomposició del graf en subgrafs vertical (jeràrquic) i retrilateral (arestes transversals).

**PAC3:** construcció progressiva del model causal geomètric 3D, incloent-hi la representació topològica dels subgrafs i la definició inicial de coordenades topològiques.

A partir de les dades empíriques, hom calcula estadístics destinats a extreure la informació empírica essencial del sistema causal, que tot seguit hom empra per tal de definir transformacions geomètriques aplicades a les coordenades topològiques.

El resultat d'aquest procés és la construcció de coordenades topoempíriques, que inclouen a dins d'un únic espai geomètric tant l'estructura causal del pathway com la informació empírica continguda en les dades.

**PAC4:** priorització de la validació causal del mètode mitjançant la comparació explícita de dos escenaris estructurals competidors -amb i sense la inclusió de l'aresta Erk → Akt- tot emprant criteris geomètrics derivats del con i del casquet esferic causals. Aquesta fase és centrada en avaluar la coherència causal del model geomètric proposat.

**PAC5:** consolidació de la memòria escrita del TFM, integrant metodologia, resultats i discussió, i preparació de la presentació preliminar del treball.

Durant l'execució del projecte han sorgit diversos riscs potencials, als quals ha calgut donar resposta.

1. El desbordament temporal degut a la complexitat conceptual del model causal geomètric i del procés de validació multiescala, hom l'ha resolt redefinint l'abast del treball i la priorització del nucli de validació del mètode causal geomètric.
2. Les dificultats tècniques i computacionals associades a la implementació del model geomètric i al càlcul de magnituds derivades (fluxos, angles, cons i casquets causals) s'han abordat mitjançant un desenvolupament progressiu del codi i una validació rigorosa dels càlculs, basada en la contrastació sistemàtica dels resultats i en l'ús d'eines d'assistència computacional com a suport al raonament i a la verificació conceptual.

3. Les dificultats associades a l'aplicació d'un criteri metodològic definit a priori davant dels resultats empírics obtinguts -com en la comparació dels escenaris amb i sense l'aresta Erk -> Akt- són resoltes mitjançant la clarificació i precisió del criteri metodològic, sense modificar el marc del mètode. Aquest procés ha permès distingir entre intuïcions inicials i criteris formals, posant de manifest que és l'anàlisi rigorosa, i no pas la intuïció, la qui ha de guiar la formulació operativa del mètode. En cap cas els resultats empírics han determinat una modificació del mètode proposat, sinó que han servit per refinjar la seva formulació operativa i els criteris d'interpretació, tot mantenint intacte el marc metodològic general.
4. La possible limitació temporal en la fase final de redacció de la memòria ha estat anticipada mitjançant una estratègia de redacció progressiva, consistent en l'elaboració, per a cada fase del treball, de reports de treball específics adreçats als co-tutors, independents dels lliuraments oficials de les PACs. Aquests reports han servit com a base per a la posterior integració, revisió i refinament del contingut en la memòria definitiva.

#### Diagrama de Gantt definitiu

```
# --- Paquets necessaris ---
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.3

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.4.2

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

# --- Dades del calendari ---
gantt <- tribble(
  ~Tasca, ~Inici, ~Final,
  "PAC1: Proposta i planificació", as.Date("2025-09-15"), as.Date("2025-10-08"),
  "PAC2: Implementació del mètode (pathway a DAG) i
```

```

    validació multiescala DAG original", as.Date("2025-10-09"), as.Date("2025-11-05")
  ,
  "PAC3: Model geomètric causal 3D", as.Date("2025-11-06"), as.Date("2025-12-03"),
  "PAC4: Mètode d'avaluació causal i
  anàlisi dels resultats", as.Date("2025-12-04"), as.Date("2026-01-07"),
  "PAC5: Memòria i presentació preliminar", as.Date("2026-01-08"), as.Date("2026-01
-15"),
  "DEFENSA: Lliurament final i defensa", as.Date("2026-01-16"), as.Date("2026-01-30
"),
  # --- La memòria ocupa tot el període ---
  "Redacció de la memòria", as.Date("2025-09-15"), as.Date("2026-01-30")
)

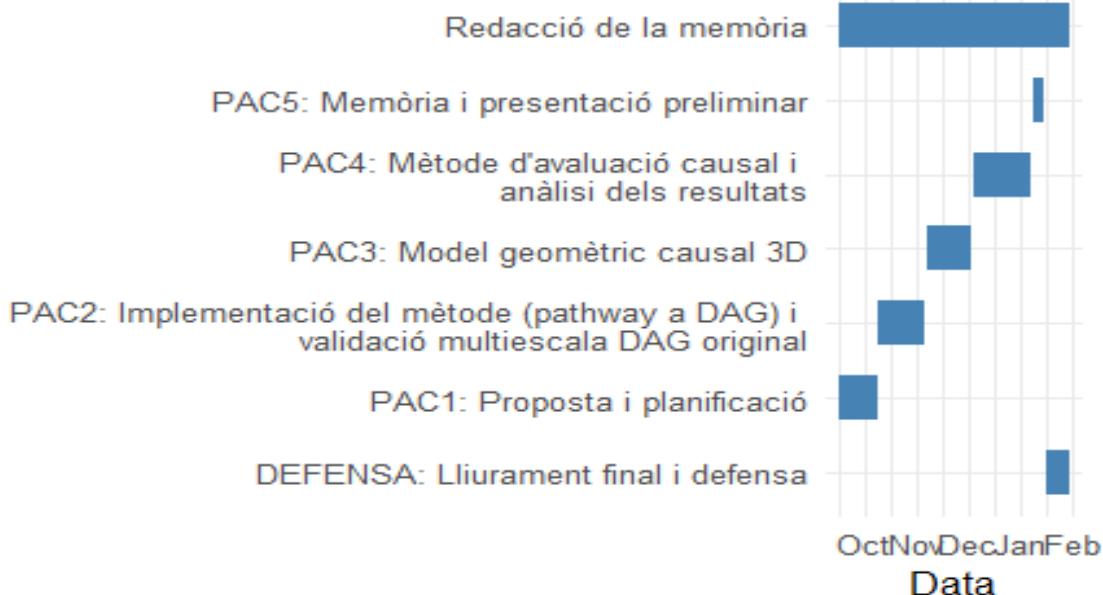
# --- Gràfic Gantt ---
ggplot(gantt, aes(y = Tasca, x = Inici, xend = Final, yend = Tasca,
  color = Tasca == "Redacció de la memòria (en paral·lel)")) +
  geom_segment(size = 8) +
  scale_color_manual(values = c("steelblue", "tomato")) +
  labs(
    title = "Diagrama de Gantt Definitiu",
    subtitle = "memòria en paral·lel a totes les PACs",
    x = "Data", y = NULL
  ) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    plot.title = element_text(face = "bold"),
    plot.margin = margin(t=10, r=30, b=10, l=20),
    axis.text.y = element_text(size = 10)
  )
}

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## [i] Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

## Diagrama de Gantt

memòria en paral·lel



### 1.6. Breu sumari de productes obtinguts

Els productes que hom materialitza gràcies al desenvolupament i a la validació del mètode causal geomètric proposat són els següents:

- Un DAG combinatori validat, derivat de la reconversió sistemàtica d'un pathway de senyalització biològica, amb identificació explícita de subgrafs complementàriament excloents vertical i retrorretrolateral.
- Un model causal geomètric tridimensional, que inclou la informació estructural del DAG i la informació empírica procedent de les dades experimentals, totes dues integrades en forma de coordenades topoempíriques.
- Una validació causal, que inclou validació local mitjançant tests d'independència condicionada (aquí mitjançant validació estructural a partir de la comparació d'escenaris competidors amb i sense l'aresta Erk → Akt) i validació global del model.
- Un aplec de criteris geomètrics de coherència causal, basats en el flux causal, els cons i casquets causals i el tancament toroidal de les direccions causals.
- Un codi computacional comentat, desenvolupat en R, que implementa totes les fases del mètode i permet la seva reproducció i extensió.

## 1.7. Breu descripció dels altres capítols de la memòria

La resta de la memòria és organitzada de manera progressiva per conduir el lector des del marc teòric fins als resultats i les conclusions del treball.

A <l'estat de l'art> hi ha una revisió crítica de la literatura rellevant en inferència causal, models basats en DAGs i pathways de senyalització biològica, així com de les principals limitacions dels enfocaments existents que motiven la proposta d'un model causal geomètric.

A s'hi descriu amb detall l'aplec de dades fetes servir, el procés de reconversió del pathway a un DAG combinatori, la descomposició en subgrafs, la construcció del model causal geomètric tridimensional i els procediments de validació causal i geomètrica emprats al llarg del treball.

A hom hi recull els resultats obtinguts en l'aplicació del mètode causal proposat, incloent-hi la comparació dels escenaris estructurals competidors, l'anàlisi del flux causal i els criteris geomètrics de coherència del model.

Al capítol de [hi ha la interpretació](#) dels resultats obtinguts en relació amb l'estat de l'art i amb l'estudi de referència emprat com a , tot posant de relleu les aportacions metodològiques del treball, les seves implicacions i les seves limitacions.

A hom hi sintetitza les principals conclusions del TFM i planteja possibles línies de continuació i extensió del mètode causal geomètric desenvolupat.

A la fi, als capítols , i [hi ha respectivament](#) la llista de termes emprats; la llista de cites de llibres, articles, tesis, enllaços diversos (font del joc de dades aquí emprat i materials complementaris necessaris per garantir la reproductibilitat del treball)... i el bloc final són els tres annexos (el codi de tot el procés del mètode causal geomètric, el codi per a la validació estructural del model i el codi per a la validació del DAG combinatori per mitjà de l'aprenentatge estadístic o machine learning).

## 2. Estat de l'art

L'anàlisi causal de sistemes complexos ha esdevingut un àmbit central tant en estadística com en bioinformàtica, també en l'estudi de xarxes de senyalització cel·lular. En aquest context, la comunitat científica ha consolidat els DAGs com una de les eines formals a fi de representar relacions causals entre variables, permetent descriure dependències o independències condicionades així com estructures diverses (en essència n'hi han tres: cadena  $a \rightarrow b \rightarrow c$ , forca  $a \leftarrow b \rightarrow c$  i col·lidor  $a \rightarrow b \leftarrow c$ ). El treball de Sachs et al. (2005) constitueix un punt de referència àmpliament acceptat, atès que proposa una xarxa de senyalització proteica inferida a partir de dades experimentals mitjançant tècniques probabilístiques basades en DAGs, i és habitualment considerat un en aquest àmbit.

Malgrat la seva potència formal, els DAGs clàssics presenten limitacions conegudes. En primer lloc, la representació és combinatòria i discreta, fet que dificulta la incorporació directa d'informació tant quantitativa procedent de dades empíriques com qualitativa dels mateixos pathways biomèdics. Aquesta separació dóna lloc, en la pràctica, a un treball fragmentat entre el pathway, el DAG i les dades empíriques, sense un marc unificat que en permeti una integració estructural.

En segon lloc, l'avaluació de la coherència global d'un model causal sovint és reduït a criteris locals, com ara els tests d'independència condicionada, o bé a criteris de validació predictiva. En models probabilístics, com els DAGs bayesians, hom empra sovint mesures com la log-likelihood, el BIC o l'AIC a fi d'avaluar fins a quin punt el model s'adapta a les dades observades; tanmateix, aquests criteris mesuren l'ajust probabilístic del model, no la correcció de la seva estructura causal. De manera anàloga, en la validació predictiva hom avalua la capacitat del model per predir les dades (per exemple, mitjançant errors de predicció), però no se'n deriva una interpretació causal explícita.

La inferència causal clàssica, basada en tests locals d'independència condicionada —com els tests de Fisher o de khi-quadrat—, es fonamenta així en criteris fragmentaris o combinatoris, sense fornir d'una noció geomètrica ni estructural del model com a totalitat. En conseqüència, manca sovint una definició clara de consistència estructural a escala global, fet que limita l'anàlisi del sistema causal més enllà de verificacions locals. Aquesta limitació posa de manifest la necessitat d'eines holístiques capaces de captar propietats estructurals del conjunt del model, com ara les que es deriven de la torització del model geomètric en el seu estadi topològic.

Finalment, la comparació entre escenaris estructurals alternatius —com la presència o absència d'una aresta concreta— acostuma a dependre de criteris estadístics puntuals, amb dificultats per integrar-ne una lectura estructural unificada dins d'un mateix marc causal.

En paral·lel, s'han desenvolupat aproximacions basades en aprenentatge automàtic, com les màquines de vectors de suport i les xarxes neuronals, que en aquest treball s'utilitzen de manera explícita com a eines de validació empírica direccional del DAG combinatori derivat del pathway. Aquestes tècniques no s'empren per inferir causalitat, sinó per avaluar la consistència global i la plausibilitat direccional de l'estructura causal proposada a partir de les dades, abans de la seva projecció en l'espai geomètric tridimensional.

D'altra banda, diverses línies de recerca han explorat la representació geomètrica o topològica de dades complexes, especialment en reducció de dimensionalitat i visualització. Tot i l'existència d'aproximacions que combinen inferència causal amb mètodes estadístics o d'aprenentatge automàtic, així com de treballs en representació geomètrica o topològica de dades complexes, aquests enfocaments rarament s'han integrat de manera sistemàtica en un marc únic que incorpori alhora estructura causal formal, validació empírica i criteris geomètrics globals aplicats al mateix objecte causal. *En una gran part dels casos, la geometria actua com un recurs instrumental o descriptiu; en contrast, el marc proposat en aquest treball assumeix una geometria de caràcter ontològic, en què la topologia forma part constitutiva de la representació causal.*

En aquest context, emergeix la necessitat d'un marc metodològic inclusiu que permeti combinar: (i) l'estructura causal formal dels DAGs provinent d'un pathway (ii) la informació quantitativa procedent de dades empíriques (iii) una noció de coherència global del model que vagi més enllà de validacions locals o purament predictives

Aquest treball s'inscriu en aquest buit metodològic i proposa un model causal geomètric que reconverteix un pathway biològic en un DAG combinatori, per esser projectat tot seguit a dins d'un

espai geomètric tridimensional, on la causalitat pot esser analitzada en termes de direccions, angles, magnituds i propietats geomètriques.

A partir d'aquest estat de l'art, hom formula les hipòtesis centrals del treball gràcies a les possibilitats que la representació geomètrica permet:

(i) detectar inconsistències estructurals no evidents en el DAG combinatori sense haver de recórrer exclusivament a tests locals externs al DAG (ii) comparar escenaris causals alternatius de manera més directa i informativa mitjançant intervencions directes sobre el model geomètric (iii) avaluar la coherència global del model causal mitjançant criteris geomètrics independents dels tests estadístics clàssics, per exemple, la torització del model geomètric o l'usatge de machine learning (encara que aquí sols hom assaja l'aprenentatge estadístic a la fase de DAG combinatori, abans de fer el salt cap a l'espai geomètric 3D).

### *3. Materials i mètode: projecció i transformació geomètrica (topològica i empírica)*

L'objectiu en aquesta secció no és sols detallar els materials emprats i el disseny metodològic, sinó explicitar el procés de transformació causal així com les decisions preses al llarg del desenvolupament del Model Causal Geomètric, des del pathway biològic fins al model geomètric topoempíric final i els criteris emprats per validar-lo.

Com a materials hom entén el pathway biològic de senyalització proteica proposat per Sachs et al. (2005) i les dades empíriques que corresponen a mesures experimentals de fosforilació proteica, emprades per tal de validar empíricament les estructures causals considerades.

La implementació s'ha dut a terme principalment en R, fent servir llibreries per a representació i manipulació de DAGs, càlcul de tests d'independència condicionada, transformacions geomètriques i vectorials, visualització topològica i geomètrica en 3D.

El pathway és una representació descriptiva mecànica, diu què interactua amb què segons el coneixement biològic; accepta cicles, retroalimentacions, paral·lelismes i redundàncies... no és pensat per fer inferència formal directa, respon a la manera com rutlla el sistema segons el coneixement actual. Les arestes dirigides són polisèmiques, no tenen un significat únic.

En canvi el DAG combinatori no descriu mecanismes, no li cal reproduir fidelment la dinàmica biomèdica completa sinó esser un objecte formal adient per fer inferència causal. El DAG (combinatori) no és ni geomètric ni topològic per natura, és una estructura combinatòria derivada del pathway.

També és important assenyalar que un DAG inclou absències, si no hi ha una aresta, això és una hipòtesi forta: no existeix cap efecte causal directe. El DAG és una estructura combinatòria que codifica relacions causals qualitatives independentament de qualsevol representació geomètrica o mètrica; segons Pearl <the causal graph [DAG] represents qualitative causal assumptions>. A un DAG, les arestes dirigides són úniques, per exemple, en (A->B) A sols és una causa directa potencial de B.

En conclusió: el Pathway és una mena de coneixement mecànic, descriptiu per natura, és fet per tal d'entendre processos com a finalitat i el lligam amb les dades és il·lustrativa; en canvi, el DAG és una

mena de coneixement causal, axiomàtic per natura, és fet per tal d'inferir efectes com a finalitat i el lligam amb les dades és operativa.

El salt del coneixement maquinal brut (pathway) a relacions causals mínimes i coherentes (DAG combinatori) —aquí és fàcil perquè el mateix article de Sachs et al. ja reportava el DAG combinatori— és fet imposant l'única restricció formal indispensable: l'absència de cicles causals a fi que el gràfic sigui d'ús per a fer inferència causal; el DAG imposa un ordre causal. Hom no ha corregit ni redundàncies, ni paralelismes, ni graus\_in, ni graus\_out... perquè —tot i que el DAG combinatori tendeix a esser abstracte i qualitatiu— no és incompatible d'entrada amb un DAG i pot aportar informació causal. Tot i que l'única restricció explícita és l'aciclicitat, el model assumeix que:

-> hi ha un ordre causal dominant, en què no manquen causes comunes rellevants (la suficiència causal de Pearl) -> la intervenció en un node no canvia la forma de la resta de relacions, sols els valors -> la causalitat es transmet per les arestes presents -> la topologia del pathway conté la informació causal essencial Totes aquestes assumpcions són controlades en certa mesura per mitjà de comparances d'escenaris, del con causal (hom en parla més endavant) i de l'anàlisi de col·lididors i camins actius.

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. L'espai empíric és formalment  $R^{11}$  (les dades crues tenen 11 dimensions) però causalment no pas perquè aquestes dimensions no són independentes, són del tot estructurades per un DAG\_combinatori causal, cosa que explana que una gran part de la informació és redundant, condicionada o col·lapsable. Per tant, l'espai causal efectiu no és  $R^{11}$  encara que les dades crues en siguin. Quan hom basteix l'objecte topològic (DAG\_combinatori derivat al seu torn del pathway), aquest ja no viu a  $R^{11}$  sinó amb relació a relacions, dependències, jerarquies, col·lididors... aquest objecte ja no és un espai vectorial.

Tanmateix el model geomètric introduceix una estructura que redueix les n dimencions a 3 a fi d'atènyer una geometria 3D interpretable, 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\_combinatori derivat del pathway).

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).

La projecció justament fa que cada node adquireixi coordenades en 3D: ara cada node del DAG geomètric representa una variable observada (per exemple, una proteïna, un gen, ...) a dins d'un espai causal geomètricament significatiu, interpretable i cada aresta una relació de dependència. Cada punt reflecteix el grau d'entrada (dependència causal), el grau de sortida (efecte causal), la seva jerarquia/centralitat, les relacions retrolateroverticals.

En resum, havent feta la immersió, cada node del DAG queda associat a un punt concret a dins de l'espai tridimensional (X, Y, Z). Aquestes coordenades no provenen de dades empíriques crues, sinó de les propietats causals derivades de l'estructura del DAG: grau d'entrada, grau de sortida, profunditat jeràrquica i contribucions retrolateroverticals. Així, la lectura topològica crea l'espai geomètric; i, a dins d'aquest espai, cada node ocupa la seva posició natural segons el seu rol causal. El model no redueix directament l'espai empíric de  $R^{11}$  a  $R^3$  sinó que primer infereix una estructura causal directa (DAG\_combinatori) i tot seguit en construeix una representació tridimensional basada en propietats causals.

#3.1 DAG i TRÍADA EN MACHINE LEARNING L'enfocament metodològic adoptat consisteix en el desenvolupament d'un mètode nou, no pas en l'adaptació d'un producte existent. El treball es fonamenta en una seqüència de transformacions causals i geomètriques explícites, estructurades en etapes successives i conceptualment diferenciades:

El pathway biològic inicial és reconvertit en un DAG combinatori, mitjançant l'eliminació de cicles, la definició d'un ordre causal funcional coherent amb el coneixement biològic i amb les dades. Aquest DAG constitueix l'objecte causal formal de partida. Havent obtingut el DAG combinatori, hom pot emprar les tècniques d'aprenentatge automàtic (SVM, xarxes neuronals) com a mecanisme d'anàlisi validació del DAG (qui no són emprades per tal d'inferir causalitat, sinó per avaluar la consistència global del model estructural).

-> *RANDOM FORESTS (RF): VALIDACIÓ DE NODES FULLA* Hom pretén detectar variables estructuralment irrelevants o redundants, és a dir, identificar nodes que o bé són fulles febles o bé col·lididors sense impacte empíric. Hom no ha pogut desenvolupar la validació de nodes però la idea és validar els nodes essencials per al flux causal global aquí anomenats col·lididor fulla.

-> *SUPPORT VECTOR MACHINE (SVM): VALIDACIÓ D'ARESTES A ESCALA LOCAL* Hom ha fet servir els models d'aprenentatge estadístic per tal de construir una validació empírica creuada entre la inferència causal formal (DAG i tests d'independència condicionada) i validació empírica predictiva local (SVM). On hi ha convergència entre el test d'independència condicionada i la validació empírica via SVM (millora predictiva significativa), la relació causal proposada al DAG sembla esser robusta. És a dir, el test causal defineix l'escenari i el SVM evalua si resta informació empírica non-explicada. Ras i curt, aquí hom fa el contrast empíric del localTest, no pas com a substitut del DAG.

La convergència entre els resultats dels tests causals i la millora predictiva observada en els models SVM reforça la plausibilitat causal de l'estructura proposada. En canvi, qualsevol divergència entre ambdós resultats és interpretada com un senyal analític rellevant que obre la porta a una revisió estructural del DAG o a una anàlisi més profundida de la natura de la dependència observada, més que no pas com un error del procediment.

-> *CONVOLUTIONAL NEURAL NETWORK (CNN): VALIDACIÓ D'ARESTES A ESCALA GLOBAL* Amb un model CNN1 constret segons l'estructura del DAG i un segon model CNN2 segons el mateix DAG però saturat: si una arquitectura neuronal constreta per l'estructura del DAG pot esser definida amb una complexitat paramètrica equivalent a la d'un model saturat, sense necessitat d'introduir camins funcionals addicionals, llavors aquest resultat dóna suport a la coherència estructural global del DAG des del punt de vista de la seva implementabilitat funcional.

-> *INTEGRACIÓ MULTIESCALA* Aquest punt és clau, hom fa una lectura integral dels resultats obtinguts a les tres escales: si una relació falla a totes les escales, aquesta és sospitosa; però si la coincidència transversal és molt informativa... és a dir, si és robusta en cada nivell (variable:nodes> local:arestes> global:xarxa), llavors és un DAG estructuralment sólid; a més a més, aquesta lectura integral pot esser d'ajut en la definició d'una causalitat com a estructura de diversos nivells.

#3.2 *SUBGRAFS* El DAG combinatori és descompost segons la teoria de grafs en un subgraf vertical, que recull l'estructura jeràrquica causal, i en un subgraf retrolateral, que articula les dependències transversals i col·lididors fulla. Aquesta descomposició és clau per a l'anàlisi estructural i geomètrica que en segueix.

La coherència causal local del DAG és validada mitjançant tests d'independència condicionada, aplicats al DAG combinatori global. Aquests tests permeten contrastar empíricament la plausibilitat de les relacions causals proposades a partir de les dades, d'acord amb els criteris clàssics de la inferència causal local.

Tanmateix, els subgrafs derivats del DAG (vertical i retrolateral) no són validats directament mitjançant tests estadístics, sinó que hom els analitza a partir de la seva estructura causal mínima, altrettament dit, identificant patrons elementals (cadena, forca i col·lidor) i sobretot el paper funcional dels nodes fulla (col·lididors o no). Aquesta anàlisi estructural permet caracteritzar el comportament causal local dels subgrafs sense reduir-los a contrasts estadístics puntuals.

#3.3 *FLUX CAUSAL i FLUX LOCAL* A més, la validació causal local inclou explícitament la distinció entre flux causal global i flux local. El flux local es defineix a partir de vectors direccionals associats a relacions causals concretes dins dels subgrafs, mentre que el flux causal global emergeix de la integració d'aquests vectors locals en el model geomètric. Aquesta distinció permet detectar discrepàncies entre la coherència local de determinades relacions i el comportament causal del sistema en conjunt.

D'aquesta manera, la validació causal local combina criteris estadístics clàssics aplicats al DAG global amb una anàlisi estructural i direccional dels subgrafs, establint el punt de partida necessari per a la validació geomètrica i global del model causal.

#3.4 *DUES MENES DE FLUX CAUSAL GLOBAL: O BÉ EL VERTICAL O BÉ L'HOLÍSTIC* El vertical és construït a partir dels nodes fulla del subgraf vertical; en canvi, l'holístic és bastit a partir dels nodes fulla del subgraf retrovertical.

El vertical sols té en compte l'eix Z amb XY lliure; per contra, l'holístic cerca el sentit del flux causal del model òptim a partir de maximitzar Z i Y i alhora minimitzar X (IMPORTANT: vegeu el codi comentat de l'ANNEX1). L'holístic és molt més robust que no pas el vertical.

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 càcul o d'aproximació del flux causal i comparar els resultats de localTest() amb els resultats del comportament del model geomètric. És per això que hom pot calcular:

-> o bé l'angle del flux causal vertical sols respecte de Z (un únic angle respecte l'eix Z, el pla XY és lliure) i comparar-lo amb l'angle de flux local a fi de comprovar si el local convergeix amb el flux causal vertical

-> 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 dos nodes non-adjacents que inclouen fins a 3 nodes com a condicionants al test d'independència.

#3.5 MODEL CAUSAL GEOMÈTRIC La construcció del model causal geomètric és fonamentat en una representació topològica tridimensional. En aquesta etapa, els subgrafs vertical i retrolateral es projecten en un espai topològic tridimensional, en el qual els nodes adquireixen coordenades definides per tres dimensions que emergeixen directament de la descomposició estructural del DAG. Aquestes dimensions corresponen al grau d'entrada (dependència causal), al grau de sortida (efecte causal) i al nivell jeràrquic del flux causal (relació vertical). Aquesta representació constitueix el model topològic causal, que codifica l'estructura subjacent del DAG en termes geomètrics.

A continuació, hom integra la informació empírica, les dades experimentals no són incloses directament al graf, sinó mitjançant un procés en dues fases. En una primera fase, hom n'extreu els estadístics rellevants, de les dades empíriques, amb l'objectiu de captar la informació essencial del sistema. Segonament, aquests estadístics es projecten geomètricament sobre les coordenades topològiques, mitjançant transformacions geomètriques.

El resultat d'aquest procés són les coordenades topoempíriques, que integren a dins d'un únic espai geomètric tant la informació estructural a priori derivada del DAG combinatori com la informació empírica a posteriori procedent de les dades experimentals.

## LA METÀFORA BAYESIANA

A tall de síntesi hom es pot valdre de 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-quantitatativa. 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, refinir 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.

En resum, hom fa la 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 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).

Havent fet una visió holística del mètode, hom en reescriu de forma més aprofundida les fases:

### **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:

$\$ X' = \frac{X - \bar{X}}{s_X} \\ Y' = \frac{Y - \bar{Y}}{s_Y} \\ Z' = \frac{Z - \bar{Z}}{s_Z}$  on  $s_i$  és la desviació estàndard  $\$$

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 intermigs 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 columna 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 \space coords3Dn_{diag}= matriu\space coords_3Dn · matriu\space diagonal\space dades_{diag} ##
```

El producte de cada fila de la matriu pel valor corresponent del vector de valors  $\mu/\sigma_{\text{global}}$  que és equivalent a multiplicar per una matriu diagonal. Per exemple, pel node Mek:

```
##coords3Dn = (0.519, 3.255, 1.864)\\" escala = 6.864877\\" coords3Dn_diag = (0.519 · 6.864, 3.255 · 6.864, 1.864 · 6.864) = (3.565, 22.347, 12.796)\\" Matriucoords\space topoemp1 = Matriucoords\space topo · Matriuemp\space 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).

```
## Matriucorr\space arestes = Matriucorrelacions · Matriuadjacent\\" DataFrame_{refós} = DataFrame_{corr\space arestes} + DataFrame_{coords\space 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.

## **JUSTIFICACIÓ D'ÚS DE LA CORRELACIÓ A LA SEGONA TRANSFORMACIÓ GEOMÈTRICA: l'ús de la funció localTest()**

Atès que hom pren la funció localTest() com a eina de referència per tal de comparar els resultats en el model geomètric 3D, la normalització individual es basa en la correlació. Però el més important és que entre les mesures estàndard és en especial adient per a embeddings causals geomètrics, ja que és acotada, té signe, és simètrica i és directament interpretable com una magnitud angular. D'altres mesures com l'odds ràtio o la informació mútua, hom no veu que disposin d'una interpretació geomètrica natural, i, per tant, no semblen gaire adients per tal de definir direccions de flux causal.

Si els tests de localTest() de daggity comproven amb les dades si hi ha o no independència condicional tot condicionant/blocant (si cal) per cap camí de rere-porta i, així, determinar si hi ha evidència correlativa; a partir d'aquí hom es pot demanar si hi ha una probable aresta absent entre aquests dos nodes non-adjacents.

El nucli dur de la funció localTest() té tres línies rellevants de codi: →la primera *tests <- impliedConditionalIndependencies(x)* agafa el DAG\_combinatori i diu quines independències condicionals harien d'esser certes, sols les que el DAG\_combinatori imposa si és correcte. Cosa que inclou tant les independències sense condicionament (*node1 ||| node2*) perquè *no hi ha cap camí obert de rere-porta com les independències amb condicionament (node1 ||| node2 | node3)* perquè a través del node3 node1 i node2 tenen un camí de contacte →la segona *f <- function(i).ci.test.covmat(sample.cov, sample.nobs, i, ...)* converteix les covariàncies a correlacions i fa el test a fi de veure si la correlació és (simple sense condicionament o parcial amb condicionament) és zero →la tercera *\*\*p.values <- 2 \*pnorm(abs(z), lower.tail=FALSE)\*\** en treu la p-vàlua

És clar, el model causal geomètric ha metritzat les dades registrades de l'estudi concret a dins de l'espai topològic, és a dir, les mesures empíriques de cada variable són traduïdes en termes geomètrics, cosa que implica sintetitzar les dades crues a informació estructural: hom malda perquè les correlacions de les dades siguin traduïdes en termes geomètrics i incloses al model topològic (traduir dades multidimensionals a una representació geomètrica, la qual cosa implica una metrització, reduir un espai de moltes dimensions a un espai 3D amb propietats mètriques que mantenen les relacions causals).

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 tots els jocs de dades que hi ha a l'abast que han servit per al disseny actual del pathway*.

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 anisotròpic. 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.

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.

## *JUSTIFICACIÓ D'ÚS DE LA CORRELACIÓ A LA SEGONA TRANSFORMACIÓ GEOMÈTRICA: l'ús de la funció localTest()*

Atès que hom pren la funció `localTest()` com a eina de referència per tal de comparar els resultats en el model geomètric 3D, la normalització individual es basa en la correlació. Però el més important és que entre les mesures estàndard és en especial adient per a embeddings causals geomètrics, ja que és acotada, té signe, és simètrica i és directament interpretable com una magnitud angular. D'altres mesures com l'odds ràtio o la informació mútua, hom no veu que disposin d'una interpretació geomètrica natural, i, per tant, no semblen gaire adients per tal de definir direccions de flux causal.

Si els tests de `localTest()` de `dagitty` comproven amb les dades si hi ha o no independència condicional tot condicionant/blocant (si cal) per cap camí de rere-porta i, així, determinar si hi ha evidència

correlativa; a partir d'aquí hom es pot demanar si hi ha una probable aresta absent entre aquests dos nodes non-adjacents.

El nucli dur de la funció localTest() té tres línies rellevants de codi: →la primera *tests <- impliedConditionalIndependencies(x)* agafa el DAG\_combinatori i diu quines independències condicionals harien d'esser certes, sols les que el DAG\_combinatori imposa si és correcte. Cosa que inclou tant les independències sense condicionament (*node1 ||| node2*) perquè no hi ha cap camí obert de rere-porta com les independències amb condicionament (*node1 ||| node2 | node3*) perquè a través del node3 node1 i node2 tenen un camí de contacte →la segona *f <- function(i).ci.test.covmat(samplecov, sample.nobs, i, ...)* converteix les covariàncies a correlacions i fa el test a fi de veure si la correlació és (simple sense condicionament o parcial amb condicionament) és zero →la tercera *\*\*p.values <- 2\*pnorm(abs(z), lower.tail=FALSE)\*\** en treu la p-vàlua

És clar, el model causal geomètric ha metritzat les dades registrades de l'estudi concret a dins de l'espai topològic, és a dir, les mesures empíriques de cada variable són traduïdes en termes geomètrics, cosa que implica sintetitzar les dades crues a informació estructural: hom malda perquè les correlacions de les dades siguin traduïdes en termes geomètrics i incloses al model topològic (traduir dades multidimensionals a una representació geomètrica, la qual cosa implica una metrització, reduir un espai de moltes de dimensions a un espai 3D amb propietats mètriques que mantenen les relacions causals).

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 tots els jocs de dades que hi ha a l'abast que han servit per al disseny actual del pathway*.

**3.6 MÈTODE GEOMÈTRIC: COMPARANÇA DELS DOS ESCENARIS** Havent validat estructuralment el model topoempíric, que és feta comprovant-ne el tancament topològic del model mitjançant la seva torització, hom segueix amb l'avaluació del model amb la comparació de dos escenaris: un escenari amb l'aresta Erk → Akt imposada i una altra en què aquesta aresta es manté exclosa. La comparació no és basada únicament en criteris estadístics locals, sinó en la coherència geomètrica del flux causal, l'anàlisi de cons i casquets causals de cada escenari.

### 3.6.1 CONCLUSIONS EN VALIDACIÓ TEÒRICA PER TIPUS CAUSALS MÍNIMS

El primer pas més raonable fóra convertir aquests subgrafs mínims de 3 nodes com a grafs, és a dir, com a exemples triviais, de models geomètrics senzills basats sols en 3 nodes amb un joc de dades de sols 3 columnes construïdes a priori per tal de preveure'n el comportament. Entendre si l'heurística geomètrica aquí definida és consistent amb la semàntica causal en condicions pures. Partir dels tipus causals mínims de 3 nodes com a laboratori conceptual a fi de validar la semàntica causal del model geomètric. A dins d'aquest marc pur sense soroll biològic, sols causalitat "pura" teòrica, hom pot triar exactament: direcció vertical, inclinació lateral, distància, divergència, convergència..., comprovar si la diferència d'angles agrupa els tests o bé en positius o bé en negatius.

Hi ha qualche test positiu que té diferències d'angle grans i qualche test negatiu que té diferències petites; el signe (+1/-1) sols en diu direcció (ascendent o descendent), no pas convergent o divergent en sentit causal. Si hom treballa amb un model reduït subsidiari del model global hom veu que el criteri geomètric aplicat no és alineat amb la semàntica del LocalTest. En canvi, hom veurà que segons el cas treballat si el subgraf és representat a dins del model general i el seu vector director local juga a favor del flux global (els fluxos local i causal convergeixen a dins del con causal), llavors sí que el LocalTest és positiu; i si el seu vector director es desenganxa del flux causal (no convergeixen a dins del con causal) llavors hi juga en contra i el LocalTest és negatiu.

Idealment una 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\_combinatori biològic real les cadenes de 3 nodes no són rectes; a més, a dins d'un DAG\_combinatori hom no veu la disposició entre nivells ni tampoc lateralment.

Hom assereix que 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) de forma aïllada del model causal geomètric general, hom ha emprat el joc de dades per tal d'identificar tipus causals mínims i establir un procediment de validació PROGRESSIVA del model; *CAL FER LA COMPARANÇA AMB EL FLUX CAUSAL GENERAL* o bé respecte l'eix vertical Z com a aproximació (flux causal vertical) o bé respecte dels tres eixos com a mètode robust (flux causal holístic) i fer la comparança entre els resultats dels tests i el con causal (hom introduceix aquest concepte més endavant, vegeu ANNEX2 ~ A13), cosa que implica treballar amb el plot del model geomètric 3D sencer, no pas un gràfic amb sols 3 nodes d'un tipus causal aïllat del model sencer.

### *3.6.2 ASSAIG DE VALIDACIÓ SIMULADA A ESCALA LOCAL EN CAMINS DE MÉS DE TRES NODES*

Quan hom es demana sobre el flux causal d'un camí compost de diversos tipus causals mínims entre dos nodes non-adjacents, el més important és veure si el camí (sense condicionament o amb l'ajut de condicionament per qualche node) és actiu/obert o blocat/tancat; si és obert, hi haurà un únic vector director talment com si el camí es tractés d'un cadena de n nodes, aquí hi ha flux causal respecte del camí proposat. Ara bé, si el camí proposat entre dos nodes non-adjacents és blocat (amb o sense condicionament) llavors sembla que hi hagin d'haver tants de vectors directors com tipus causals mínims; hom condensa el règim de camí blocat fent la mitjana vectorial d'aquests vectors directors locals en un únic vector direccional representatiu i després comparar-lo amb el vector director del camí actiu; hom fa la mitjana vectorial perquè no és gens probable que tots els vectors directors locals del camí blocat tinguin una mateixa direcció al casquet causal sinó que hom es demana per una mesura de coherència general d'aquests vectors directors locals.

Arribats aquí hom té un vector director provenint de l'escenari de camí actiu i una sèrie de vectors directors provinents de cada mínim tipus causal que compon el camí blocat. En lloc d'imposar d'entrada una ortogonalitat estricta (angle de 90 graus), hom empra una noció d'ortogonalitat direccional discreta definida per la dominància de components en eixos diferents de l'espai 3D metrificat; *així, dos vectors són causalment desacoblats si no incideixen a dins del con causal, la base del qual delimita un casquet de l'esfera de direccions.*

L'ortogonalitat estricta definida mitjançant el producte escalar s'ha d'entendre com un criteri teòric ideal, aplicable únicament en contextos on la metrització geomètrica és estable i el model presenta una coherència global estricta. En un espai causal mètric construït a partir de dades empíriques —inevitablement afectades per soroll, variabilitat experimental i dependències no observades— no és metodològicament prudent exigir una ortogonalitat exacta.

Fins i tot en absència d'una validació completa de la coherència global del model, l'ús d'angles i desviacions respecte de l'ortogonalitat ha de ser interpretat en termes relativs i comparatius, no com a condicions exactes. En aquest marc, l'ortogonalitat deixa de ser un requisit absolut i esdevé un indicador geomètric de compatibilitat causal, sensible a la naturalesa empírica de les dades i a l'estadi topològic del model.

El flux causal general del model geomètric pot esser calculat de dues maneres: → *FLUX CAUSAL GENERAL VERTICAL* (*triatge angular laxo*): a partir del subgraf boscoïde vertical, hom calcula la mitjana vectorial de tots els vectors directors provinents dels nodes fulla del susdit subgraf (tot corregint els nodes fulla que reben retrofluxos) amb ponderació dels mòduls. Al vertical hom hi optimitza sols l'eix Z (màxim Z; en canvi X i Y són lliures), és a dir, amb Z dominant, d'aquí que hom obté totes les fulles del vertical. Al subgraf vertical, el triatge de fluxos fulla és feta sols amb la restricció angular sobre la component Z dels vectors directors normalitzats, mentre que les components X i Y romanen lliures. Això defineix un casquet canònic a l'esfera de direccions, amb eix Z, que recull la variabilitat lateral pròpria del flux vertical. La tria dels fluxos fulla hom la basa en criteris angulars, però la construcció del flux causal i del con associat és feta a partir de vectors complets, preservat-ne el mòdul, que actua com a pes causal.

→ *FLUX CAUSAL GENERAL HOLÍSTIC* (*triatge angular estricto*): a partir del subgraf retrolateral, hom calcula la mitjana vectorial de tots els vectors directors provinents dels nodes fulla de l'esmentat subgraf amb ponderació dels mòduls. A l'holístic hom optimitza tots tres eixos (màxims Z i Y, mínim X), cosa que duu a triar els nodes fulla del subgraf retrolateral. En el cas holístic, la tria dels fluxos fulla hom la fa imposant restriccions angulars combinades sobre les components Z, Y i X dels vectors de flux, cosa que defineix una regió angular més restringida a l'espai de direccions. Aquest triatge identifica les fulles del subgraf retrolateral, amb menys de dispersió direccional; la construcció del flux causal que en segueix és feta a partir de vectors complets, preservant-ne el mòdul com a pes causal en la mitjana vectorial.

Encara que al vertical hom optimitza sols l'eix Z (els angles tenen un rang de 0 a 90 graus. 0 si el vector és paral·lel a l'eix, vertical pur, pujant o baixant; 90 si el vector és perpendicular a l'eix, horitzontal total), tant al vertical com a l'holístic hom treballa gràficament en tres dimensions: al vertical amb Z màxim i eixos X i Y lliures; a l'holístic amb Z i Y màxims i X mínim. Per tant, gràficament hom treballa amb angles de 360 graus, amb angles polars i azimutals alhora perquè són coordenades esfèriques, un vector 3D és formalitzat com: r com a mòdul del vector, theta (0–180) com n'és de vertical i phi (0–360) en quin sentit va en el pla horitzontal. Així hi ha un únic vector director causal (segons el subgraf triat per a calcular-ne el flux) a dins de l'esfera de direccions. A un vector 3D li calen aquests dos angles a fi de definir-se: ->theta: angle polar que va de 0 a 180 (inclinació respecte Z) ->phi: angle azimutal que va de 0 a 360 (orientació en el pla XY)

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.

### *3.6.3 FLUX CAUSAL GENERAL del model geomètric topoempíric general*

#### *ASSUMPCIÓ ESFERA DE DIRECCIONS*

hom assumeix que el radi de l'esfera de direccions és donada pel mòdul del vector director del flux causal global. Aquesta tria garanteix que l'esfera representa el conjunt sencer de direccions possibles, mentre que els vectors causals locals (col·lididors fulla) poden travessar-la sense restricció.

Tanmateix, per tal d'assegurar que tots els nodes i arestes del model romanen continguts a dins de l'esfera geomètrica associada al núvol de punts, hom imposa la condició següent: si el mòdul de flux xausal global és més petita que no el radi mínim calgit per contenir el núvol de punts, el radi de l'esfera és establerta com el màxim de distàncies del núvol respecte el centre geomètric (anomenat aquí el centre de masses).

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.

Caldria escatir quin és l'origen òptim en termes causals del vector director del flux causal holístic. Tot i que hom pot tenir en compte tant la bisectriu dels col·lididors fulla com la bisectriu dels nodes arrel, hom en pren en aquest TFM el centre de masses del model com a punt d'origen, no sols com a opció neutral sinó també conservadora: aquesta tria evitar moure artificialment el centre del con causal ni cap avant ni cap rere al llarg del flux.

El centre de masses sembla esdevé el punt de referència més robust a fi d'avaluar la coherència direccional global del model, atès que minimitza biaixos geomètrics induïts per la mateixa tria de l'origen.

#### *IMPLICACIONS D'UN MODEL GEOMÈTRIC AMB O SENSE ARESTA EN QÜESTIÓ*

A un col·lididor k-éssim, l'efecte geomètric d'una aresta afegida depèn tant del seu mòdul (força) com de la seva direcció angular (sentit en l'esfera de direccions). Així, fins i tot una correlació feble pot alterar en certa mesura el vector resultant de la mitjana vectorial si la seva direcció és molt diferent dels restants.

Cal reprendre el fil a la segona transformació geomètrica de la cadena d'operacions: les correlacions entre variables són associades a les arestes entre nodes (sols les arestes presents que pertanyen al pathway, no pas arestes absents), aquí hom força una aresta absent com si fos present, cosa que implica tornar a refer les coordenades topoempíriques finals.

El DAG\_combinatori no inclou les dades empíriques en qüestió, el model causal geomètric sí, d'aquí que, per tal de plantejar els mateixos tests locals, cal treballar a dins del mateix terreny de joc: cal forçar l'aresta absent a dins del model o treure'n-la (segons el cas) i calcular el con causal modificat i

comparar-ne el mòdul i angle del con causal causal holístic. Si el mòdul és més gran i l'angle més petit, l'aresta absent és probable en termes de causalitat; altrament, si el mòdul és més petit i l'angle més gran, l'aresta absent no sembla que sigui probable en el marc d'un test geomètric que força una única aresta (forçar o bé la inclusió d'una aresta absent o bé l'exclusió d'una aresta present). Ara el subgraf retrolateral ja és connex: dos nodes fulla i un collididor fulla, en total 3 fulles. Que l'aresta absent proposada per l'article d'standard gold com a present sigui just aquesta és molt interessant perquè és just la que uneix els dos components del subgraf retrolateral.

**3.7 PRODUCTE** Els productes obtinguts com a resultat del procés metodològic són, per tant, un DAG combinatori validat, una descomposició estructural causal explícita, un model causal geomètric tridimensional topoempíric, criteris geomètrics de validació causal global, un conjunt d'eines computacionals reproduïbles.

### *3.8 Respecte de la torització del model causal geomètric (vegeu ANNEX2)*

Tot i que el tancament topològic del sistema hom pot interpretar-lo mitjançant un tor funcional, tots els criteris operatius de coherència direccional i admissibilitat causal es defineixen completament a l'esfera de direccions. El tor no introduceix noves mètriques ni criteris de decisió, sinó que forneix d'un marc interpretatiu per entendre el flux causal com un fenomen orientat però no escapable. Aquest tancament no introduceix noves magnituds ni criteris geomètrics, sinó que estableix que el flux causal, tot i esser orientat, no s'esmuny del domini modelitzat; el flux causal global definit a l'esfera no s'escapa del domini del model

El flux causal té una direcció global definida, però el sistema no permet que aquesta causalitat surti fora del domini funcional; el flux es manté contingut sense generar bucles causals. El caràcter no escapable del flux causal assenyala que totes les influències causals generades dins del sistema romanen contingudes a dins del domini funcional modelitzat, sense introduir dependències causals externes ni bucles causals.

Llavors, què aporta veritablement el plegament? Sols que la causalitat no té una "sortida final" fora del model. És a dir: que estructuralment no hi hagi efectes causals fora del sistema, que no cal postular variables externes, altrament dit, que no hi ha "fuita" del flux causal. Si un DAG sense saturació i parcimoniós no admet tancament funcional, això assenyala la presència implícita de causes externes o mecanismes absents.

### *Dos exemples triviais antagònics: tancament trivial vs tancament informatiu*

==> Un DAG amb estructura estrictament arbòria no admet tancament funcional, atès que el flux causal sols divergeix i no disposa de punts de convergència que permetin definir una direcció global. Aquesta obertura no és una propietat neutra, sinó l'expressió d'una manca estructural del model. Així, el flux causal no és autosuficient i des d'un punt de vista causal implica que el model no explica per si sol els efectes que genera. És a dir, hi poden haver causes que surtin del model o efectes que no tornen a ser absorbides funcionalment, al sistema li cal una node extern causal; el model no explica per si sol els efectes que genera.

==> A l'altre extrem, un DAG saturat, qualsevol DAG saturat és plegable per construcció. Tanmateix, aquest tancament és trivial i non-informatiu, perquè resulta de l'absència de restriccions causals. El tancament sols és significatiu quan es produeix en un DAG sense saturació i parcimonios.

En conclusió, un DAG obert, de tipus arbori, posa de manifest limitacions estructurals, perquè no permet un tancament funcional global i presenta mancances en termes de coherència causal. A l'extrem oposat, un DAG saturat pot assolir un tancament formal, però ho fa de manera trivial, amagant les deficiències estructurals i sense aportar una informació causal rellevant.

En canvi, un DAG no saturat i adientment especificat pot atényer un tancament funcional non-trivial i presentar una alta vàlua causal, en tant que reflecteix una estructura mínima però suficient per capturar les dependències causals essencials del sistema. En aquest sentit, un punt de partença metodològicament sòlid no consisteix en partir d'un graf saturat i eliminar-ne arestes a posteriori, sinó a construir una estructura mínima capaç de tancar-se funcionalment cap endavant.

#### *Lligam entre la connexitat del subgraf retrolateral amb el plegament toroïdal*

La non-connexitat del subgraf retrolateral no impedeix el tancament funcional global del sistema, atès que aquest és garantit pel domini topoempíric finit i pel rol estructurador dels col·lididors fulla. Tanmateix, aquesta non-connexitat asenyala una manca de replegament funcional intern, de manera que el tancament global no és sostingut de forma homogènia al llarg de l'estructura causal. La incorporació de l'aresta Erk → Akt restableix la connexitat del subgraf retrolateral i reforça la coherència funcional del plegament, cosa que contribueix a una articulació més integrada de les dependències transversals del sistema.

## 4. Resultats

### 4.1 VALIDACIÓ MULTIESCALA DEL DAG 2D COMBINATORI PER MACHINE LEARNING O APRENENTATGE ESTADÍSTIC

#### *Random Forests (variables), SVM (validació local) i ANN/CNN (validació global)*

El pathway original és reconvertit en un DAG combinatori format per 11 nodes i 18 arestes, gràcies a la informació continguda al mateix article hom ha pogut eliminar els cicles i obtenir una estructura vàlida.

→ *RANDOM FORESTS (RF): VALIDACIÓ DE NODES FULLA* Secció sense desenvolupament

→ *SUPPORT VECTOR MACHINE (SVM): VALIDACIÓ D'ARESTES A ESCALA LOCAL* Amb l'objectiu de validar empíricament els resultats dels tests d'independència condicionada, es defineix un procediment automatitzat basat en SVM aplicable a totes les independències estructurals llistades a l'objecte res (vegeu ANNEX3). Així, tot i que el procediment és extensible a tot el conjunt de tests locals, hom analitza sols les tres independències més extremes:

(Akt // Erk | Mek, PKA); (Akt // Erk | PKA, PKC); (Akt // Erk | PIP3, PKA);

La comparació entre models SVM amb i sense la variable Akt mostra una reducció significativa de l'error de predicció d'Erk quan Akt és inclosa, amb p-valors nuls en tots els tests considerats. Aquest resultat proporciona evidència empírica consistent que la dependència entre Akt i Erk no és espúria sota els condicionaments definits pel DAG, i dona suport a la reconsideració de l'aresta absent en l'estructura causal proposada.

-> *CONVOLUTIONAL NEURAL NETWORK (CNN): VALIDACIÓ D'ARESTES A ESCALA GLOBAL* Amb un model CNN1 constret segons l'estructura del DAG i un segon model CNN2 segons el mateix DAG però saturat: no mostren gaires diferències en la coherència global del DAG; altrettament dit, les taules de model.summary() dels dos models mostren un nombre de paràmetres gairebé iguals.

-> *INTEGRACIÓ MULTIESCALA* Quan hom fa una lectura integral dels resultats obtinguts no a les tres escales sinó dues, s'adona que sols hi ha una aresta absent informativa: tant a nivell d'aresta com a nivell global hi ha una millora inclont-la al model. La incorporació de l'aresta Erk -> Akt no degrada el model global i produeix una millora específica i localitzada en la predicció d'Akt (baixa una mica la mitjana d'error quadràtica: baixa de 0.2173 a 0.2161), sense afectar la resta de nodes. Això assenyalà que l'aresta és compatible amb l'estructura causal global i no introduceix incoherències funcionals.

MODEL 1 (sense Erk → Akt)==> MSE Erk: 0.2649 | MSE Akt: 0.2173 MODEL 3 (amb Erk → Akt) ==> MSE Erk: 0.2649 | MSE Akt: 0.2161

## 4.2 VALIDACIÓ LOCAL DEL MODEL CAUSAL GEOMÈTRIC

### 4.1.1 FLUX CAUSAL LOCAL de la cadena lateral en el marc del model geomètric topoempíric general

Aquí hom rebutja la independència (de fet, Akt i Erk no són mai independents no importa quins siguin els nodes condicionants):

estimate p.value 2.5% 97.5% Akt // Erk | PKA 0.8974968 0.9641513 0.9757488

### 4.1.2 FLUX CAUSAL VERTICAL GENERAL del model geomètric topoempíric general

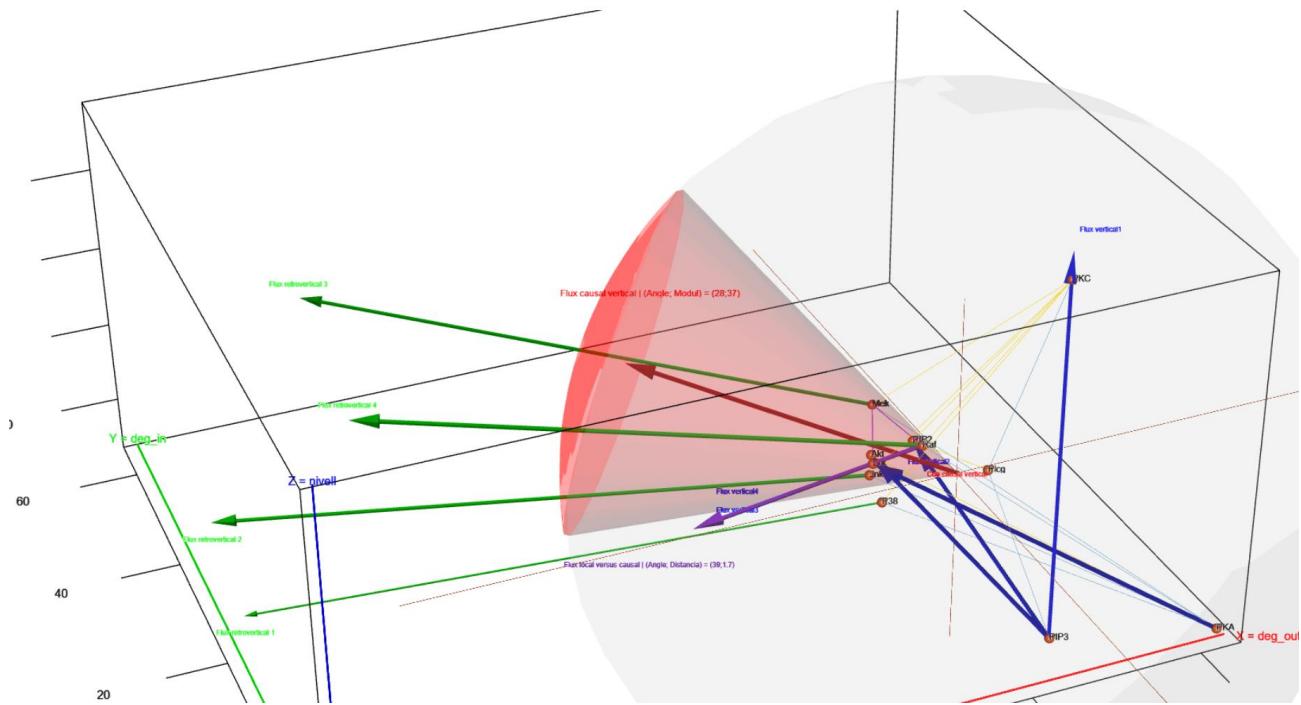
Essent fidel a l'arbre vertical sols n'hi han tres, de vectors directors verticals purs, és a dir, que són estrictament verticals sense contraflux (si hom considerés els dos col·lididors-fulla verticals, PIP3->Akt<-PKA i Plcg->PKC<-PIP2, els resultats foren molt semblants i no canviarien gaire els angles; hom n'agafa sols l'aresta de cadascuna que té la tendència a esser més significativa en el conjunt del model (recordeu la proporció de tests vàlids respecte dels fallits segons la funció de localTest()), 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)

## ASSUMPCIÓ ESFERA DE DIRECCIONS:

Hom hi afegeix la segeta del flux causal general vertical (amb l'origen del seu vector director al centre de masses). **IMPORTANT:** SENSE forçar l'aresta absent Erk->Akt, el flux local i el general vertical no sembla que convergeixin a dins de l'espai conoïdal. *Cal després assajar un nou model tot forçant l'aresta Erk->Akt, cosa que implica recalcular les coodenades topoempíriques de nou i tornar a fer el plot.*

### 4.1.3 FLUX CAUSAL VERTICAL (8 col·lididors fulla) AMB COORDENADES coords3Dn\_diag\_corr1 (SENSE ARESTA Erk->Akt)

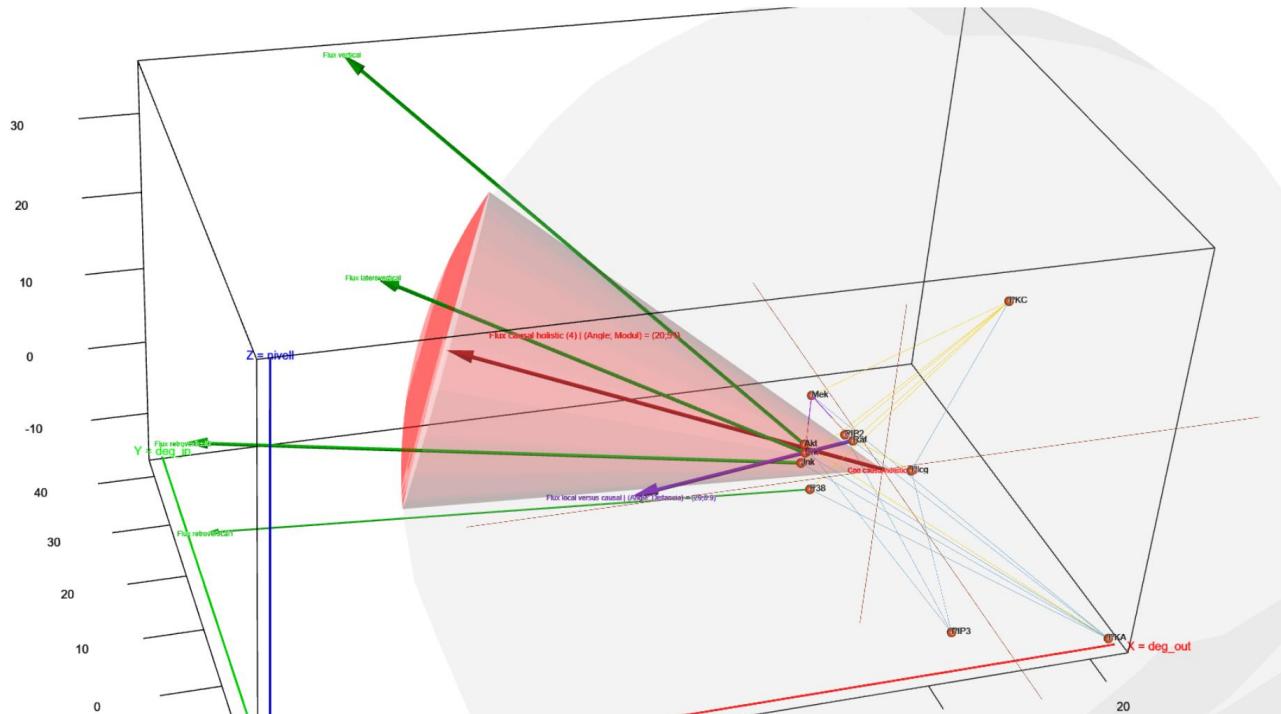
Hom pot veure com el vector director del flux local (vegeu segeta lila) travessa el con causal però surt fora de l'espai conoïdal; el seu sentit no sembla convergir a dins del casquet de direccions. SENSE forçar l'aresta absent Erk->Akt el resultat sembla, per tant, sense convergència.



### 4.1.4 FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn\_diag\_corr1 (SENSE ARESTA Erk->Akt)

Hi han 4 col·lididors fulla: 1 vertical, 1 laterovertical i 2 retroverticals. Aquesta funció permet determinar l'angle entre el flux local que és en estudi i el flux causal global. El flux local —tot i atravesar el con— cau a fora del con causal i, per tant, no sembla que sigui a dins del casquet de

l'esfera de direccions. Un altre fet a ressaltar és que l'angle del con ara és més petit: abans amb el flux causal vertical hi ha un angle de 19.58 i ara l'angle del con és de 18.37 (hom recorda que l'angle del con és definit com la mitjana angular dels col·lididors fulla); sembla que el càlcul del flux causal per mitjà del subgraf retrolateral dóna un con més òptim en termes de model: *angle mitjà més petit i mòdul més llarg*. El gràfic mostra quin flux causal hom obté a partir del subgraf retrolateral (encara hom no força l'aresta absent com a present Erk->Akt amb el recàlcul consegüent de les coordenades topoempíriques).

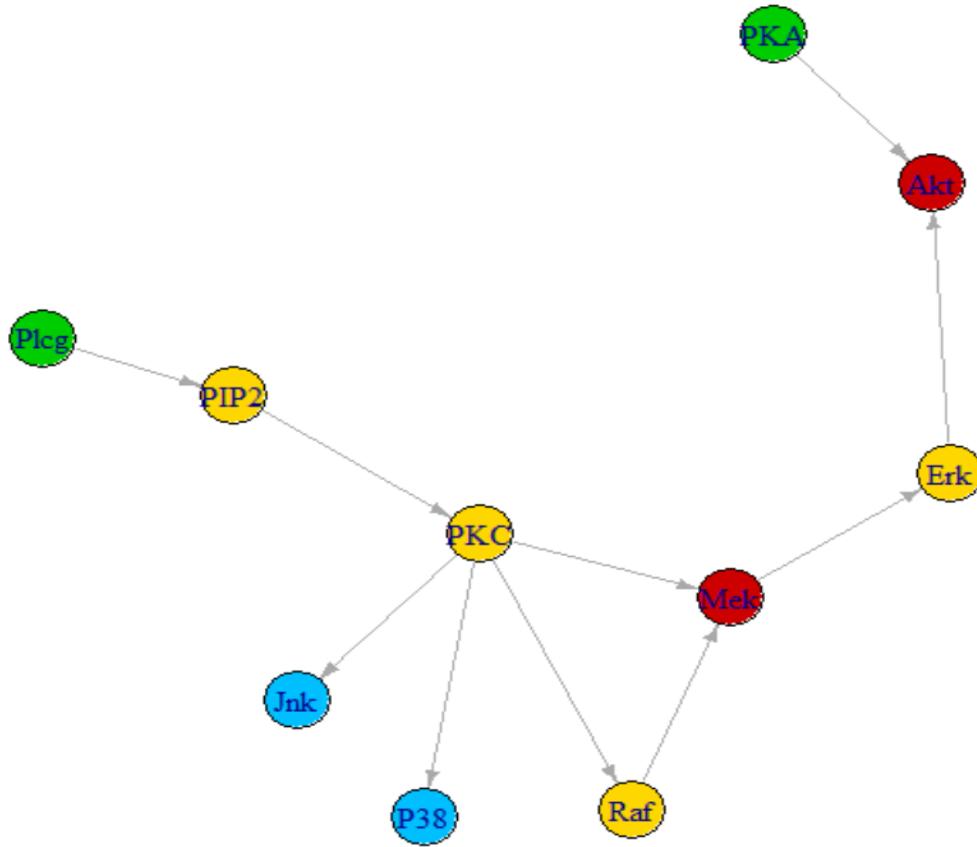


### *Hom força la inclusió de l'aresta absent Erk -> Akt com a hipòtesi estructural alternativa*

El cas d'estudi és l'aresta absent Erk -> Akt, si hom força l'aresta a dins del model causal geomètric, causa la reducció de nodes col·lididors fulla de 4 a 3.

Ara el subgraf retrolateral ja és connex: dos nodes fulla i un collididor fulla, en total 3 fulles. Que l'aresta absent proposada per l'article d'standard gold com a present sigui just aquesta és molt interessant perquè és just la que uneix els dos components del subgraf retrolateral.

### subgraf retro lateral connex dirigit

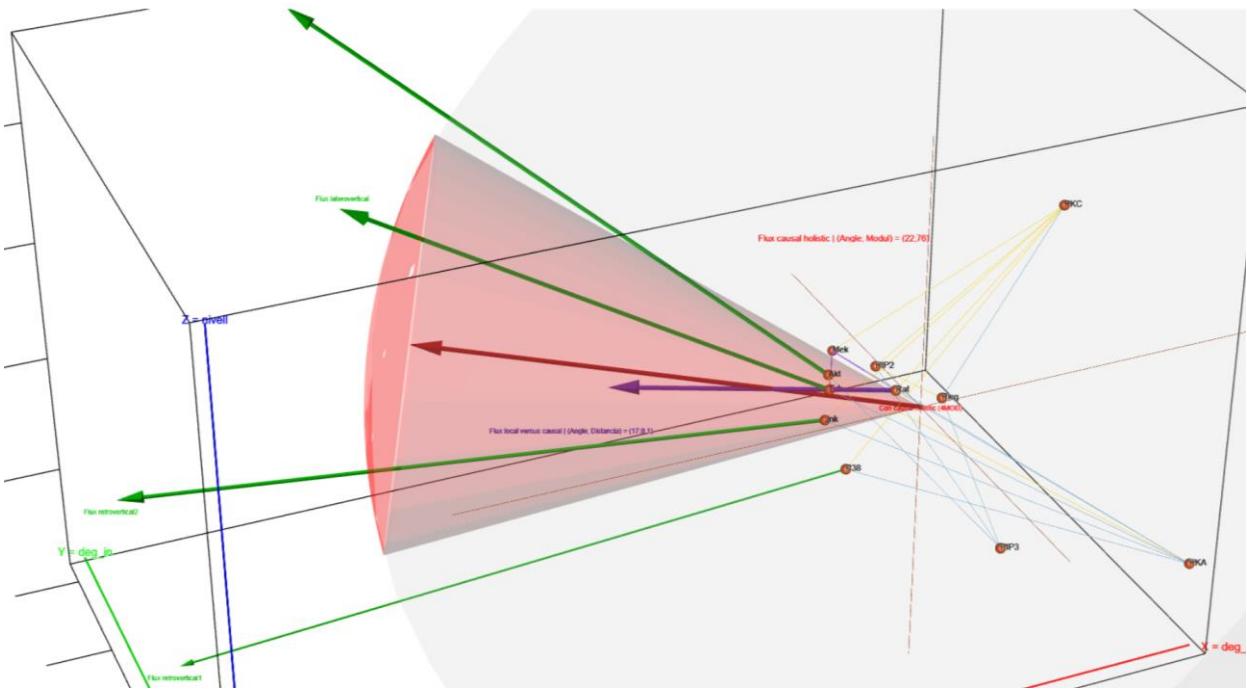


Una nova cadena (Mek -> Erk-> Akt) i un nou col·lididor (Erk -> Akt <- PKA), aquest darrer canvia del tot el paper d'Akt dins del subgraf, la qual cosa no altera l'estructura retro lateral basada en PKC però transforma Erk en medial i transmforma Akt en un nou punt de confluència causal. Tots dos canvis transformen qualitativament la geometria del flux causal holístic.

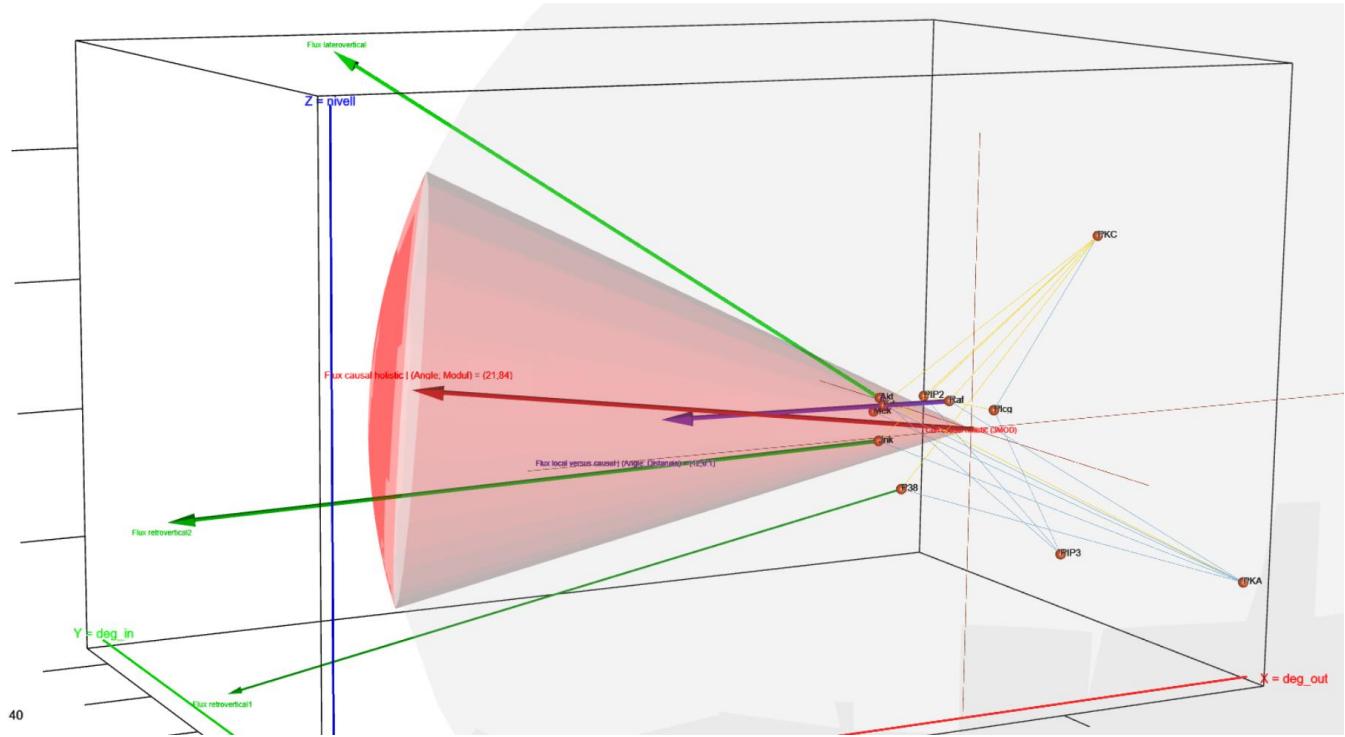
Hi han els mateixos nodes que a g\_retrolateral1 i sols una única aresta més, però el sorgiment de nous col·lididors sembla que acreix la probabilitat de transició de non-connexitat a connexitat del subgraf tal com en aquest cas en efecte s'ha esdevingut. El col·lididor és una estructura de convergència (pot reduir el nombre de fulles, cosa que tanca el con causal i reorienta el flux holístic).

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

#### *4.1.5 FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn\_diag\_corr1\_MOD (AMB ARESTA Erk->Akt)*



**4.1.6 FLUX CAUSAL RETROLATERAL (*3 col·lididors fulla*) AMB COORDENADES  
coords3Dn\_diag\_corr1\_MOD (AMB ARESTA Erk->Akt)**



## *VALIDACIÓ DE L'ARESTA ABSENT DE L'ARTICLE DE SACHS ET AL. COM A HIPÒTESI ESTRUCTURAL ALTERNATIVA*

Dins del marc 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, és a dir, que és absent al DAG 2D i caldria que fos present) és on hom centra la validació local del model. És a dir, sabent el resultat no sols de l'article de referència sinó també del test estadístic aplicat al DAG combinatori, hom analitza en quina mesura convergeixen els susdits resultats amb els del model geomètric.

L'anàlisi en què hom centra l'anàlisi és la via Akt\_||\_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 un espai conoïdal i el seu casquet esfèric associat a fi de comprovar la convergència geomètrica entre el flux causal holístic i el flux local definit pel vector director de la cadena Raf->Mek->Erk (també hom podria haver assajat amb la forca Akt<-PKA->Erk, vegeu punt 2.2 o amb d'altres fets reportats per l'esmentat article). 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 exclusiu de flux cap a una fulla final (PKC acaba essent un node medial), sols un node col·lididor-fulla vertical compartida amb PKA → PKA té un camí exclusiu de flux cap a una fulla pròpria final però retroalateralitzada 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 perquè els seus fluxos de sentit contrari xoquen als nodes col·lididors).

Concloent, els dos nodes terminals de flux vertical que semblen d'entrada operatius (actius) són un col·lididor-fulla compartit (Akt) de PIP3 i PKA i un node-fulla exclusiu (Erk) però retroalateralitzat 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 retroalateralitzat Erk és connectat a l'únic node col·lididor de flux vertical: els resultats del model geomètric proven que l'efecte dels nodes arrels convergeix a Akt. És a dir, l'angle de flux causal holístic respecte de la cadena Raf->Mek->Erk retroalateral convergeixen a dins del con causal i, per tant, comparteixen l'àrea de direccions definida pel casquet esfèric, la base del qual és el con.

## *VALIDACIÓ GLOBAL DEL MODEL CAUSAL GEOMÈTRIC: subgraf retroalateral versus torització del model*

D'una banda la descomposició del DAG combinatori ha donat dos subgrafs vàlids, és a dir, hom identifica de forma explícita explícita tant un subgraf vertical com un de retroalateral, tots dos sense connexitat però vàlids perquè l'agrupament de nodes en cada subgrafs és raonable (dos components en cada subgraf). El nombre de nodes per component a cada subgraf és diferent: al vertical són dos arbres amb un nombre de nodes equiparable (5 versus 6); però al subgraf retroalateral, els dos

components són molt descompensats (2 versus 6). Sigui com sigui no hi han molts de components de nodes aïllats, cosa que hauria estat un problema.

D'una altra hi ha tancament toroidal en ambdós escenaris, sigui quin sigui l'estat del subgraf retrolateral (connex o non-connex, amb l'aresta d'Erk->Akt o sense); tot i que cal dir que la geometria resultant difereix en termes de cohesió interna pel que fa al subgraf retrolateral: hom passa d'un subgraf retrolateral non-connex a un de connex.

El lligam entre la torització del model topològic i el subgraf retrolateral és clara: un subgraf retrolateral amb molts de components aïllats hauria compromès la torització. No ha estat el cas, al model topològic non-connex li correspon un subgraf retrolateral de dos components i un d'ells és molt petit, sols de dos nodes; en canvi, el gran agrupa tota la resta de nodes. Cosa que explica que no afecti l'estabilitat del tancament en la seva torització.

### 4.3. Objectius: anàlisi crítica dels resultats

Aquests són els quatre models: el verd amb què el flux causal vertical és basat en el subgraf vertical amb 8 col·lididors fulla sense incloure l'aresta absent, el blau amb què el flux causal holístic és basat en el subgraf retrovertical amb 4 col·lididors fulla sense incloure l'aresta absent. I els dos taronges amb què el flux causal holístic és basat en el subgraf retrovertical amb 4 i 3 col·lididors fulla tot incloent ara sí l'aresta absent, cosa que implica un recàcul de les coordenades topoempíriques. Cal afegir que el model retrolateral amb 4 col·lididors fulla amb l'aresta en qüestió no hi hauria d'esser per definició, perquè en incloure l'aresta Erk->Akt sols hi han tres nodes fulles no pas quatre, tanmateix hom ha forçat la situació per veure'n si la teoria s'accorda amb la pràctica i els resultats com així ha estat.

Hom pot veure que d'entrada el subgraf retrolateral gràcies a la inclusió al model de l'aresta Erk->Akt passa a esser connex. Després tenim la taula de resultats: els valors de les columnes milloren a mesura que hom va del primer model (1a fila: vertical 8 col·lididors fulla) fins al darrer model (3a fila: retrolateral 3 col·lididors fulla).

→2a columna: el millor flux causal global segons els resultats és l'holístic, és el més robust, tots tres són entre 20 i 22, sols amb l'angle mitjà ponderat no n'hi ha prou, cal comparar angle mitjà ponderat i mòdul alhora.

→3a columna: aquí es veu el salt qualitatiu del darrer model amb 3 col·lididors fulla amb l'areta amb el segon amb 4 col·lididors i sense l'aresta, gairebé el doble de força. Per tant, tots dos models tenen un angle gairebé igual però quan hom inclou l'aresta, el mòdul s'acreix en un 65%, té molta més de força.

→4a columna: respecte el con causal és molt clar, sense l'aresta el flux local de la cadena Raf->Mek-> cau a fora de l'espai conoïdal; en canvi, amb l'aresta a dins. Geomètricament, la convergència de fluxos és a dins de l'espai conoïdal i dessota del casquet esfèric de direccions.

→5a columna: ja comentat abans, gràcies a la inclusió de l'aresta Erk->Akt, els dos components del subgraf retrolateral s'apleguen en un únic cos. Cosa que és molt significativa: la connexitat té una forta relació amb el flux causal.

→6a columna: hom pot definir la relació geomètrica entre el flux local i el flux causal global per l'angle que hi ha entre tots dos fluxos i la distància mínima. En aquesta columna hom veu com l'angle té un valor mínim en el darrer model un altre cop de 12 graus, molt per dessota de la resta de models.

→7a columna: la distància mínima entre els mòduls dels dos fluxos suara esmentats torna a esser la que correspon al darrer model amb 0.05 unitats, molt per dessota respecte dels dos primers models que no tenen l'aresta en qüestió.

TAULA RESUM RESULTATS							
subgraf retrolateral non-connex dirigit				subgraf retrolateral connex dirigit			
MODEL	FLUX CAUSAL GLOBAL			CON CAUSAL	CONNEXITAT	FLUX LOCAL VERSUS CAUSAL	
	Subgraf d'ús per al triatge dels col·lididors fulla	Angle mitjà ponderat	Mòdul resultant	flux local respecte l'espai conoïdal	Nombre de components al subgraf	Angle interfluxos	Distància mínima

			(força del flux)				
SENSE Erk->Akt FLUX VERTICAL	VERTICAL 8 col·lididors fulla	28	37	FORA	2	39	1.664
SENSE Erk->Akt FLUX HOLÍSTIC	RETROLATERAL 4 col·lididors fulla	20	51	FORA	2	29	0.867
AMB Erk->Akt FLUX HOLÍSTIC	RETROLATERAL 4 col·lididors fulla	22	76	DINS	1	17	0.055
AMB Erk->Akt FLUX HOLÍSTIC	RETROLATERAL 3 col·lididors fulla	21	84	DINS	1	12	0.050

**Si no hi ha cap error de càlcul que comprometi els resultats, hom conclou que el mètode causal geomètric arriba als mateixos resultats que l'article de referència de Sachs et al. amb relació a la necessitat d'incloure l'aresta Erk->Akt al DAG combinatori.**

**Tots els aspectes (del flux causal global, del con causal i del casquet esfèric de direccions associat, de la connexitat del subgraf de referència sigui el vertical o el retrolateral i de la relació geomètrica entre flux local i flux global) analitzats conjuntament a la taula fan coherència en la mateixa direcció.**

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

La gran quantitat de pathways que hi han ([1]vegeu la llista dessota), 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 (molt més que no pas qualsevol

assaig clínic concret). En el marc de l'assignatura del màster de Biologia Estructural, per exemple, hi ha una gran quantitat de malalties descrites com a pathway.

[1]: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...: *en biomedicina*, Signaling pathway, signal transduction pathway (camí de transducció de senyal), Metabolic pathway (Via metabòlica), Regulatory pathway (Via reguladora), Biochemical pathway (Via bioquímica), Molecular pathway (Via molecular), Cellular pathway (Via cel·lular); *en bioinformàtica i biologia de sistemes*, Causal network / causal graph ((xarxa causal / DAG), Regulatory network (Xarxa reguladora), Interaction network (Xarxa d'interaccions), Network module (Mòdul o subxarxa funcional), Gene set / Gene module, Mechanistic model (Model mecanístic); *en farmacologia i toxicologia*, Mechanism of action (Mecanisme d'acció: és un pathway funcional del fàrmac), Mode of action, Drug pathway / response pathway, Adverse outcome pathway (Model dels toxicòlegs per a efectes adversos o DAG); *en genètica / epigenètica*, Gene regulatory pathway (Via reguladora gènica), Chromatin pathway (Via epigenètica / modificacions), Developmental pathway (Via de desenvolupament); *en immunologia*, Immune cascade (Cascada immune), Cytokine signaling pathway, Activation cascade (Cascada d'activació); *en enginyeria i control* (quan els biomèdics col·laboren amb enginyers), Control system (Sistema de control), Directed graph (Graf dirigit), Flow diagram (Diagrama de flux), Dependency structure (Estructura de dependències); *en economia / Ciències Socials*, Causal chain, Path analysis, Causal route, Impact pathway; *en física / sistemes dinàmics*, Transition pathway, Energy landscape path, Reaction pathway; *en ecologia / evolució*, Interaction chain, Trophic pathway (cadenes tròfiques); *en informàtica / IA*, Pipeline, Workflow, State transition system, DAG (Deep learning: computational graph); *en epidemiologia*....

El model geomètric pot esser una bona solució, per exemple, als DAG amb què treballen els epidemiòlegs, són DAGs molt barrocs, molt recarregats de nodes i arestes. Són models d'alta complexitat combinatòria, però això no afecta el model causal geomètric. Fins i tot hom pot treballar amb models saturats, per exemple, respecte dels nodes fulla com a punt de partença d'un estudi (un model del tot saturat també és possible però infinitament més complex i dens). Que el problema causal sigui definible com a espaciotemporal pot esser un gran avenç metodològic.

## 6. Conclusions i treballs futurs: línies de continuació de desenvolupament del mètode

L'estrucció del model causal geomètric és fixada en les fases aquí dessota detallades, però no hi ha un únic com: malgrat que al llarg del desenvolupament del mètode hom empra el verb o a dins d'un context determinat (per exemple en les definicions); hom els empra com a presa de decisió d'una opció respecte d'altres de possibles, altrement dit, *és una metodologia en desenvolupament*. Hom no imposa de forma exclusiva 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 o eines estadístiques com és el localTest() per al DAG 2D.

→A la fase de descomposició del DAG 2D, hom tria el graf arbori 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 col·lididors fulla, la seva partió (quina aresta roman al subgraf vertical i quina és remesa al retrolateral) pot fer-se segons d'altres criteris, aquí hom proposa per mitjà del localTest(), amb el benentès que hom accepti aplicar la definició d'arbre/bosc a un pathway. La ràtio entre tests coherents versus fallits amb localTest() pot esser una manera de quantificar quants de tests coherents hom obté si hom manté una aresta versus una altra del col·lididor fulla al subgraf vertical: quantitat de tests coherents versus quantitat de tests de fallits en cada opció a fi de triar-ne una per al subgraf vertical i la rebutjada, al subgraf retrolateral.

→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 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 mètode 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 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 model és a través de la mitjana vectorial global però hom pot explorar d'altres eines com fóra la regressió per tal de determinar el flux general causal.

També caldria aprofundir en la validació multiescala del DAG combinatori, així com fer una extensió de la validació multiescala del DAG combinatori amb rols clarament diferenciats segons l'escala d'anàlisi: a més a més de la validació local de les relacions causals mitjançant SVM i la validació global de

l'estructura causal mitjançant xarxes neuronals constretes per l'estructura del DAG, caldria desenvolupar la validació de la rellevància de les variables (nodes) mitjançant Random Forests amb l'objectiu d'avaluar la importància empírica dels nodes del DAG i detectar possibles redundàncies o nodes amb impacte marginal.

Amb els tres nivells treballant en bloc, caldria integrar els resultats obtinguts amb l'objectiu d'extreure conclusions sobre la validesa estructural i robustesa empírica a fi de fer el salt del DAG combinatori al model causal geomètric amb més de garanties.

Per acabar, recordar que el cas d'estudi de Sachs et al. és emprat per la comunitat científica com a banc d'assaigs (gold standard) a fi de validar nous mètodes, també aquí per al causal geomètric. Tot i que aquest mètode proposat no té cap precedent i suposa un repte dotar-lo d'eines i de claus d'interpretació, els resultats són coherents amb els resultats de l'article de referència pel que fa a la necessitat d'incloure l'aresta Erk->Akt. Dit això, atès que els resultats 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 a d'altres pathways i jocs de dades.

## 7. GLOSSARI DEL MÈTODE 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\_combinatori 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\_combinatori, 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\_combinatori 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\_combinatori ja no és sols una eina de perfeccionament i/o validació del mateix DAG\_combinatori 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 diferentes 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\_combinatori 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 combinatori: 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\_combinatori (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, deixin de sobreposar-se i ocupen el seu lloc a l'espai).

## *A3. Test d'independència condicionada 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  $-AB$ , 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.

Per exemple: si hom condiciona pel node medial d'una forca ( $A \leftarrow B \rightarrow C$ ) sota la definició que totes les arestes\_OUT és com si esdevinguessin arestes\_IN ( $A \rightarrow B \leftarrow C$ ) i aplica una pura geometria algebraica, llavors tant condicionament ( $A \leftarrow B \rightarrow C$ ) com condicionament ( $A \rightarrow B \leftarrow C$ ) 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). 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.

No és gens clar que la cadena pugui esser d'ajut per comparar la forca condicionament ( $A \leftarrow B \rightarrow C$ ) i la mateixa condicionament ( $A \rightarrow B \rightarrow C$ ), perquè hom en paga 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.

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  $A \leftarrow B \rightarrow C$ ,  $B \rightarrow A$  és molt vertical i  $B \rightarrow C$  és molt horitzontal, les dues no poden esser perfectament opositives perquè són arrelades a la geometria espacial dels nodes:  $\rightarrow$  la direcció sortint no és simètrica (el vectors directors no tenen la mateixa inclinació)  $\rightarrow$  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  $A \rightarrow B$  suposa anar en contra la verticalitat i  $B \rightarrow C$  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 GEOMETRIA ALGEBRAICA LINEAL AMB TRES NODES: geometria euclidiana**

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 (les dades ja són projectades a dins del mateix model geomètric).

~> per un cadena (vector director cadena versus vector director col·lididor): talment com si una de les arestes\_OUT revertís el seu sentit cap al mateix node condicionat; o just a l'inrevés, que una de les arestes\_IN revertís 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): talment com si una de les dues arestes\_OUT revertís 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): talment com si 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.

El flux d'un tipus causal mínim (cadena, col·lididor o forca de 3 nodes) és tallat o connectat a causa d'un node medial condicionat (quan hom diu no és que el flux canviï de sentit sinó que el flux s'obre malgrat el sentit de les arestes, hom ho diu en termes il·lustratius per tal de copsar que el flux causal passa d'una determinada manera). Ídem per al paràgraf d'aquí dessota .

#### *EN GEOMETRIA FÍSICA O CAUSAL AMB TRES NODES: geometria vectorial non-euclidiana a dins d'un espai causalment estratificat*

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 (les dades ja són projectades a dins del mateix model geomètric):

~> per un cadena (vector director cadena versus vector director col·lididor): talment com si una de les arestes\_OUT revertís 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): talment com si les dues arestes\_OUT sotmeses a test d'independència condicionada revertissin el seu sentit cap al mateix node condicionat. ~> per un col·lididor (vector director col·lididor versus vector director forca): talment com si les dues arestes\_IN sotmeses a test d'independència condicionada revertissin el seu sentit cap al node condicionat.

#### *A4. validació empírica multiescala del DAG resultant*

Hom podria validar el DAG\_combinatori, el model causal geomètric 3D 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\_combinatori 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 talment com si hom passés el grau\_out com a grau\_in, altrament 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 (força 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\_combinatori 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 treu 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 (altrament 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 anular-se o esser negatiu en termes de flux causal) i un angle lateralitzat que influeixi 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 vectors directors locals verticals, quan hi han 3 o més direccions locals o angles dispersos) i els angles entre aquests poden

anar més enllà dels 45 graus, 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 de cada aresta, on el mòdul de cada vector (la seva força en termes de flux causal local) 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.

## A12. 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\_combinatori 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 diferents... o fins i tot una cadena retrovertical. Una senzilla cadena a dins de l'espai geomètric 3D pot tenir un impacte causal molt divers, fins al punt que una cadena pot esser antagònica a una altra, per exemple, els casos de cadena vertical versus lateral. Aquestes distincions són invisibles al DAG\_combinatori en 2D, a dins del pla totes són cadenes.

Així que ara el marc geomètric imposa una lectura de cada tipus causal (cadena, col·lididor i forca) 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 l'espai geomètric 3D.

## A13. DEFINICIONS CLAU FINALS

→*Convergència* La convergència de fluxos local i causal no pot venir donada ni per l'angle sols de Z de forma aïllada, ni pel seu signe perquè dos vectors directors amb diferent signe poden tenir un angle molt petit entre ells i, en canvi, dos amb el mateix signe poden tenir sentits gairebé contraris. D'aquí que calgui una regió admissible de direccions, una regió angular tancada de direccions al voltant d'un vector director causal. La base del con causal, determinada per un radi angular, delimita un casquet únic de direccions a dins de l'esfera.

→*Esfera de direccions*: hom considera l'esfera de direccions centrada a l'origen geomètric del sistema, de radi  $R_{\text{esfera}}$ , que representa el conjunt complet de direccions possibles a dins de l'espai tridimensional.

→*CASQUET ESFÈRIC COM A SUBCONJUNT MESURABLE DE L'ESFERA*: la fracció esfèrica associada al casquet causal forneix una mesura invariant d'escala de la concentració angular del flux causal emergent (si és causalment concentrada o nul·la si el casquet depassa la semiesfera)

->*Vector causal global*: el vector causal global obtingut mitjançant la mitjana vectorial ponderada dels col·lididors fulla. Aquest vector defineix una direcció privilegiada del sistema, que actua com a eix del con causal.

->*Con causal*: hom el defineix com el conjunt de direccions que formen un angle més petit o igual que  $\theta_{con}$  amb el vector  $v_{causal}$ , on

$$\theta_{con}$$

correspon a l'angle mitjà (mitjana vectorial com a definició de flux causal emergent, perquè inclou tant intensitat com coherència direccional) dels col·lididors fulla respecte del flux causal global. La interpretació del con causal com a mesura de coherència direccional sols és significativa quan el nombre de col·lididors fulla és més gran que no un. Un angle d'obertura mitjà petit assenyala una concentració direccional alta dels vectors locals (col·lididors fulla) al voltant del vector causal holístic. Això suggereix baixa dispersió angular i, per tant, una alta consistència geomètrica interna del model en termes de direcció del flux. Aquesta conclusió és vàlida sempre que l'angle s'hagi calculat exclusivament a partir de vectors unitats (direccions) i que la tria del centre geomètric ( $p_0$ ) i del vector holístic sigui coherent amb la definició del flux causal.

En el cas límit d'un únic col·lididor, l'angle d'obertura és nul per construcció i no pot esser interpretat com a evidència de fiabilitat del model. Aquí el con causal degenera i perd capacitat informativa. En aquest escenari, la inferència no pot basar-se en dispersió angular, sinó en la comparació directa de direccions entre el flux local i el flux holístic, complementada amb criteris de mòdul.

->*Definició del casquet angular*: hom el defineix com la intersecció entre l'esfera de direccions i el con causal. Aquest conjunt constitueix un subconjunt mesurable de l'esfera, delimitat per un angle sòlid centrat en la direcció causal global.

->*Interpretació geomètrica i causal*: el casquet angular representa el conjunt de direccions compatibles amb el flux causal emergent del sistema, tenint en compte la variabilitat angular observada als col·lididors fulla. No depèn de la distribució espacial del núvol de punts, sinó exclusivament de la geometria angular induïda pel flux causal.

->*Funció del radi de l'esfera*: aquest fixa únicament l'escala geomètrica de representació i garanteix que l'esfera engloba el sistema sencer. La informació causal rellevant resideix en l'obertura/tancament angular del casquet.

## 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, Seminars

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>

## 9. Annexos

### ANNEX1 CODI R COMENTAT

IMPORTANT. Les dificultats tècniques i computacionals associades a la implementació del model causal geomètric i al càlcul de magnituds derivades (fluxos, angles, cons i casquets causals) són resoltes mitjançant un desenvolupament progressiu del codi, combinat amb una validació rigorosa dels càlculs. Aquesta validació s'ha dut a terme a través de la contrastació sistemàtica dels resultats i del suport d'eines d'assistència computacional, emprades exclusivament com a ajut al raonament i a la verificació conceptual i no pas com a font de decisió metodològica.

```
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:dplyr':
##      combine, intersect, setdiff, union

## The following objects are masked from 'package:lubridate':
##      intersect, setdiff, union

## 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:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:lubridate':
##
##     %--%, union

## 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)
library(pracma)

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/grau/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 combinatori (al model geomètric toroïdal hom contempla la possibilitat que hi hagin cicles locals sense comprometre l'aciclicitat general)

*Figura 1*

```

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
}
')

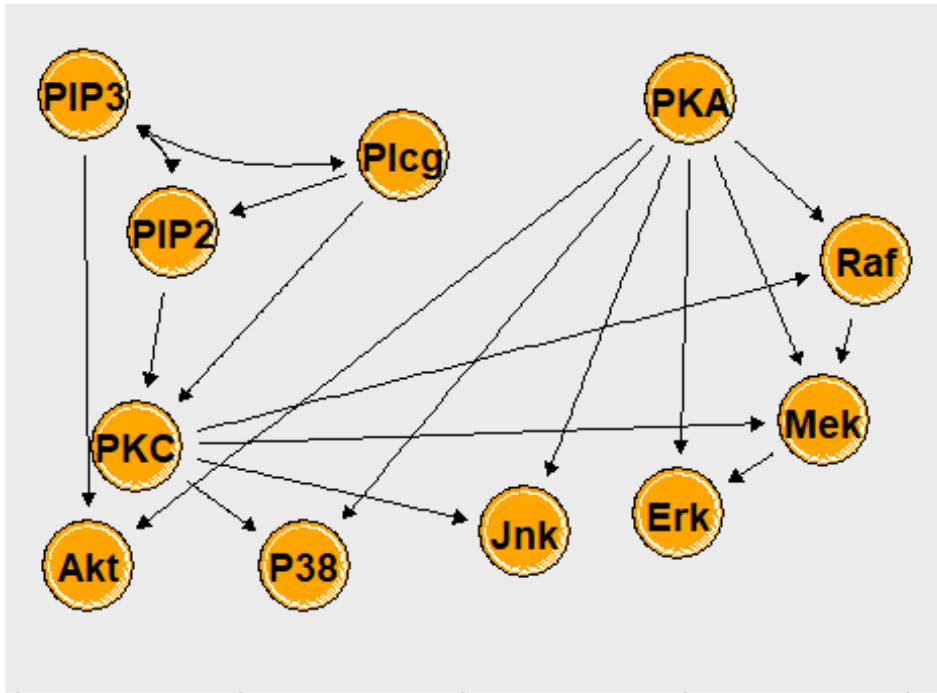
```

```

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,  $\text{PIP3} \leftrightarrow \text{PIP2}$  i  $\text{PIP2} \leftrightarrow \text{Plcg}$ , 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\_combinatori), les relacions són unidireccionals. Així que hom tria la direcció que maximitza la probabilitat condicional de l'aplec de dades; altrement 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'unidireccionalitat funcional; per exemple,  $\text{PIP3}$  és la

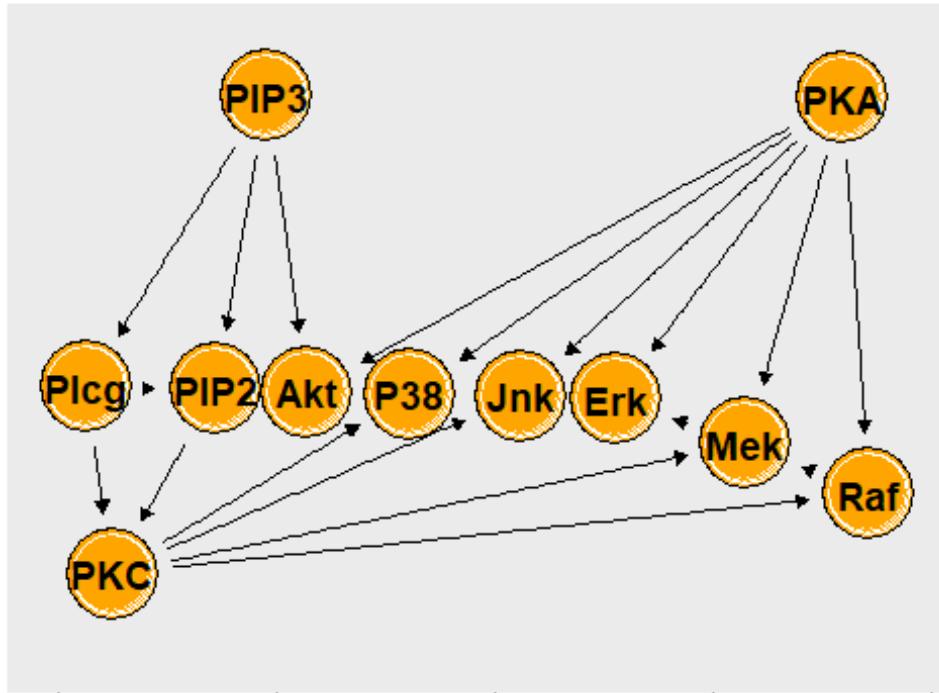
causa de canvi observat a Plcg, per tant, PIP3->Plcg; també amb PIP3->PIP2, atès que PIP3 és considerat qui genera el senyal (upstream).

Figura 2

```
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
  
```

Tots els test d'Erk-Akt rebutgen la independència condicionada sota tots els conjunts condicionants que el DAG combinatori permet, és a dir, l'absència de qualsevol independència condicionada entre Erk i Akt, fins i tot després de condicionar pels mediadors del pathway, assenyala que el DAG sense l'aresta Erk ->Akt és empíricament inconsistent amb les dades.

Taules 1 <impliedConditionalIndependencies & localTests>

```
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 04ac8e8 DN-- 11 18 --
## + attr: name (v/c)
## + edges from 04ac8e8 (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

table(E(g_bosc)$type=="lateral")

##
## FALSE   TRUE
##    14     4

table(E(g_bosc)$type=="col·lididor")

##
## 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 04b5874 DN-- 11 11 --
## + attr: name (v/c), nivell (v/n), type (e/c)
## + edges from 04b5874 (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.

Figura 4

```
#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]

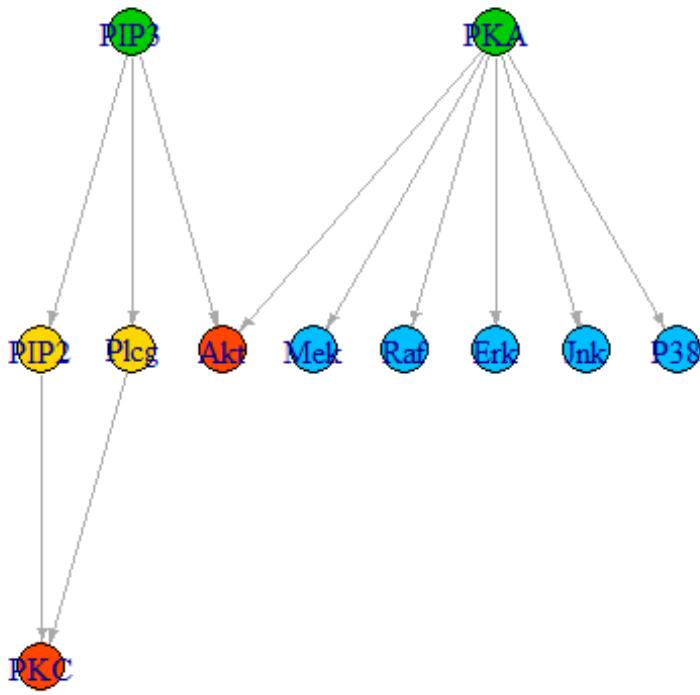
noms_nodes <- V(g_bosc_vertical)$name

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", "deepskyb
lue", "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 forca 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

*Figura 5 <subgraf bosc vertical dirigit sense col·lididors-fulla>*

```

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

arrels1 <- V(g_bosc_vertical1)$name[degree(g_bosc_vertical1, mode="in") == 0]
fulls1 <- V(g_bosc_vertical1)$name[degree(g_bosc_vertical1, mode="out") == 0]
noms_nodes1 <- V(g_bosc_vertical1)$name

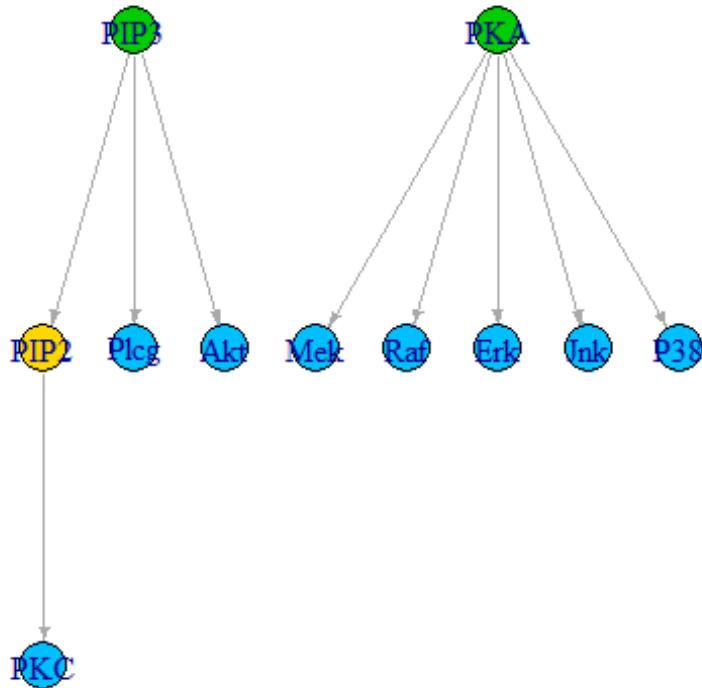
V(g_bosc_vertical1)$role <- ifelse(!is.na(match(noms_nodes1, arrels1)), "arrel",
ifelse(!is.na(match(noms_nodes1,fulls1 )), "fulla", "mitjanc
  
```

```
er"))
```

```
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 global

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

## IGRAPH 04c4231 DN-- 8 7 --
## + attr: name (v/c), nivell (v/n), type (e/c)
## + edges from 04c4231 (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 trivial (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 (vegeu l'annex codi, com a argument del plot retrolateral) al plot retrolateral calcula per nivells a fi de mantenir la direccionalitat acíclica del DAG.

Hom ja intueix que el node central del retrolateral és PKC #### Figura 6

```
#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]

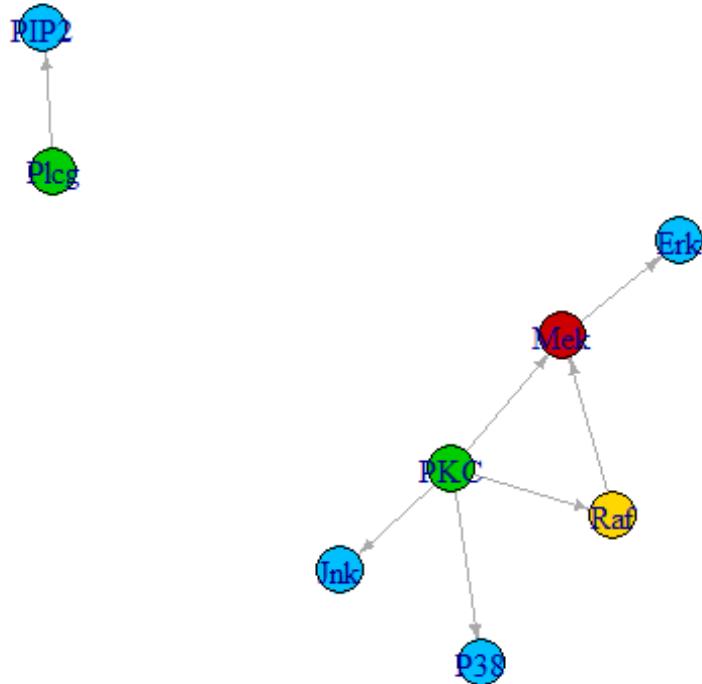
noms_nodes <- V(g_retrolateral)$name

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 04ce214 DN-- 10 9 --
## + attr: name (v/c), nivell (v/n), role (v/c), color (v/c), type (e/c)
## + edges from 04ce214 (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 confonadors, col·lididors, cadenes... segons la definició formal en teoria de grafs una cadena també pugui esser composta d'un o més nodes, però en aquest mètode 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).

*Figura 7*

```

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"
                                         )))
                                         )))

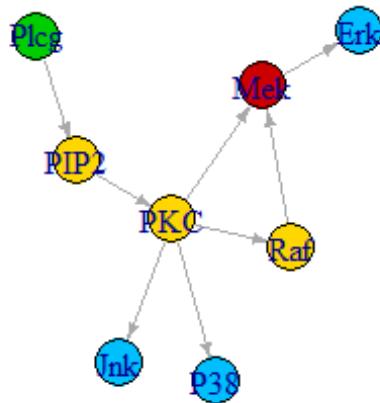
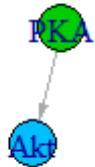
V(g_retrolateral1)$color <- ifelse(V(g_retrolateral1)$role == "col·lididor", "red3",
                                      ifelse(V(g_retrolateral1)$role == "arrel" , "green3",
                                             ifelse(V(g_retrolateral1)$role == "fulla", "deepskyblue", "gold"
             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

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 forques, cadenes i un col·lididor. És el node que més distorsiona el flux causal general i cal tenir-lo molt en compte.

Havent aïllada l'estructura vertical, hom examina el retrolateral, no sols com a pas exploratori per tal d'identificar patrons (de confusió, col·lisionadors, redundàncies o dependències laterals que podrien comprometre la inferència) sinó també en termes de causalitat (més endavant hom veu com el flux causal tant és definit a partir del subgraf vertical com a partir del subgraf retrolateral); tot fent una anàlisi separada del subgraf vertical, és a dir, la proposta d'entrada és analitzar-ho per dimensions aïllades a fi de detectar riscos causals abans d'afrontar el model complet. L'anàlisi progressiva de les implicacions de les arestes retrolaterals sobre el subgraf vertical i a l'inrevés, sense perdre de vista quines interaccions aporten informació causal i quines introduceixen soroll.

```
classifica_triada_minima <- function(sg) {
  stopifnot(igraph::is_igraph(sg))
  if (vcount(sg) != 3)
    stop("La tríada ha de tenir exactament 3 nodes")

  nodes <- V(sg)$name

  # COL·LISIONADOR (A -> B <- C)
  for (b in nodes) {
```

```

inn <- names(neighbors(sg, b, mode = "in"))
if (length(inn) >= 2) {
  return("Col·lididor")
}
}

# FORÇA (B -> A i B -> C)
for (b in nodes) {
  out <- names(neighbors(sg, b, mode = "out"))
  if (length(out) >= 2) {
    return("Força")
  }
}

# CADENA (A -> B -> C o C -> B -> A)
for (b in nodes) {
  if (length(neighbors(sg, b, mode = "in")) >= 1 &&
      length(neighbors(sg, b, mode = "out")) >= 1) {
    return("Cadena")
  }
}

return(NA_character_)
}

```

### Taula 2

```

triades_contigues <- function(g) {
  stopifnot(is_igraph(g))
  if (!is_directed(g))
    stop("El graf ha d'esser dirigit")

  combs <- combn(V(g)$name, 3, simplify = FALSE)

  purrr::map_dfr(combs, function(nodes3) {
    sg <- induced_subgraph(g, vids = nodes3)

    if (!is_connected(as_undirected(sg))) {
      return(
        tibble(
          n1 = nodes3[1],
          n2 = nodes3[2],
          n3 = nodes3[3],
          tipus = NA_character_,
          edges = NA_character_
        )
      )
    }

    ed <- igraph::as_data_frame(sg, what = "edges")
  })
}

```

```

tipus <- classifica_triada_minima(sg)
edges_txt <- if (nrow(ed) == 0)
  ""
else
  paste0(ed$from, "->", ed$to, collapse = ", ")

tibble(
  n1 = nodes3[1],
  n2 = nodes3[2],
  n3 = nodes3[3],
  tipus = tipus,
  edges = edges_txt
)
}) %>%
  filter(!is.na(tipus)) %>%
  arrange(tipus, n1, n2, n3)
}

triades_contigues(g_retrolateral1)

## # A tibble: 13 × 5
##   n1     n2     n3   tipus      edges
##   <chr> <chr> <chr> <chr>    <chr>
## 1 Mek    PIP2   PKC   Cadena  PIP2->PKC, PKC->Mek
## 2 Mek    PKC    Erk    Cadena  Mek->Erk, PKC->Mek
## 3 Mek    Raf    Erk    Cadena  Mek->Erk, Raf->Mek
## 4 PIP2   PKC    Jnk    Cadena  PIP2->PKC, PKC->Jnk
## 5 PIP2   PKC    P38   Cadena  PIP2->PKC, PKC->P38
## 6 PIP2   PKC    Plcg   Cadena  PIP2->PKC, Plcg->PIP2
## 7 PIP2   PKC    Raf    Cadena  PIP2->PKC, PKC->Raf
## 8 Mek    PKC    Raf    Col·lididor PKC->Mek, PKC->Raf, Raf->Mek
## 9 Mek    PKC    Jnk    Forca   PKC->Mek, PKC->Jnk
## 10 Mek   PKC    P38   Forca   PKC->Mek, PKC->P38
## 11 PKC   Jnk    P38   Forca   PKC->Jnk, PKC->P38
## 12 PKC   Raf    Jnk   Forca   PKC->Raf, PKC->Jnk
## 13 PKC   Raf    P38   Forca   PKC->Raf, PKC->P38

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 retrolateral 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; per tant, el retrolateral 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 subgrafs encara siguin més connectats. Ésa dir: amb `distant_sub`, hom es demana si una variació al subgraf vertical pot propagar-se al subgraf retrolateral tenint com a marc el DAG global. I si és així, en quina profunditat com una propietat estructural (independentament que aquests cadascun dels subgrafs siguin connexos o no en siguin internament). Així, amb llargària 1 hi ha relació directa, amb 2 mediació simple i amb 3 o més hi ha una estructura causal composta.

### Taula 3

```

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)$name
nodes_retrolat <- V(g_retrolateral1)$name

distant_gen <- distances(igraph_pathway_net, mode="out")
distant_sub <- distant_gen[nodes_ver, nodes_retrolat, 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$llargaria_cami),] #nombre d'ares
tes a nu camí dirigit de nodes_ver -> nodes_retrolat
(connexions_df <- connexions_df[connexions_df$llargaria_cami>0,])

##      origen destinacio llargaria_cami
## 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, segons la teoria de grafs, el pathway en els dos subgrafs, vertical i retrilateral, abans de començar el procés de conversió geomètrica, primer topològica (segons els tres eixos:

verticalitat, graus\_in, graus\_out) i després empírica amb la normalització espacial (la cadena de transformacions de les dades crues del dataset en termes geomètrics, és a dir, l'associació del quotient mitjana columna per desviació tipus global com a transformació radial de les coordenades topològiques i l'associació de la matriu de correlacions dels nodes columna com a transformació d'allargament/escurçament de cada aresta d'entre nodes de forma individual), a fi d'obtenir un model geomètric 3D amb un espai mètric coherent, on els angles representen orientacions dels fluxos causals i els mòduls representen intensitats de flux.

caldria aprofundir en l'anàlisi prèvia no sols de cada subgraf sinó també del pathway\_net com a DAG general per tal de comprovar empiricament la coherència de cadascun segons les dades. Per tant, hom podria analitzar empiricament tots dos subgrafs en termes de coherència, tant del vertical (hom sols té en compte l'eix Z per al flux causal global) a causa del flux causal vertical del qual es deriva i també el de retroflux inclòs en el subgraf retrolateral a causa del flux causal holístic (hom té en compte tots tres eixos XYZ per al flux causal global).

Hom es val d'un usatge sistemàtic dels tests d'independència condicional amb la funció localTest(), la qual assereix: , 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 funció localTest() 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 els definitius subgrafs vertical i retrolateral.

#### *4.1 Vertical amb tots dos col·lididors-fulla inclosos*

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 independentes 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\*\*\*)

*Taula 5 <Nombre de parelles localTest()>*

```
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 DAG\_combinatori, aquestes independències condicionals haurien d'esser validades empíricament sols amb les dades; aquestes relacions d'independència condicionades també són comprobables però ara directament dins del model causal geomètric 3D (creat a partir de les coordenades topoempíriques que inclouen dins del mateix model causal geomètric tant la informació del pathway a priori com les dades empíriques del cas concret a posteriori).

Taula 6 <impliedConditionalIndependencies()>

**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

```

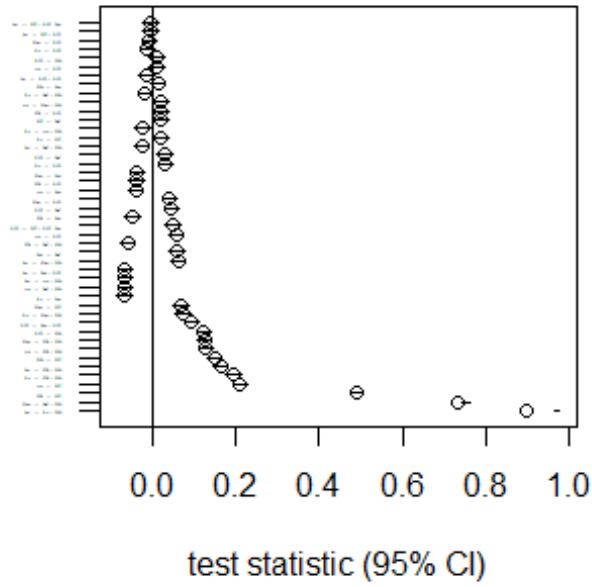
Hom pot representar també gràficament els resultats dels tests d'independència condicional #####  
Figura 8 <Plot dels tests d'independència condicional>

```

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

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))

```



Amb la llista dels resultats dels tests d'independències condicionades, ara hom veu quines 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"

```

La ràtio entre tests coherents versus fallits amb localTest() pot esser una manera de quantificar quants de tests coherents hom obté si hom manté els col·lididors-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 cap col·lididors-fulla

Si hom assagés totes les independències condicionals implícites del model sense els col·lididors-fulla — asserint la definició d'arbre/bosc— amb el conjunt de dades: en total surten 48 independències condicionals (chunk dessota)

```

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 ipgraph 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: 11 de coherents versus 37 de fallits

```

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 (0.20 vs 0.23), la diferència no és estadísticament significativa

**9/(9+36)**

```
## [1] 0.2
```

**11/(11+37)**

```

## [1] 0.2291667

prop.test(x = c(9, 11), n = c(45, 48), correct = FALSE)

##
## 2-sample test for equality of proportions without continuity correction
##
## data: c(9, 11) out of c(45, 48)
## X-squared = 0.11705, df = 1, p-value = 0.7323
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.1958875 0.1375542
## sample estimates:
## prop 1 prop 2
## 0.2000000 0.2291667

```

### 4.3 Retrolateral

Si hom assagés totes les independències condicionals implícites del model retrolateral —asserint la definició complementàriament excloent d'arbre/bosc— amb el conjunt de dades: en total surten 45 independències condicionals (chunk dessota)

```

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

```

Els resultats són: 15 de coherents versus 30 de fallits. Ara si hom fa la comparança entre les ràtios del submodel vertical sense col·lididors i el submodel retrolateral, el test de proporcions segueixen sense esser significatiu però la p-valor ara és molt més baixa i el zero —tot i esser inclòs a dins de l'interval de confiança— no és un valor central.

```

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

## res_retrolat_coherent05
## coherent   violat
##      15       30

9/(9+36)
## [1] 0.2

15/(15+30)
## [1] 0.3333333

prop.test(x = c(9, 15), n = c(45, 45), correct = FALSE)

##
## 2-sample test for equality of proportions without continuity correction
##
## data: c(9, 15) out of c(45, 45)
## X-squared = 2.0455, df = 1, p-value = 0.1527
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.31396746  0.04730079
## sample estimates:
##    prop 1    prop 2
## 0.2000000 0.3333333

```

Test per a mostres petites, sense diferència significativa

```

fisher.test(matrix(c(9,45-9, 15,45-15), nrow=2))

##
## Fisher's Exact Test for Count Data
##
## data: matrix(c(9, 45 - 9, 15, 45 - 15), nrow = 2)
## p-value = 0.233
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:

```

```

## 0.1684321 1.4339769
## sample estimates:
## odds ratio
## 0.5038841

```

## 4.4 Retrovertical (*contraflux vertical*)

Si hom se circumscriu sols a les arestes retroverticals: en total surten 6 independències condicionals (chunk dessota)

```

dag_retrovertical <- dagitty('dag {
  PKC ->Jnk
  PKC ->Mek
  PKC ->P38
  PKC ->Raf
}')
```

```

node_status(dag_retrovertical)

## # 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 que el DAG mostra) és Jnk // Mek | PKC ( $p = 0.237$ ).

```
(retro_ind_con <- localTests(x = dag_retrovertical, 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 sembla tenir lògica perquè és just el contraflux a la causalitat, (el flux de la informació de les dades empíriques va just en sentit contrari) i és just el màxim de graus\_out (arestes sortints d'un node).

```

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

En total surten 59 independències condicionals: 14 de coherents versus 45 de fallits (chunk dessota)

```
(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

```

Si hom compara la millor ràtio, el retrolateral, versus la ràtio general; el retrolateral segueix essent el millor encara que sense validació de test estadístic.

**14/(14+45)**

```
## [1] 0.2372881
```

**15/(15+30)**

```
## [1] 0.3333333
```

```

prop.test(x = c(14, 15), n = c(59, 45), correct = FALSE)

##
## 2-sample test for equality of proportions without continuity correction
##
## data: c(14, 15) out of c(59, 45)
## X-squared = 1.1711, df = 1, p-value = 0.2792
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## -0.27141293 0.07932253
## sample estimates:
## prop 1   prop 2
## 0.2372881 0.3333333

```

El resultat d'aquest test de Fisher (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 asserir 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

```

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 33% al model retrolateral versus 23% al general, tot i que la diferència no és estadísticament significativa amb una p-va=0.377 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.

## 5. MODEL CAUSAL GEOMÈTRIC 3D

```

##Grau d'entrada o graus_in(dependència), grau de sortida o graus_out (efecte) i nivell (jerarquia de flux causal): les dades empíriques sols poden deformar allò que el model topològic causal ja permet

```

La descomposició causal vertical/retrolateral basada en la teoria de grafs clàssica serveix de punt de partença per tal d'elaborar un model geomètric. Un model que 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* a un de topoempíric. El model geomètric és contruït com capes d'una ceba: →primer hi ha la conversió del pathway a DAG\_combinatori, – >després del DAG\_combinatori al model geomètric topològic gràcies a les tres dimensions vertical-graus\_in-graus\_out (és a dir, el topològic és definit com les coordenades de la causalitat en 3D: nivells o verticalitat, graus\_in o arestes entrants i graus\_out o arestes sortints) →del model topològic al model topoempíric: ara hom extreu la informació essencial de les dades crues (la base de dades) i s'inclou al model en termes geomètrics (hom associa l'estadístic mu/sigma\_global a una transformació geomètrica radial i la correlació estadística com una mesura del mòdul de cada una de les arestes presents). L'estructura topològica és preeixent a la correlació, hom deforma l'estructura causal topològica amb informació empírica, és a dir, la correlació no és causació sinó que afaiçona geomètricament una estructura causal prèvia.

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 subgrafts sinó un espai mètric on la distància entre nodes té un significat i les magnituds poden esser contínues i escalaritzables.

\$\$

Graf=(Vertexs, Arestes)=> G(V,E) \ subgraf distàncies: G\_D^{sub}=(V,E\_{vertical}) \ subgraf graus :  
 $G_G^{sub}=G(V,E)-G^{sub}(V,E_{vertical})=(V,E_{lateral}) \ E_G=E_{vertical} E_{lateral} \ .$  Hom salta de la  
unió de subgrafts \ G = G\_{vertical} V\_{lateral} \ . al producte vectorial \ = (grau\_{in}, grau\_{out}, nivell)  
=> G\_{3D}=G(R^3) \ . Finalment del vectorial al producte cartesià tancat \ (vegeu  
l'annexmodeldetorfuncional/empíricdeformable)\ (grau\_{in}, grau\_{out}, nivell) \cdot [c,d] \cdot [e,f] =>  
(S^{1 \times 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

*Taula 7*

```
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

```

*Figura 6*

```

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])

```

```

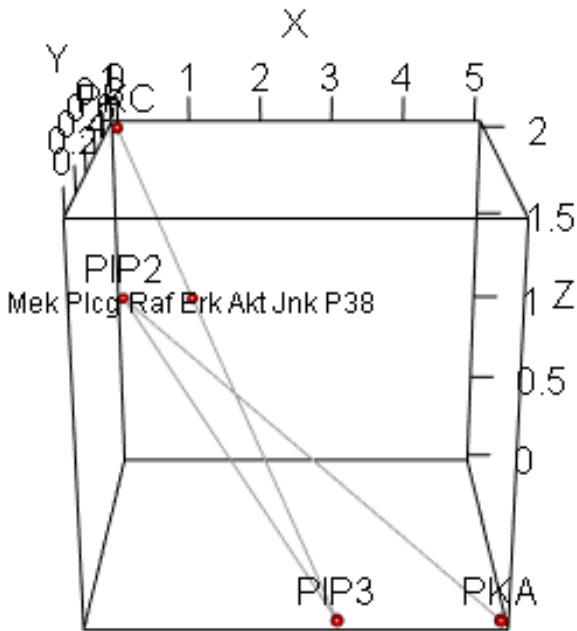
v2 <- which(g_bosc_vertical1$name == edges[i,2])
segments3d(rbind(coords[v1,], coords[v2,]), col = "grey60", lwd = 0.9)
}

text3d(coords[2:5,], texts = 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 = 0.75,
       depth_test="always")

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

Taula 3

```

#AQUÍ HOM CENTRA I TIPIFICA ELS EIXOS TOPOLOGICS
coodrsn <- scale(coords)
#coodrsn <- apply(coodrsn,2,function(col)col/max(abs(col)))
(coodrsn <- apply(coodrsn,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 horitzontal del centre de masses, cosa que afavoreix el flux causal vertical

#### Figura 9

```

open3d()

## wgl
## 4

plot3d(coordsn, 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(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 = 0.75,
       depth_test="always")

bbox <- c(
  min(coordsn[, "X"]), max(coordsn[, "X"]),
  min(coordsn[, "Y"]), max(coordsn[, "Y"]),
  min(coordsn[, "Z"]), max(coordsn[, "Z"]))

```

```

min(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)

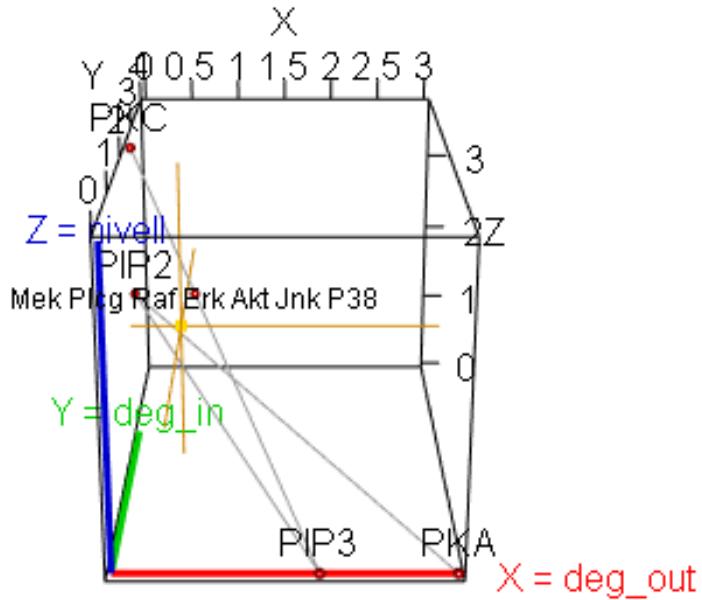
#segments3d(rbind(abs(centre - c(0, dy, 0)), centre + c(0, dy, 0)), col="orange3",
#Lwd=0.9)
#segments3d(rbind(abs(centre - c(0, 0, dz)), centre + c(0, 0, dz)), col="orange3",
#Lwd=0.9)

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  $V(g\_bosc)\$nivell$  #### Taula 8

```
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 assenyala les dues arestes espúries verticals (pseudoverticals) com a blaves (són les arestes rebutjades com a verticals, sorgides de la descomposició dels dos col·lididors fulla; és a dir, els dos col·lididors fulla tenen graus\_in=2, dues arestes entrants, una ha estat valorada com a vertical i mantinguda al subgraf vertical i la segona és inclosa al retrolateral com a pseudovertical).

#### Figura 10

```

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)

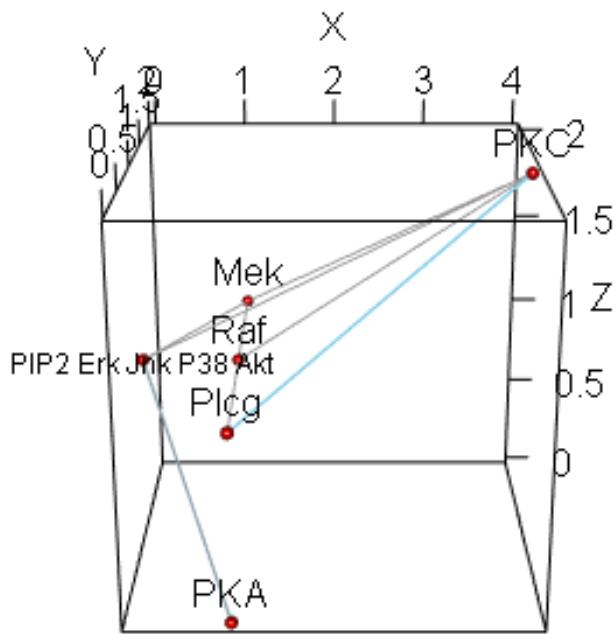
aspect3d(1,1,1)

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 = 0.75, depth_text="always")

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 #### Taula 9

```

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; caldrà tenir en compte tant les arestes pseudoverticals com les arestes verticals que tenen retroflux (les arestes que van en sentit contrari al flux causal vertical) a fi de definir el primer dels mètodes (el vertical, basat sols en l'eix Z) per trobar el flux causal del model.

#### #### Figura 11

```
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=0.75, depth_text="always")

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)

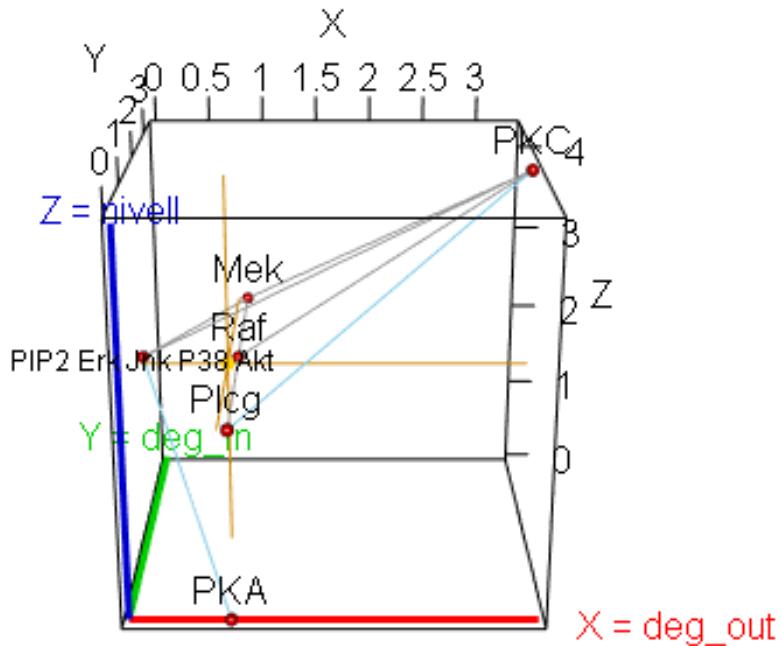
#segments3d(rbind(abs(centre - c(0, dy, 0)), centre + c(0, dy, 0)), col="orange3",
#Lwd=0.9)
#segments3d(rbind(abs(centre - c(0, 0, dz)), centre + c(0, 0, dz)), col="orange3",
#Lwd=0.9)

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), però abans de fer inferència geomètrica cal entendre que el DAG\_combinatori és subsidiari de les dades crues, el DAG\_combinatori no és un model en si mateix sinó una estructura dependent del joc de dades, no és autoreferencial, li cal la validació externa o arbitratge constant del joc de dades per mitjà d'eines estadístiques. En canvi, el model geomètric causal fusiona tant la informació general del pathway a priori així com les dades empíriques d'un assaig concret a posteriori. Per tant, els corresponents resultats dels tests de localTest() aplicat a un DAG\_combinatori són amb càlculs geomètrics del mateix model 3D.

Ara tot el model topològic (verticalitat i retrolateralitat) ##### Taula 10 <Coordenades topològiques (retrolateral i vertical) sense escalatge>

```
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. Caldria visualitzar els nodes coincidents a l'espai sense sobreposició (de fet la sobreposició de nodes segons les coordenades topològiques es resol amb la normalització empírica espacial). Altrament dit, les matrius geomètriques que reconfiguren les coordenades topològiques en coordenades topoempíriques, ho fan gràcies a la cadena de transformacions geomètriques, a partir de l'extracció de la informació rellevant de les dades crues empíriques per mitjà d'estadístics.

Sense normalitzar l'escala dels tres eixos, 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); en conseqüència, el flux causal en 3D no reflecteix relacions causals reals sinó disparitats d'escala.

##### Figura 12 <Plot topològic (retrolateral i vertical) sense escalatge>

```
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 coherentament segons noms
edges <- as_edgelist(g_bosc, names = TRUE)
edges <- apply(edges, 2, as.character)

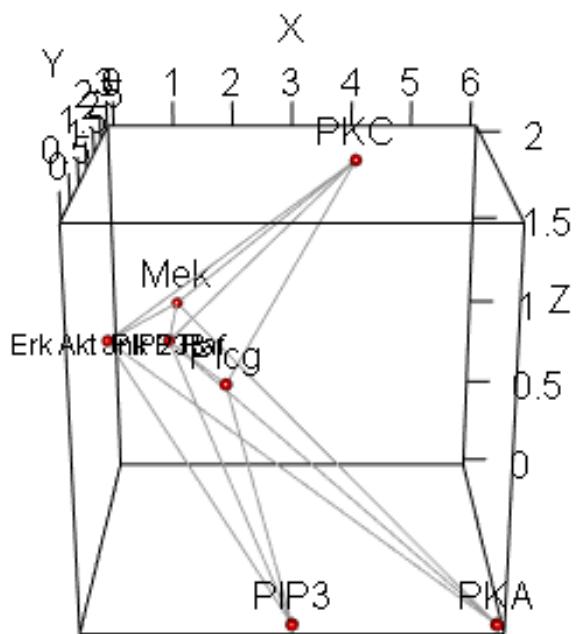
for (i in seq_len(nrow(edges))) {
  p1 <- as.numeric(coords3D[edges[i, 1], ])
  p2 <- as.numeric(coords3D[edges[i, 2], ])
  segments3d(rbind(p1, p2), col = "grey60", lwd = 0.9)
}

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=0.75, depth_test="always")
text3d(coords3D[c(2,7),], texts = c("PIP2 Raf"), cex=0.75, depth_test="always")

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

Taula 11

```
#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,] 0.01000 0.010000 0.010000
## [2,] 3.06608 3.255291 3.718099

```

Ara ja amb eixos normalitzats

#### Figura 13

```

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=0.75, depth_test="always")
text3d(coords3Dn[c(2,7),], texts = c("PIP2 Raf"), cex=0.75, depth_test="always")

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)

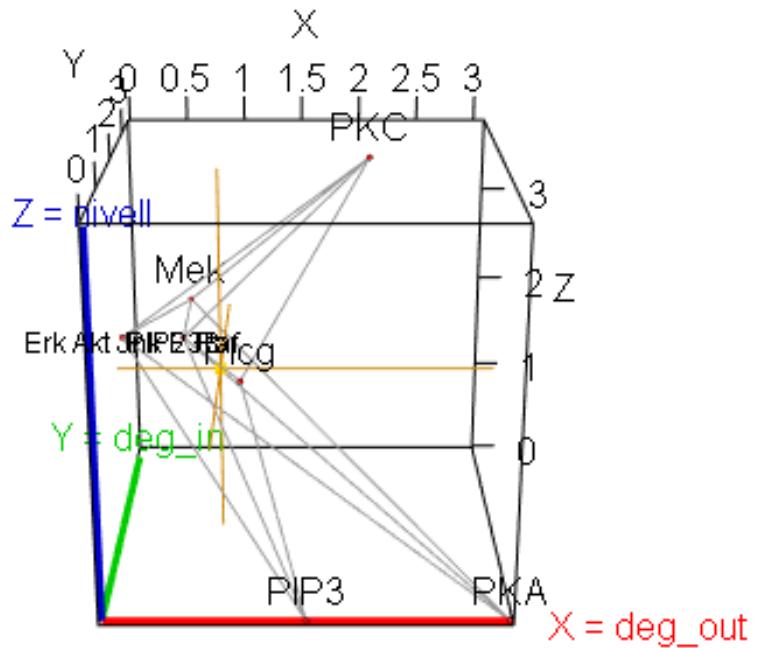
#segments3d(rbind(abs(centre - c(0, dy, 0)), centre + c(0, dy, 0)), col="orange3",
#Lwd=0.9)
#segments3d(rbind(abs(centre - c(0, 0, dz)), centre + c(0, 0, dz)), col="orange3",
#Lwd=0.9)

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 APLICACIÓ I TRANSFORMACIÓ GEOMÈTRICA (MODEL FINAL TOPOEMPÍRIC) ## PATHWAY ==> DAG(combinatori) ==> ESTRUCTURA CAUSAL(lectura topològica) ==>  $R^3$  (modelització) ## Salt de l'estructura causal del DAG\_combinatori a  $R^3$ : traduir els trets funcionals del DAG\_combinatori derivat del pathway a una representació geomètrica 3D (+temps)**

El pathway és una representació descriptiva mecànica, diu què interactua amb què segons el coneixement biològic; accepta cicles, retroalimentacions, paral·lelismes i redundàncies... no és pensat per fer inferència formal directa, respon a la manera com rutlla el sistema segons el coneixement actual. Les arestes dirigides són polisèmiques, no tenen un significat únic.

En canvi el DAG combinatori no descriu mecanismes, no li cal reproduir fidelment la dinàmica biomèdica completa sinó esser un objecte formal adient per fer inferència causal. El DAG (combinatori) no és ni geomètric ni topològic per natura, és una estructura combinatòria derivada del pathway. També és important assenyalar que un DAG inclou absències, si no hi ha una aresta, això és una hipòtesi forta: no existeix cap efecte causal directe. El DAG és una estructura combinatòria que codifica relacions causals qualitatives independentament de qualsevol representació geomètrica o mètrica; segons Pearl <the causal graph [DAG] represents qualitative causal assumptions>. A un DAG, les arestes dirigides són úniques, per exemple, en (A->B) A sols és una causa directa potencial de B.

En conclusió: el Pathway és una mena de coneixement mecànic, descriptiu per natura, és fet per tal d'entendre processos com a finalitat i el lligam amb les dades és il·lustrativa; en canvi, el DAG és una

mena de coneixement causal, axiomàtic per natura, és fet per tal d'inferir efectes com a finalitat i el lligam amb les dades és operativa.

El salt del coneixement maquinal brut (pathway) a relacions causals mínimes i coherentes (DAG combinatori) —aquí és fàcil perquè el mateix article de Sachs et al. ja reportava el DAG combinatori— és fet imposant l'única restricció formal indispensable: l'absència de cicles causals a fi que el gràfic sigui d'ús per a fer inferència causal; el DAG imposa un ordre causal. Hom no ha corregit ni redundàncies, ni paral·lelismes, ni graus\_in, ni graus\_out... perquè —tot i que el DAG combinatori tendeix a esser abstracte i qualitatius— no és incompatible d'entrada amb un DAG i pot aportar informació causal. Tot i que l'única restricció explícita és l'aciclicitat, el model assumeix que: -> hi ha un ordre causal dominant, en què no manquen causes comunes rellevants (la suficiència causal de Pearl) -> la intervenció en un node no canvia la forma de la resta de relacions, sols els valors -> la causalitat es transmet per les arestes presents -> la topologia del pathway conté la informació causal essencial Totes aquestes assumptions són controlades en certa mesura per mitjà de comparances d'escenaris, del con causal (hom en parla més endavant) i de l'anàlisi de col·lididors i camins actius.

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. L'espai empíric és formalment  $R^{11}$  (les dades crues tenen 11 dimensions) però causalment no pas perquè aquestes dimensions no són independents, són del tot estructurades per un DAG\_combinatori causal, cosa que explana que una gran part de la informació és redundant, condicionada o col·lapsable. Per tant, l'espai causal efectiu no és  $R^{11}$  encara que les dades crues en siguin. Quan hom basteix l'objecte topològic (DAG\_combinatori derivat al seu torn del pathway), aquest ja no viu a  $R^{11}$  sinó amb relació a relacions, dependències, jerarquies, col·lididors... aquest objecte ja no és un espai vectorial.

Tanmateix el model geomètric introduceix una estructura que redueix les n dimencions a 3 a fi d'atènyer una geometria 3D interpretable, 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\_combinatori derivat del pathway).

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).

La projecció justament fa que cada node adquireixi coordenades en 3D: ara cada node del DAG geomètric representa una variable observada (per exemple, una proteïna, un gen, ...) a dins d'un espai causal geomètricament significatiu, interpretable i cada aresta una relació de dependència. Cada punt reflecteix el grau d'entrada (dependència causal), el grau de sortida (efecte casual), la seva jerarquia/centralitat, les relacions retrolateroverticals.

En resum, havent feta la immersió, cada node del DAG queda associat a un punt concret a dins de l'espai tridimensional (X, Y, Z). Aquestes coordenades no provenen de dades empíriques crues, sinó de

les propietats causals derivades de l'estructura del DAG: grau d'entrada, grau de sortida, profunditat jeràrquica i contribucions retrolateroverticals. Així, la lectura topològica crea l'espai geomètric; i, a dins d'aquest espai, cada node ocupa la seva posició natural segons el seu rol causal. El model no redueix directament l'espai empíric de  $R^{11}$  a  $R^3$  sinó que primer infereix una estructura causal directa (DAG\_combinatori) i tot seguit en construeix una representació tridimensional basada en propietats causals.

## CADENA DE TRANSFORMACIONS GEOMÈTRIQUES A PARTIR DE LES DADES

Hom afaiçona les coordenades topològiques (les que provenen del DAG\_combinatori que prové d'un pathway sense cicles) segons les dades empíriques, és a dir, hom reubica els nodes a partir d'estadístics que resumeixen la informació continguda en les dades: d'aquí que hom aprofita 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, a 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 intermigs hi ha molta més de 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 mesurades en l'espai d'immersió en 3D siguin commensurables, 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) 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, mòduls, fluxos... Cal traduir les mesures empíriques de cada variable en termes geomètrics, cosa que implica sintetitzar les dades crues a informació estructural rellevant. Per exemple, és un procediment semblant al que fan els físics: traduir dades multidimensionals a una representació geomètrica 3D (+temps), cosa que implica també una metrització.

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 automatitzar 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'enriquir les coordenades topològiques a partir de la cadena d'estadístics. Així, el quotient de la mitjana per la desviació tipus global recol·loca la posició geomètrica topològica dels nodes; i la correlació de les columnes variables a la llargària (allargament/escurçament) de les arestes entre nodes 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 definitives coordenades anomenades topoempíriques un cop feta la segona transformació geomètrica)

$$\text{Mcorr\_arestes} = \text{Mcorrelacions} \cdot \text{Madjacent DFrefós} = \text{DFcorr\_arestes} + \text{DFcoords\_topoemp1}$$

La correlació entre variables, un ajustament 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 de manera equidistant entre els dos nodes). Totes dues transformacions geomètriques defineixen la variabilitat local geomètrica topoempírica 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 respecte del centre de masses i els nodes guanyen pes i visibilitat.

## ***JUSTIFICACIÓ D'ÚS DE LA CORRELACIÓ A LA SEGONA TRANSFORMACIÓ GEOMÈTRICA: l'ús de la funció localTest()***

Atès que hom pren la funció localTest() com a eina de referència per tal de comparar els resultats en el model geomètric 3D, la normalització individual es basa en la correlació. Però el més important és que entre les mesures estàndard és en especial adient per a embeddings causals geomètrics, ja que és acotada, té signe, és simètrica i és directament interpretable com una magnitud angular. D'altres mesures com l'odds ràtio o la informació mútua, hom no veu que disposin d'una interpretació geomètrica natural, i, per tant, no semblen gaire adients per tal de definir direccions de flux causal.

Si els tests de localTest() de daggity comproven amb les dades si hi ha o no independència condicional tot condicionant/blocant (si cal) per cap camí de rere-porta i, així, determinar si hi ha evidència correlativa; a partir d'aquí hom es pot demanar si hi ha una probable aresta absent entre aquests dos nodes non-adjacents.

El nucli dur de la funció localTest() té tres línies rellevants de codi: →la primera *tests* <- *impliedConditionalIndependencies(x)* agafa el DAG\_combinatori i diu quines independències condicionals harien d'esser certes, sols les que el DAG\_combinatori imposa si és correcte. Cosa que inclou tant les independències sense condicionament (*node1 ||| node2*) perquè no hi ha cap camí obert de rere-porta com les independències amb condicionament (*node1 ||| node2 | node3*) perquè a través del *node3* *node1* i *node2* tenen un camí de contacte →la segona *f <- function(i)*

`.ci.test.covmat(sample.cov, sample.nobs, i, ...)` converteix les covariàncies a correlacions i fa el test a fi de veure si la correlació és (simple sense condicionament o parcial amb condicionament) és zero → la tercera `**p.values <- 2*pnorm(abs(z), lower.tail=FALSE)**` en treu la p-vàlua

És clar, el model causal geomètric ha metritzat les dades registrades de l'estudi concret a dins de l'espai topològic, és a dir, les mesures empíriques de cada variable són traduïdes en termes geomètrics, cosa que implica sintetitzar les dades crues a informació estructural: hom malda perquè les correlacions de les dades siguin traduïdes en termes geomètrics i incloses al model topològic (traduir dades multidimensionals a una representació geomètrica, la qual cosa implica una metrització, reduir un espai de moltes de dimensions a un espai 3D amb propietats mètriques que mantenen les relacions causals).

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 tots els jocs de dades que hi ha a l'abast que han servit per al disseny actual del pathway.*

### *Normalització vectorial: no hi és inclosa*

transforma cada vector en un vector unitari, cosa que no sembla gens coherent, 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, és en essència direccional, dóna orientació.

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

la informació topològica juga el mateix paper que una prèvia (a priori) en inferència bayesiana, és a dir, el coneixement causal previ, allò que sabem del sistema abans de registrar les dades específiques de l'estudi actual que és objecte d'anàlisi; és empíric (prové d'assaigs previs o de la literatura biomèdica), però no és pas numèric (no té magnituds o intensitats assignades, llevat que el pathway s'hagi bastit a partir d'altres assaigs clínics cadascun dels quals tingui el seu joc de dades empíriques), el pathway és una estructura empírica d'entrada non-quantitativa. Però no és una a priori trivial sinó molt robusta.

Les dades de l'estudi en qüestió 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 topoempíric, 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 DAG\_combinatori 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 d’entrada 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 d’entrada 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 (angles, 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) i també als eixos retrolaterals YX per la connectivitat topològica d’un node (graus\_in i graus\_out), perquè és traduïda a dins d’un espai angular local definit pels vectors de flux incident i emergent; aquesta estructura angular primària determina la divergència o convergència del flux causal general (sigui sols vertical Z o holístic XYZ) 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’una cadena d’estadístics que n’extreuen la informació estructural rellevant, del joc de dades de l’assaig de Sachs et al., amb escalatge previ. Així, hom passa del model geomètric topològic (del DAG\_combinatori derivat del pathway) al model topoempíric, és a dir, l’estructura a priori (DAG\_combinatori) 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.

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 com a estat de la qüestió de la literatura en un tema determinat) sinó una estructura deformada pel flux causal observat: ara la distància entre nodes reflecteix diferències reals d’influència causal; angles i longituds tridimensionals empíricament significatius, comparables amb els resultats de localTest().

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 inferit segons les dades topològiques inferides a priori del DAG combinatori (al seu torn aquest darrer pervingut del pathway) i les transformacions geomètriques induïdes pels estadístics a posteriori. Altrament dit, *hom converteix la xarxa bioquímica en un volum o superfície causals vius, deformable segons la dinàmica observada*. Hom pot imaginar —per comparança— l’estructura geomètrica causal com una entitat tridimensional “molecular”: analitzar quins canvis topoempírics (en arestes in/out) corresponen a canvis físics (plegament, interacció...) al llarg del temps, on cada unitat temporal correspon a un joc de

dades actualitzat. A la fi, aquesta representació permet detectar punts de transició on la geometria causal es trenca: indicadors de canvi funcional, mutacions, inhibicions...

Normalització espacial per metrització radial: no cal centrar ni tipificar perquè tots els nodes es mouen dins d'un rang molt homogeni (entre 0.7 i 4.5). Un bon estadístic (a fi de captar el màxim d'informació empírica de les dades registrades) pot esser el quocient de la mitjana de cada columna per la desviació típica global, vindria a esser la intensitat radial mitjana ponderada de cada node. ##### Taula 12

```
# 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)

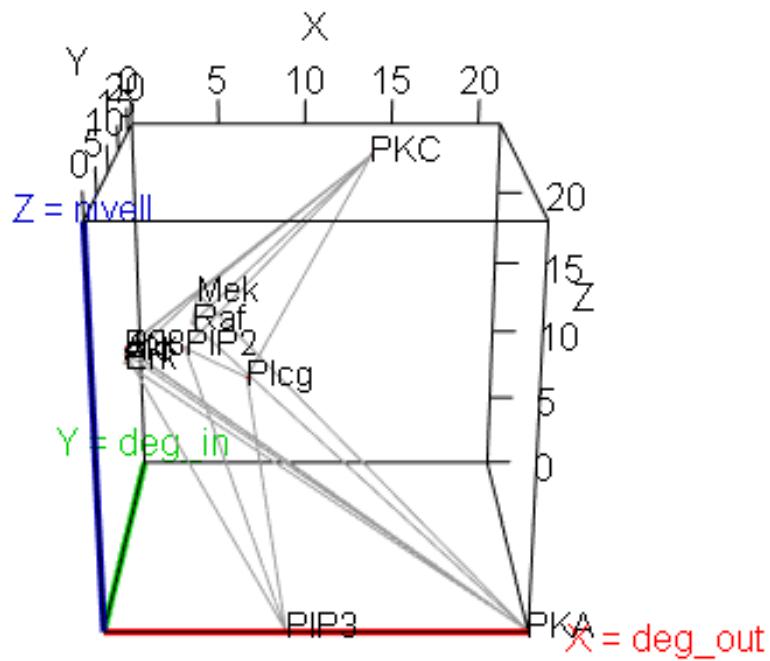
##          Raf          Mek          Plcg          PIP2          PIP3          Erk          Akt          PKA
## [1,] 1.711091 1.539091 1.236145 1.325193 1.395194 1.092741 1.120213 0.7119922
## [2,] 4.500000 4.500000 4.500000 4.500000 4.500000 4.500000 4.500000 4.5000000
##          PKC          P38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

Figura 14

```
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 índexs  
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=0.95, depth_test="always")  
  
  
  
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
```



#### 5.4.2 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). ##### Taula 13

```
# 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 Longitudo
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 DIRECCIÓ 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),])

```

Taula 14

```

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

```

Taula 15 <coords3Dn\_diag\_corr1: reajust de la llargària de cada aresta segons la correlació>

```

# 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]

  # indexes 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 actual (de <des> cap a <fins>)
  v <- p2 - p1
  len_v <- sqrt(sum(v^2)) # Llargada actual de l'aresta
  v_unit <- v / len_v # direcció unitària

  # Factor d'ajust (= 1 + corr )
  f <- df_fos$prop[i] # prop = llarg_nova / llarg_vella

  # increment de llargada que hom vol afegir o treure (repartit entre els 2 nodes)
  delta_L <- (f - 1) * len_v / 2

  # desplaçament geomètric al llarg de la direcció de l'aresta
  delta <- v_unit * delta_L

  # a partir de les POSICIONS ACTUALS (p1, p2), movem simètricament
  coords3Dn_diag[v1, ] <- p1 - delta
  coords3Dn_diag[v2, ] <- p2 + delta
}

normalitza_coords <- function(M){
  center <- colMeans(M)
  sweep(M, 2, center, "-")
}

(coords3Dn_diag_corr1 <- normalitza_coords(coords3Dn_diag))

##          X          Y          Z
## Mek   -2.940711  14.534541  0.9888282
## PIP2  -3.091212   4.352000  2.2741977
## PIP3   4.300551 -13.842702 -13.4126383
## PKA   22.826598 -16.797465 -15.4390495
## PKC   13.921936   2.493688  18.9533110
## Plcg  1.414107  -4.145035  2.4496031
## Raf   -3.142287   2.018696  3.0235362

```

```

## Erk   -8.514236  2.777365  1.6767666
## Akt   -8.242826  3.772759  2.1148075
## Jnk   -8.646380  3.211742  0.0101337
## P38   -7.885538  1.624411  -2.6394962

```

*Figura 15*

```

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=0.95, depth_text="always")

centre <- colMeans(coords3Dn_diag_corr1)

xrange <- range(coords3Dn_diag_corr1[,1])
yrange <- range(coords3Dn_diag_corr1[,2])
zrange <- range(coords3Dn_diag_corr1[,3])

dx <- diff(xrange) / 2
dy <- diff(yrange) / 2
dz <- diff(zrange) / 2

# punt central
points3d(centre[1], centre[2], centre[3], col="orange", size=5)

# Creu d'eixos centrada al punt de masses
segments3d(rbind(centre - c(dx, 0, 0),

```

```

    centre + c(dx, 0, 0)), col="orange", lwd=0.9)

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

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

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

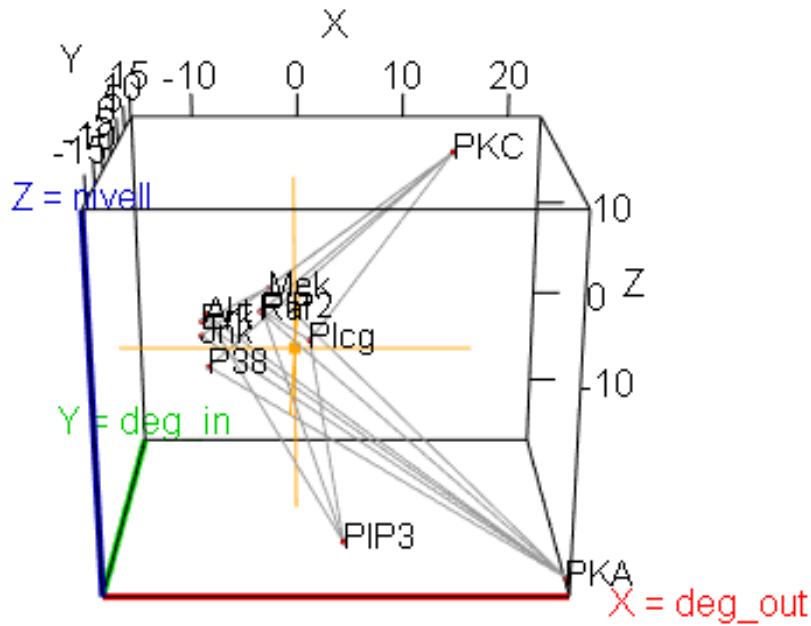
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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*

El flux causal geomètric general és un vector director que assenyala cap a on es dirigeix la causalitat a dins de l'espai geomètric: → FLUX CAUSAL VERTICAL: si hom sols defineix el flux causal general respecte de l'eix Z vertical, és a dir, projectant-lo sobre un únic eix, el flux causal roman limitat a un únic pla vertical. Pot esser útil si hom vol una primera exploració del flux causal o com a aproximació simplificada.

→ FLUX CAUSAL HOLÍSTIC: si hom defineix el flux causal general respecte dels tres eixos, és a dir, considerant alhora totes les components espacials, el flux causal pren una única direcció a dins de l'esfera unitària, cosa que explana el qualificatiu d'. Aquest darrer és el més robust a fi de determinar el flux causal global del model.

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

Fer 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).

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\_in i graus\_out a dins de cada tipus d'arestes.

Al subgraf vertical no hi poden haver col·lididors-fulla per definició d'arbre/bosc de la teoria de grafs, al DAG general podria semblar que hi ha de forma natural un col·lididor-fulla vertical però això no és real si hom accepta les definicions que impliquen la descomposició del DAG general en subgrafs vertical i retrolateral. Tanmateix, sí que cal tenir en compte els col·lididors-fulla retroverticals, és a dir, aquells en què una aresta retro i una de vertical xoquen a un node fulla del model geomètric 3D. Calen aquests col·lididors fulla retroverticals per tal de corregir el flux vertical boscoïde segons les branques que tinguin contraflux (aquests nodes són identificables a l'espai de transició entre els subgrafs vertical i retrolateral (vegeu annex1)).

Plot 3D que mostra en groc el retroflux i en blau el flux vertical si hom es basa en la definició d'arbre/bosc ##### Figura 16

```
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)

xrange <- range(coords3Dn_diag_corr1[,1])
yrange <- range(coords3Dn_diag_corr1[,2])
zrange <- range(coords3Dn_diag_corr1[,3])

dx <- diff(xrange) / 2
dy <- diff(yrange) / 2
dz <- diff(zrange) / 2

# punt central
points3d(centre[1], centre[2], centre[3], col="orange", size=5)

# Creu d'eixos centrada al punt de masses
segments3d(rbind(centre - c(dx, 0, 0),
                  centre + c(dx, 0, 0)), col="orange", lwd=0.9)

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

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

```

```

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

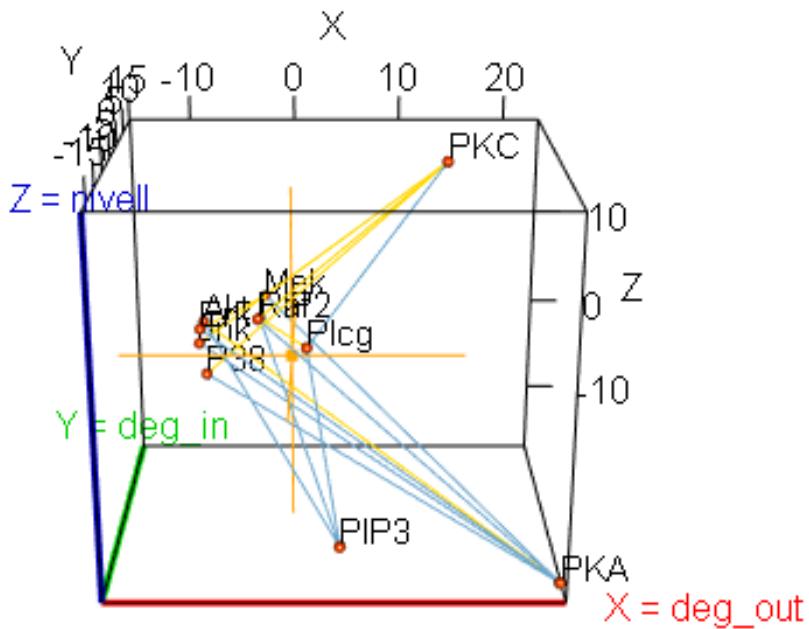
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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 provinents de tots els nodes fulla del subgraf retrolateral: siguin pseudoverticals (les arestes rebutjades en descompondre els col·lididors fulla per aplicació de la definició d'arbre/bosc), siguin retroverticals o siguin lateroverticals al model geomètric 3D.

La definició de flux de causalitat que hom lliga sols amb l'eix vertical Z (A) és senzilla però no és l'òptima, aquí sols treballa un únic l'eix Z en termes de flux causal. Tanmateix pot esser bona via exploratòria o simplificada.

Però de fet n'hi ha una segona, de manera, més holística i robusta (B): el flux causal general del model 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 tant si és un node final d'una cadena com si és el node medial d'un col·lididor fulla (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, l'anomenat centre de masses.

Amb (B) hom pot llegir el problema causal com a optimització geomètrica, perquè optimitzant els tres eixos hom pot llistar les vies causals prioritàries: 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. Fins i tot, hom pot detectar arestes absents al model de treball, en situacions d'asimetria, és susceptibles de plantejar una aresta absent com a hipòtesi: → si tots dos nodes tenen 0 outs i els mateixos IN, però uns IN amb diferència d'alçada vertical → si tots dos nodes tenen 0 outs i els mateixos IN, sense diferència d'alçada, però un IN col·lididor vertical i l'altre un IN col·lididor retrovertical, aquesta asimetria també juga a favor d'una aresta absent del col·lididor retrovertical cap al vertical.

### 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 que ens forneix la funció localTest() respecte del DAG combinatori i dels resultats de l'article de Sachs et al., hom assaja dues estratègies de càlcul del flux causal general (per general hom entén ).

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

convergeix al flux causal vertical o a dins del seu espai conoïdal (el concepte de con causal hom el veu més endavant).

→ o bé els angles del *flux causal holòstic* respecte de Z i YX (dos angles: un respecte l'eix Z i un altre respecte del pla YX a l'esfera unitària de direccions) i comparar-los amb els angles del flux local en estudi a fi de comprovar si que aquest darrer convergeix al flux causal holòstic.

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

En aquest treball no es defineix una distribució estadística en el sentit clàssic, sinó una configuració geomètrica induïda per un DAG causal. Les dades són expressades com a vectors a dins d' $R^3$  i la inferència no es basa en probabilitats, sinó en criteris geomètrics de coherència causal. En concret, la presència o absència d'una aresta es valida segons el seu efecte sobre el radi mitjà del con causal holòstic (concepte introduït més endavant amb la validació local del mètode), que mesura la convergència angular dels fluxos locals. Aquest criteri constitueix una reinterpretació geomètrica de la independència condicionada: una aresta és causalment coherent si redueix la dispersió angular global del sistema. El procediment s'aplica de manera local, aresta per aresta, evitant la combinatòria completa, i permet una inferència causal consistent sense necessitat d'assumir un model probabilístic explícit, procediment que hom pot sistematitzar a fi d'establir l'angle (Z) o angles (Z i XY) òptims del flux causal general.

## *5.6 ASSAIG DE VALIDACIÓ SIMULADA A ESCALA LOCAL A PARTIR DEL TIPUS CAUSALS MÍNIMS PROVINENTS DEL MODEL GENERAL TOPOEMPÍRIC (per mitjà de LocalTest() del DAG combinatori versus la diferència d'angles dels fluxos locals ϑ\_flux\_SENSE\_CONDICIONAMENT i ϑ\_flux\_AMB\_CONDICIONAMENT)*

**IMPORTANT:** És una pràctica simulada no real perquè les dades empíriques aquí emprades no responen a un model de DAG combinatori de 3 nodes i dues aretes, hom tria els tipus causals simples per tal d'assajar casos triviais abans de fer la validació local real del model causal geomètric 3D. És talment un assaig per tal de verificar i retocar el procediment per mitjà de tipus causals mínims (cadena, col·lididor...).

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

*càlcul del vector director segons sigui quin sigui el tipus causal: cadena, col·lididor o forca*

Hom pren com a flux vertical el del subgraf local respecte de l'eix Z (el correcte fóra fer-ho respecte del flux general del model geomètric 3D, el qual encara no ha estat calculat, ni la variant vertical ni

l'holística). Aquí hom no pretén validar el flux causal local del subgraf sinó com tractar amb subgrafs (que representen els tipus causals mínims) a fi d'establir una topologia de casos triviais que assentin el mètode de treball.

calcular l'angle del flux non-condicionat, aquí hom tracta amb la cadena

```
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 sobre el qual hom condicona 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_col·lididor_artificial) <- .GlobalEnv

theta_flux_col·lididor_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 correctes: 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 horitzontal)
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_colididor_Z_fisic(g_col2, coords3D_test2)
res_col2 # físic dóna 0 (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_colididor_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 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, llavore 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 contraintuïtiva) 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) #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  
    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_colloididor_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_colididor_artificial) <- .GlobalEnv

theta_flux_colididor_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_colididor_XY_fisic) <- .GlobalEnv
```

### 5.6.2 SUBGRAF AïLLAT: CADENA VERTICAL DE N=3 NODES AMB TEST NEGATIU

Hom agafa com a mostra una cadena vertical del DAG\_combinatori, dos nodes non-adjacents; després s'hi aplica el test d'independència condicionada per mitjà de la funció de dagitty d'R localTest() segons les dades empíriques de tots els nodes i hom compara el resultat del test amb el comportament del subgraf (que de ben segur no coincideix amb el model geomètric, atès que el sugraf és sols una mostra de cadena vertical del model).

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 (sols com a punt de referència, no té perquè coincidir el flux causal general del model amb l'eix Z; de fet, és molt difícil que coincideixin).

→*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, Y=15) en direcció eix Y ascendent, amb un angle respecte el pla YX de 23.77 graus.*

Hom assaja de comparar l'angle del flux local amb condicionament versus el flux local sense condicionament entès com a vector director de la cadena, comparar la diferència d'angles sols respecte l'eix vertical Z (sols com a referència arbitrària, el flux causal real general no és calculat) perquè tots dos angles són calculats sols respecte d'aquest eix Z. Així, doncs, la diferència entre el flux sense condicionament i el flux amb condicionament és de  $58.46 - 23.77 = 34.69$  graus.

Un comentari addicional és que hom agafa una cadena vertical com a subgraf del model, dibuixada de forma ideal com una recta a dins del DAG combinatori 2D; però de fet, a dins del model geomètric 3D, hom ja no pot analitzar la recta sols com a objecte en si mateix, autoreferencial, sinó que a dins de l'espai 3D caldrà distingir quina mena de cadena (vertical, lateral, retrògada...), ara ja hi han diverses menes de cadenes.

#TALLAFOC

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

```
arrels1 <- V(cadena3n_vertical)$name[degree(cadena3n_vertical, mode="in") == 0]
fulls1 <- V(cadena3n_vertical)$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] 9.621385 16.336391 32.365949
##
## $len_u

```

```

## [1] 37.51004
##
## $flux_z
## [1] 32.36595
##
## $theta_z
## [1] 30.36066
##
## $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: 30.36 graus

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

(theta_col_Plcg_Z <- theta_flux_colididor_Z_fisic(colididor_artificial, coords3Dn_diag_corr1))

## $r_in
## [1] -15.1634831   3.0130844  -0.6318497
##
## $len_r
## [1] 15.47285
##
## $flux_z
## [1] -0.6318497
##
## $theta_z
## [1] 87.65962
##
## $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: 87.66

```

A  $\theta_{\text{col\_Plcg\_XY}}$  i a  $\theta_{\text{col\_Plcg\_XY}}$ , hom hi resta 90 graus a fi d'obtenir l'angle respecte el pla XY (aquest no és constraintiu com sí que n'és l'angle respecte l'eix Y)

Ob: cal restar-hi 90 graus si hom vol l'angle respecte el pla XY (aquest no és constraintiu 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] 9.621385 16.336391 32.365949
##
## $len_u
## [1] 37.51004
##
## $flux_z
## [1] 32.36595
##
## $theta_xy
## [1] 59.63934
##
## $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: 59.64 graus

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

(theta_col_Plcg_XY <- theta_flux_colididor_XY_fisic(colididor_artificial, coords3Dn_diag_corr1))

## $r_in
## [1] -15.3942742 3.0589441 -0.6414666
##
## $len_r
## [1] 15.70835
##
## $flux_z
## [1] -0.6414666
##
## $theta_xy
## [1] 2.340382
##
## $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: 2.34

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,
    barlen = 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,
  barblen = 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))

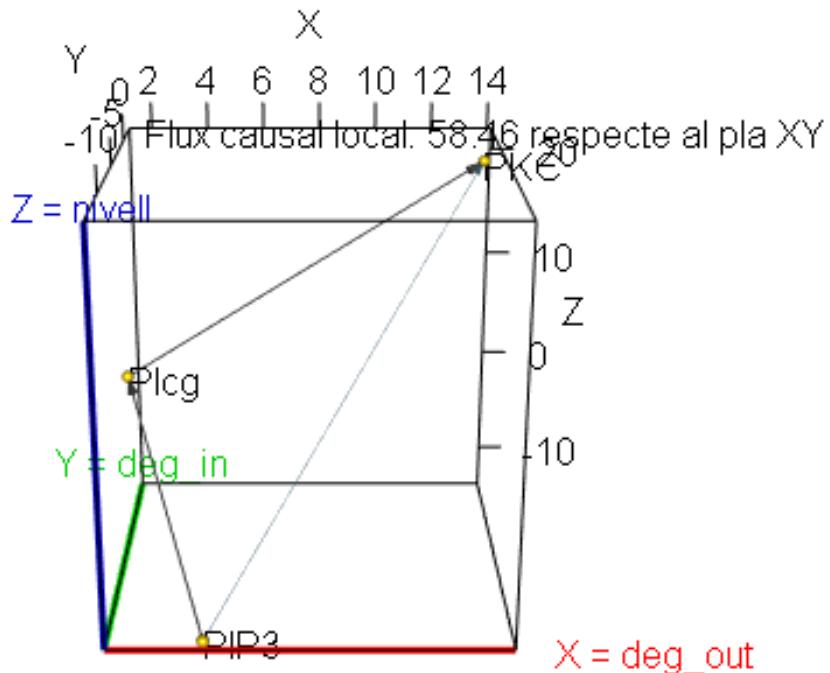
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

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

```



El resultat de localTest(), respecte de la cadena agafada del DAG\_combinatori, reporta que 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.

En general, en el marc del mètode geomètric 3D: →CADENA/FORCA: *CONDICIONAR VOL DIR QUE CANVIA L'ESTAT DEL CAMÍ D'ACTIU->BLOCAT, ÉS "COM SI" EL SENTIT DE LES ARESTES D'UNA CADENA O UNA FORCA ANESSIN CAP AL NODE CONDICIONAT*, és a dir, “com si” totes les arestes del node condicionat esdevinguessin. Aquest “com si” és estrictament il·lustratiu: *hom no modifica ni el graf, ni l'orientació real de les arestes; el que canvia és solsment l'estat del camí i, per tant, el comportament causal observable, que esdevé equivalent al d'una estructura complementària*. Així, el vector director resultant és el mateix que el d'un col·lididor entre nodes non-adjacents.

→COL·LIDIDOR: *ara bé, SI CONDICIONES PER UN COL·LIDIDOR, CANVIA L'ESTAT DEL CAMÍ DE BLOCAT->ACTIU, “com si” una aresta esdevingués i el vector director resultant és el mateix que el d'una cadena entre nodes non-adjacents; el vector director és com el d'una cadena i no pas d'una forca perquè no hi ha cap bifurcació sinó una dependència lineal constrenyida com en una cadena, és a dir, el vector resultant no surt de B sinó que connecta (obre el camí entre) A i C per mitjà de B. A i C passen a esser connectats per una dependència lineal induïda per la informació compartida sobre B. El camí A-C s'obre no perquè emeti causalitat sinó perquè el coneixement de B correlaciona A i C. Així, el vector director resultant és el mateix que el d'una cadena entre nodes non-adjacents.*

Si hom condiciona pel node medial, llavors el flux es trenca talment “com si” l’aresta Plcg→PKC esdevingués Plcg←PKC, la cadena es comporta com un col·lididor entre els nodes non-adjacents.

IMPORTANT: en tots els casos, l’expressió “com si” no descriu el canvi estructural del DAG, sinó una equivalència geomètrica del comportament causal observat, és a dir, una equivalència geomètrica del vector director resultant, *induïda pel canvi d’estat del camí (actiu/blocat)*.

```

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_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_blau_plot, "\n")
## Mòdul blau real (plot) : 37.51004

cat("Mòdul vermell real (plot) : ", mod_vermell_plot, "\n")
## Mòdul vermell real (plot) : 15.47285

# 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")
"")=====

# segeta 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]

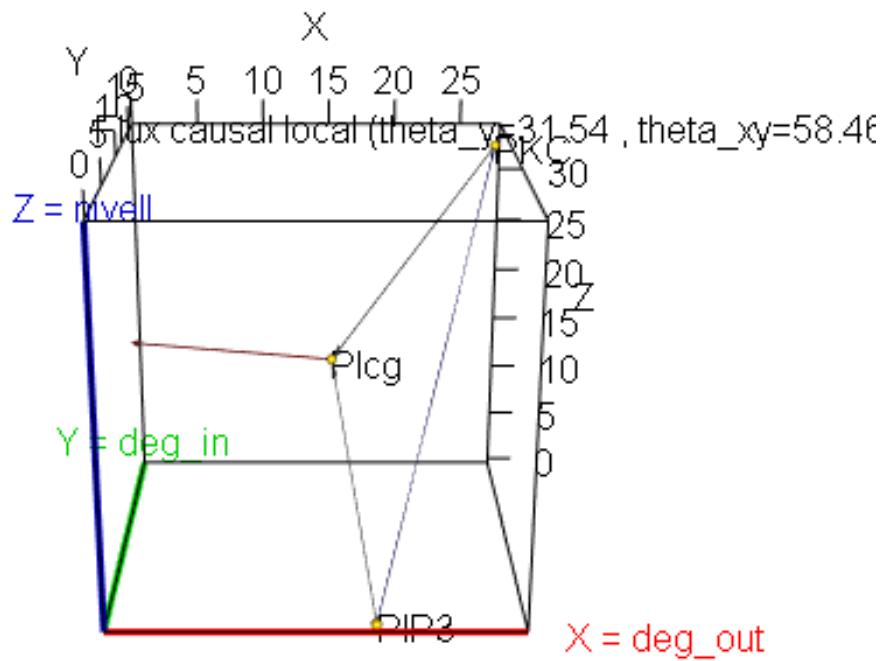
# eixos segons el contingut visible
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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] 18.81454

L_PKC
```

```

## [1] 21.74605

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

##          X          Y          Z
## -2.886444  9.697667 15.862241

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

##          X          Y          Z
## -12.507830 -6.638723 -16.503708

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

cat("Mòdul blau real (plot) : ", mod_blaue_plot, "\n")
## Mòdul blau real (plot) : 37.51004

cat("Mòdul vermell real (plot) : ", mod_vermelle_plot, "\n")
## Mòdul vermell real (plot) : 15.47285

```

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

```

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

arrels1 <- V(cadena3n_lateral)$name[degree(cadena3n_lateral, mode="in") == 0]
fulls1 <- V(cadena3n_lateral)$name[degree(cadena3n_lateral, mode="out") == 0]
noms_nodes1 <- V(cadena3n_lateral)$name

```

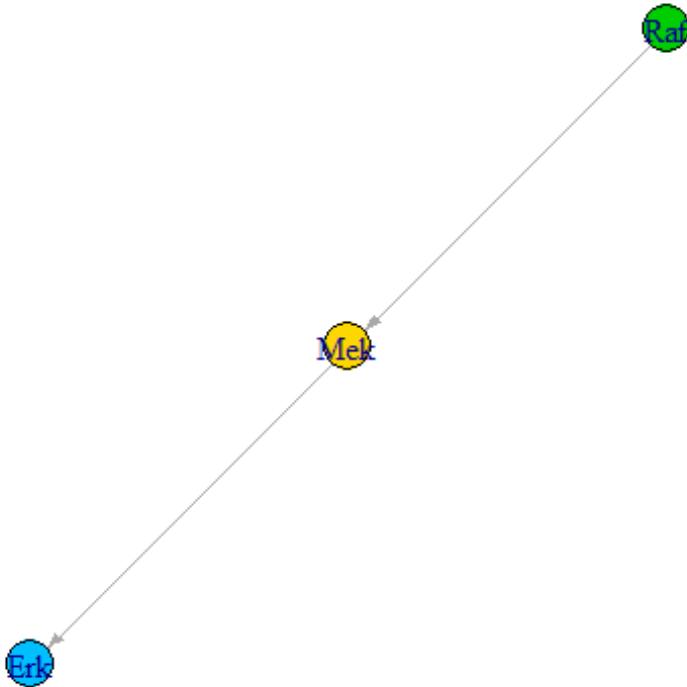
```

V(cadena3n_lateral)$role <- ifelse(!is.na(match(noms_nodes1, arrelos1)), "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_dia
g_corr1))

## $u_bar
## [1] -5.3719490  0.7586697 -1.3467696
##
## $len_u
## [1] 5.58992
##
## $flux_z
## [1] -1.34677
##
## $theta_z
## [1] 76.05867
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena Raf->Mek->Erk:", round(as.numeric(theta_ca
dena3n_lat$theta_z), 2), "graus\n")

## Angle del flux lateral de la cadena Raf->Mek->Erk: 76.06 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", "Er
k"))

(theta_col_Mek <- theta_flux_collididor_Z_fisic(collididor_artificial_lat, coords3D
n_diag_corr1))

## $r_in
## [1] 5.775075 24.272911 -2.722634
##
## $len_r
## [1] 25.09857
##
## $flux_z
## [1] -2.722634
##
## $theta_z
## [1] 83.77243
##
## $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: 83.77

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_di
ag_corr1))

## $u_bar
## [1] -5.3719490  0.7586697 -1.3467696
##
## $len_u
## [1] 5.58992
##
## $flux_z
## [1] -1.34677
##
## $theta_xy
## [1] 13.94133
##
## $signe
## [1] -1

cat("Angle del flux lateral de la cadena Raf->Mek->Erk:", round(as.numeric(theta_ca
dena3n_lat$theta_xy), 2), "graus\n")

## Angle del flux lateral de la cadena Raf->Mek->Erk: 13.94 graus

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

(theta_col_Mek <- theta_flux_collididor_XY_fisic(collididor_artificial_lat, coords3
Dn_diag_corr1))

## $r_in
## [1] 5.775101 24.273021 -2.722646
##
## $len_r
## [1] 25.09869
##
## $flux_z
## [1] -2.722646

```

```

## $theta_xy
## [1] 6.227565
##
## $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: 6.23

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 va
# lors negatius:
# el p1_flux té valors negatius i cal fer el recalc dels 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,
    barlen = 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 = "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]

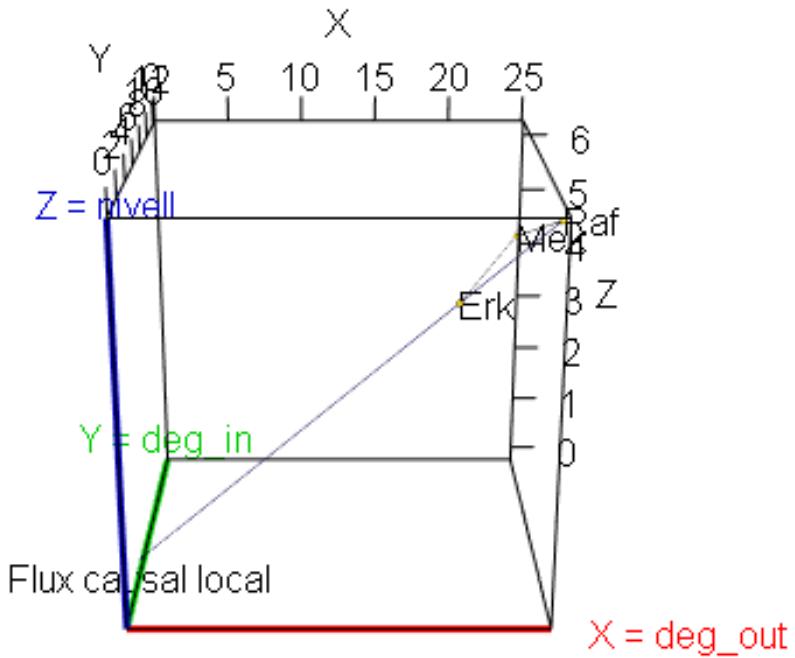
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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 va
lors negatius:
# el p1_flux té valors negatius i cal fer el recalc dels vectors abans de f
er 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_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) : 25.71129
cat("Mòdul vermell real (plot) : ", mod_vermell_plot, "\n")
## Mòdul vermell real (plot) : 25.09857
```

# 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,
  barrlen = 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)'

#=====
# segeta vermella
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]

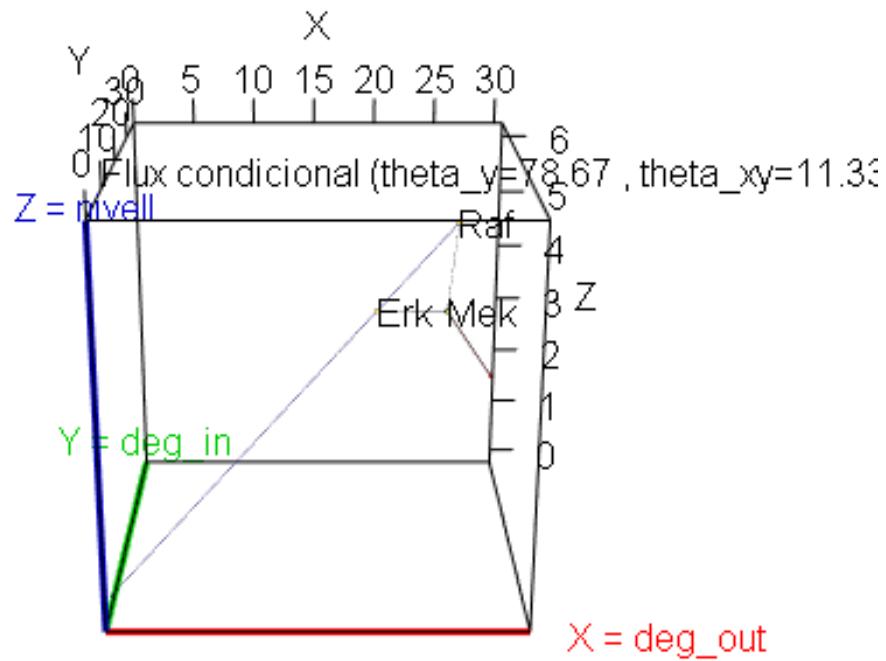
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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

Aquí hom planteja com caldria validar localment el model amb dues bateries de tests, positius i negatius, per tal d'encarar els resultats en termes geomètrics. Sols per a tipus causals de 3 nodes (caldria generalitzar les funcions a n nodes o per a composició de tipus causals a fi de traduir

geomètricament la llista d'independències condicionades de nodes non-adjacents de la funció localTest() )

Aquí hi han tots els tests positius de localTest(), és a dir, que les dades referen l'estructura del DAG\_combinatori 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 (plantejable 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] -21.8074745 -0.8692771 -21.5928073
##
## $len_u
## [1] 30.70132
```

```

## $flux_z
## [1] -21.59281
##
## $theta_z
## [1] 45.30614
##
## $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: 45.31 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] 31.41161 17.19623 53.92914
##
## $len_r
## [1] 64.73602
##
## $flux_z
## [1] 53.92914
##
## $theta_z
## [1] 33.58538
##
## $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: 33.59

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] -21.8074745 -0.8692771 -21.5928073
##
## $len_u
## [1] 30.70132
##
## $flux_z
## [1] -21.59281
##
## $theta_xy
## [1] 44.69386
##
## $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: 44.69 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] 31.42886 17.20567 53.95876
##
## $len_r
## [1] 64.77157
##
## $flux_z
## [1] 53.95876
##
## $theta_xy
## [1] 56.41462
##
## $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: 56.41

print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "P38"   "PKC"

####Jnk // PIP2 / PKC (planteable 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] -5.555168 -1.140258 -2.264064
##
## $len_u
## [1] 6.106231
##
## $flux_z
## [1] -2.264064
##
## $theta_z
## [1] 68.23631
##
## $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: 68.24 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] 39.580988 -2.576335 35.621862
##
## $len_r
## [1] 53.31237
##
## $flux_z
## [1] 35.62186
##
## $theta_z
## [1] 48.07383
##

```

```

## $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] -5.555168 -1.140258 -2.264064
##
## $len_u
## [1] 6.106231
##
## $flux_z
## [1] -2.264064
##
## $theta_xy
## [1] 21.76369
##
## $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: 21.76 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] 39.581464 -2.576366 35.622291
##
## $len_r
## [1] 53.31302
##
## $flux_z
## [1] 35.62229

```

```

## $theta_xy
## [1] 41.92617
##
## $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: 41.93

```

```
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] 17.013148 -1.858312 16.679113
##
## $len_u
## [1] 23.89756
##
## $flux_z
## [1] 16.67911
##
## $theta_z
## [1] 45.73787
##
## $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: 45.74 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] 39.448655 -2.141952 33.955084
##
## $len_r
## [1] 52.09349
##
## $flux_z
## [1] 33.95508
##
## $theta_z
## [1] 49.32176
##
## $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: 49.32

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] 17.013148 -1.858312 16.679113
##
## $len_u
## [1] 23.89756
##
## $flux_z
## [1] 16.67911
##
## $theta_xy
## [1] 44.26213
##
## $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: 44.26 graus

```

```

collididor_artificial_1 <- crea_colididor_artificial(g_bosc, c("PIP2", "PKC", "Erk"))

(theta_col_PKC <- theta_flux_colididor_XY_fisic(colididor_artificial_1, coords3Dn_diag_corr1))

## $r_in
## [1] 39.449321 -2.141989 33.955658
##
## $len_r
## [1] 52.09437
##
## $flux_z
## [1] 33.95566
##
## $theta_xy
## [1] 40.67824
##
## $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: 40.68

print(as_edgelist(colididor_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] -12.54338 17.61546 15.52745
##
## $len_u
## [1] 26.62222
##
## $flux_z
## [1] 15.52745
##
## $theta_z
## [1] 54.32044

```

```

##  

## $signe  

## [1] 1  

cat("Angle del flux lateral de la cadena PIP3->PKC->Akt:", round(as.numeric(theta_c  

adena3n_1$theta_z), 2), "graus\n")  

## Angle del flux lateral de la cadena PIP3->PKC->Akt: 54.32 graus  

collididor_artificial_1 <- crea_collididor_artificial(g_bosc, c("PIP3", "PKC", "Ak  

t"))  

(theta_col_PKC <- theta_flux_collididor_Z_fisic(collididor_artificial_1, coords3Dn_<br/>  

diag_corr1))  

## $r_in  

## [1] 31.72657 15.02910 49.11223  

##  

## $len_r  

## [1] 60.36936  

##  

## $flux_z  

## [1] 49.11223  

##  

## $theta_z  

## [1] 35.55783  

##  

## $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] -12.54338 17.61546 15.52745  

##  

## $len_u  

## [1] 26.62222  

##  

## $flux_z

```

```

## [1] 15.52745
##
## $theta_xy
## [1] 35.67956
##
## $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: 35.68 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] 31.78615 15.05732 49.20445
##
## $len_r
## [1] 60.48273
##
## $flux_z
## [1] 49.20445
##
## $theta_xy
## [1] 54.44217
##
## $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: 54.44

print(as_edgelist(collididor_artificial_1))

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "Akt"   "PKC"

```

*Mek /| PIP3 / PKC (plantejable 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] -16.86265 12.04085 -17.96448
##
## $len_u
## [1] 27.4236
##
## $flux_z
## [1] -17.96448
##
## $theta_z
## [1] 49.07475
##
## $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: 49.07 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] 26.386892 4.279782 50.145824
##
## $len_r
## [1] 56.82595
##
## $flux_z
## [1] 50.14582
##
## $theta_z
## [1] 28.06117
##
## $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] -16.86265 12.04085 -17.96448
##
## $len_u
## [1] 27.4236
##
## $flux_z
## [1] -17.96448
##
## $theta_xy
## [1] 40.92525
##
## $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: 40.93 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] 26.484033 4.295538 50.330432
##
## $len_r
## [1] 57.03515
##
## $flux_z
## [1] 50.33043
##
## $theta_xy
## [1] 61.93883
##
## $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: 61.94

print(as_edgelist(collididor_artificial_1))

```

```

##      [,1]  [,2]
## [1,] "PIP3" "PKC"
## [2,] "Mek"   "PKC"

```

### 5.6.5 CONCLUSIONS EN VALIDACIÓ TEÒRICA PER TIPUS CAUSALS MÍNIMS (no encara sobre el model geomètric 3D)

El primer pas més raonable fóra convertir aquests subgrafs mínims de 3 nodes com a grafs, és a dir, com a exemples trivials, de models geomètrics senzills basats sols en 3 nodes amb un joc de dades de sols 3 columnes construïdes a priori per tal de preveure'n el comportament. Entendre si l'heurística geomètrica aquí definida és consistent amb la semàntica causal en condicions pures. Partir dels tipus causals mínims de 3 nodes com a laboratori conceptual a fi de validar la semàntica causal del model geomètric. A dins d'aquest marc pur sense soroll biològic, sols causalitat “pura” teòrica, hom pot triar exactament: direcció vertical, inclinació lateral, distància, divergència, convergència..., comprovar si la diferència d'angles agrupa els tests o bé en positius o bé en negatius.

Hi ha qualche tests positiu que té diferències d'angle grans i qualche test negatiu que té diferències petites; el signe (+1/-1) sols diu si va cap amunt o cap avall, no pas convergent o divergent en sentit causal. Si hom treballa amb un model reduït subsidiari del model global hom veu que el criteri geomètric aplicat no és alineat amb la semàntica del LocalTest. En canvi, hom veurà que segons el cas treballat si el subgraf és representat a dins del model general i el seu vector director local juga a favor del flux global (els fluxos local i causal convergeixen a dins del con causal), llavors sí que el LocalTest és positiu; i si el seu vector director es desenganxa del flux causal (no convergeixen a dins del con causal) llavors hi juga en contra i el LocalTest és negatiu.

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\_combinatori biològic real les cadenes de 3 nodes no són rectes; a més, a dins d'un DAG\_combinatori hom no veu la disposició entre nivells ni tampoc lateralment.

Hom recorda una altra volta que 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) de forma aïllada del model causal geomètric general, hom ha emprat el joc de dades per tal d'identificar tipus causals mínims i establir un procediment de validació PROGRESSIVA del model; *CAL FER LA COMPARANÇA AMB EL FLUX CAUSAL GENERAL* o bé respecte l'eix vertical Z com a aproximació (flux causal vertical) o bé respecte dels tres eixos com a mètode robust (flux vertical holístic) i fer la comparança entre els resultats dels tests i, per exemple, el con casual (hom introduceix aquest concepte més endavant), cosa que implica treballar amb el plot del model geomètric 3D sencer, no pas un gràfic amb sols 3 nodes d'un tipus causal aïllat del model sencer.

```

$$ Node =(grau_{in}, grau_{out}, nivell) \\ Node\_A=(1,0,1) \\ Node\_B=(2,1,1) \\
Node\_C=(3,1,0) \\ . \\ \text{vector } \Delta X \space \Delta Y \space \Delta Z \space \text{Angle} \space \text{amb } Z \space AB \space [1, 1, 0] \space 90^\circ \space BC \space [1, 0, -1] \space 45^\circ \space AC \space [2, 1, -1] \space 66^\circ \\ $$

```

## **CONCEPTE DE CONVERGÈNCIA**

La convergència de fluxos local i causal no pot venir donada ni per l'angle sols de Z de forma aïllada, ni pel seu signe perquè dos vectors directors amb diferent signe poden tenir un angle molt petit entre ells i, en canvi, dos amb el mateix signe poden tenir sentits gairebé contraris. D'aquí que calgui una regió admissible de direccions, una regió angular tancada de direccions al voltant d'un vector director causal. La base del con causal, determinada per un radi angular, delimita un casquet únic de direccions a dins de l'esfera.

### **5.6.6 ASSAIG DE VALIDACIÓ SIMULADA A ESCALA LOCAL EN CAMINS DE MÉS DE TRES NODES**

Quan hom es demana sobre el flux causal d'un camí compost de diversos tipus causals mínims entre dos nodes non-adjacents, el més important és veure si el camí (sense condicionament o amb l'ajut de condicionament per qualche node) és actiu/obert o blocat/tancat; si és obert, hi haurà un únic vector director talment com si el camí es tractés d'un cadena de n nodes, aquí hi ha flux causal respecte del camí proposat. Ara bé, si el camí proposat entre dos nodes non-adjacents és blocat (amb o sense condicionament) llavors sembla que hi hagin d'haver tants de com tipus causals mínims; hom condensa el règim de camí blocat fent la mitjana vectorial d'aquests vectors directors locals en un únic vector direccional representatiu i després comparar-lo amb el vector director del camí actiu; hom fa la mitjana vectorial perquè no és gens probable que tots els vectors directors locals del camí blocat tinguin una mateixa direcció al casquet causal sinó que hom es demana per una mesura de coherència general d'aquests vectors directors locals.

Arribats aquí hom té un vector director provenint de l'escenari de camí actiu i una sèrie de vectors directors provinents de cada mínim tipus causal que compon el camí blocat. En lloc d'imposar d'entrada una ortogonalitat estricta (angle de 90 graus), hom empra una noció d'ortogonalitat direccional discreta definida per la dominància de components en eixos diferents de l'espai 3D metrificat; així, dos vectors són causalment desacoblats si no incideixen a dins del con causal, la base del qual delimita un casquet de l'esfera de direccions.

L'ortogonalitat estricta basada en el producte escalar, hom la considera un estadi metodològic posterior, aplicable quan la metrització geomètrica és estable i el model demostra coherència global. No és prudent exigir en aquest estadi ortogonalitat exacta a dins d'un espai causal mètric construït a partir de dades empíriques que contenen soroll fins que hom no valida la coherència global del model. Aquesta jerarquia permet distingir entre desacoblament causal qualitatiu (direccional) i desacoblament quantitatiu (angular), cosa que facilita una validació progressiva del model.

### **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'estructura geomètrica del model causal topoempíric 3D hom en descriu qualche tret rellevant:

→ 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 COMPARTIT amb PKA → PKA té un únic camí de flux cap a una fulla pròpria final Erk però 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 perquè els seus fluxos en sentit contrari al vertical xoquen amb els verticals provinents de PKA 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 lateralitzat de PKA. Hom ja intueix 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 laterovertical Erk és connectat a l'únic node col·lididor de flux estrictament vertical. L'efecte dels nodes arrels convergeix o no a Akt? Mentre que en termes d'inferència causal clàssica fóra si Akt i Erk són independents si hom condiciona per PKA, en termes d'inferència causal geomètrica cal veure si de la cadena lateral Raf->Mek->Erk o Raf->Mek->Erk->Akt convergeix angularment amb el flux general holístic (hom veurà si hi ha convergència angular, si el flux local cau a dins del con causal o si, fins i tot, hi ha intersecció entre el vector director del flux local amb el del flux general holístic a dins de l'esfera de direccions).

Hom recorda que el flux causal general pot esser calculat de dues maneres: → *FLUX CAUSAL GENERAL VERTICAL* (*triatge angular laxo*): a partir del subgraf boscoïde vertical, hom calcula la mitjana vectorial de tots els vectors directors provinents dels nodes fulla del susdit subgraf (tot corregint els nodes fulla que reben retrofluxos) amb ponderació dels mòduls. Al vertical hom hi optimitza sols l'eix Z (màxim Z; en canvi X i Y lliures), és a dir, amb Z dominant, d'aquí que hom obté totes les fulles del vertical. Al subgraf vertical, el triatge de fluxos fulla és feta sols amb la restricció angular sobre la component Z dels vectors directors normalitzats, mentre que les components X i Y romanen lliures. Això defineix un casquet canònic a l'esfera de direccions, amb eix Z, que recull la variabilitat lateral pròpria del flux vertical. La tria dels fluxos fulla hom la basa en criteris angulars, però la construcció del flux causal i del con associat és feta a partir de vectors complets, preservant-ne el mòdul, que actua com a pes causal.

→ *FLUX CAUSAL GENERAL HOLÍSTIC* (*triatge angular estricte*): a partir del subgraf retrolateral, hom calcula la mitjana vectorial de tots els vectors directors provinents dels nodes fulla de l'esmentat subgraf amb ponderació dels mòduls. A l'holístic hom optimitza tots tres eixos (màxim Z i Y, mínim X), cosa que duu a triar els nodes fulla del subgraf retrolateral. En el cas holístic, la tria dels fluxos fulla hom la fa imposant restriccions angulars combinades sobre les components Z, Y i X dels vectors de flux, cosa que defineix una regió angular més restringida a l'espai de direccions. Aquest triatge identifica les fulles del subgraf retrolateral, amb menys de dispersió direccional; la construcció del flux causal que en segueix és feta a partir de vectors complets, preservant-ne el mòdul com a pes causal en la mitjana vectorial.

Encara que al vertical hom optimitza sols l'eix Z (els angles tenen un rang de 0 a 90 graus. 0 si el vector és paral·lel a l'eix, vertical pur, pujant o baixant; 90 si el vector és perpendicular a l'eix, horitzontal total), tant al vertical com a l'holístic hom treballa gràficament en tres dimensions: al vertical amb Z màxim i eixos X i Y lliures; a l'holístic amb Z i Y màxims i X mínim. Per tant, gràficament hom treballa

amb angles de 360 graus, amb angles polars i azimutals alhora perquè són coordenades esfèriques, un vector 3D és formalitzat com: r com a mòdul del vector, theta (0–180) com n'és de vertical i phi (0–360) en quin sentit va en el pla horitzontal. Així hi ha un únic vector director causal (segons el subgraf triat per a calcular-ne el flux) a dins de l'esfera de direccions. A un vector 3D li calen aquests dos angles a fi de definir-se: ->theta: angle polar que va de 0 a 180 (inclinació respecte Z) ->phi: angle azimutal que va de 0 a 360 (orientació en el pla XY)

### 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.

*Figura 17 Mek->Erk>*

```
# 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_corr1, 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_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"]+offset,
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[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],
       texts = "Flux causal local", color = "black", cex = 0.75,
       depth_test="always")

aspect3d(1,1,1)

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)

bbox0 <- par3d("bbox")
xmin <- bbox0[1]; xmax <- bbox0[2]
ymin <- bbox0[3]; ymax <- bbox0[4]
zmin <- bbox0[5]; zmax <- bbox0[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))

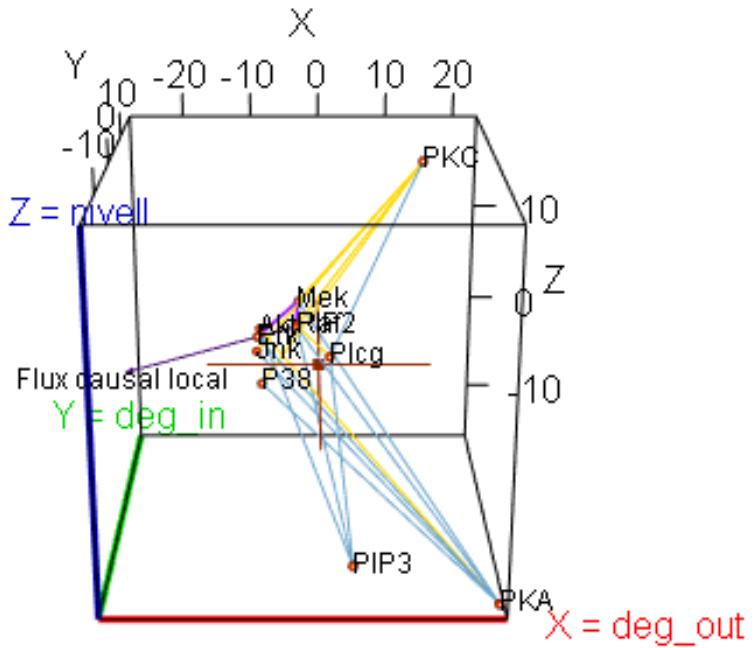
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

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 (recordeu la proporció de tests vàlids respecte dels fallits segons la funció de localTest()), 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)

*Figura 18*

```
# 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_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_corr1, 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_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"],
       coords3Dn_diag_corr1[, "Y"],
       coords3Dn_diag_corr1[, "Z"],
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),col = "blue",
#            lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[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],
       texts = "Flux vertical 1", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr1[3,] ,
  p1 = coords3Dn_diag_corr1[2,] ,

```

```

    col = "blue",
    type = "rotation",
    lwd = 2,
    width = 0.3,
    barblen = 0.02
)
p5721 = coords3Dn_diag_corr1[2,]
text3d(p5721[1]+4, p5721[2], p5721[3],
       texts = "Flux vertical 2", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr1[3,] ,
  p1 = coords3Dn_diag_corr1[9,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
p5722 = coords3Dn_diag_corr1[9,]
text3d(p5722[1]-4, p5722[2], p5722[3],
       texts = "Flux vertical 3", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = coords3Dn_diag_corr1[4,] ,
  p1 = coords3Dn_diag_corr1[8,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
p5723 = coords3Dn_diag_corr1[8,]
text3d(p5723[1]-4, p5723[2], p5723[3],
       texts = "Flux vertical 4", color = "black", cex = 0.75,
       depth_test="always")

aspect3d(1,1,1)

```

```

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]

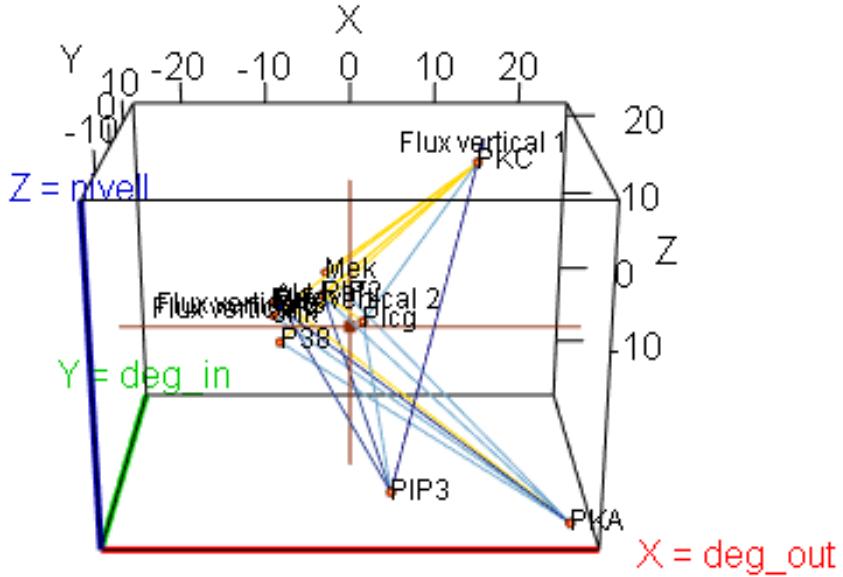
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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

```



### 5.7.3 FLUX CAUSAL RETROVERTICAL del model geomètric topoempíric general

Hi han 4 col·lididors retroverticals #### Figura 19

```
# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense va
lors 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 <- induced_subgraph(g_bosc, vids = nodes_colididor_PKA_P38_PKC)
coords_colididor_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_colididor_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_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 <- induced_subgraph(g_bosc, vids = nodes_colidador_PKA_Jnk_PKC)
coords_colidador_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_colidador_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_colidador_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_colidador_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_colidador_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_colidador_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_colidador_PKA_Raf_PKC["Raf",c("X","Y","Z")]
p1_flux4 <- p0_flux4 + len_flux4 * u_dir4

```

```

# 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

# _____ [2] _____ [2]

# 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] _____ [2]

open3d()

## wgl
## 46

plot3d(coords3Dn_diag_corr1, 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_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"],
       coords3Dn_diag_corr1[, "Y"],
       coords3Dn_diag_corr1[, "Z"],
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2), cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),col = "blue",
#            lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[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", cex = 0.75,

```

```

    depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux2[1], p1_flux2[2], p1_flux2[3] + 4,
       texts = "Flux retrovertical 2", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux3[1], p1_flux3[2], p1_flux3[3] + 4,
       texts = "Flux retrovertical 3", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux4[1], p1_flux4[2], p1_flux4[3] + 4,
       texts = "Flux retrovertical 4", color = "black", cex = 0.75,
       depth_test="always")

aspect3d(1,1,1)

```

```

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]

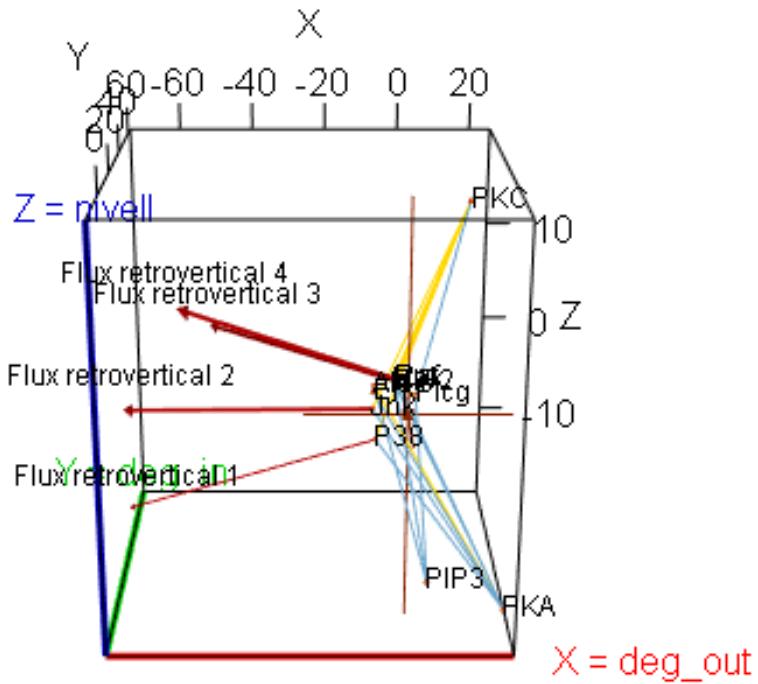
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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

```



#### 5.7.4 FLUX CAUSAL CORREGIT VERTICAL del model geomètric topoempíric general

Figura 20

```

nodes_cadena_PIP3_Plcg_PKC <- c("PIP3", "Plcg", "PKC")
cadena3n_PIP3_Plcg_PKC <- induced_subgraph(g_bosc, vids = nodes_cadena_PIP3_Plcg_PK C)
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_co rr1)
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_colididor_PKA_Mek_PKC["Mek",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3

#_____

```

---

```

nodes_colididor_PKA_Raf_PKC <- c("PKA", "Raf", "PKC")
collididor3n_PKA_Raf_PKC <- induced_subgraph(g_bosc, vids = nodes_colididor_PKA_Raf_PKC)
coords_colididor_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_colididor_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_colididor_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_corr1, 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 {
    return("darkred") # flux horizontal (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_corr1[v1, ], coords3Dn_diag_corr1[v2, ]),
    col = edge_colors[i], lwd = 0.9, cex = 0.75,
    depth_test="always")
}

#offset <- seq(-0.04, 0.04, Length.out=nrow(coords3Dn_diag_corr1))
text3d(coords3Dn_diag_corr1[, "X"],
  coords3Dn_diag_corr1[, "Y"],
  coords3Dn_diag_corr1[, "Z"],
  texts = rownames(coords3Dn_diag_corr1),
  adj = c(0, 0.2),
  cex = 0.75,
  depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),col = "blue",
#" , lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[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],
  texts = "Flux retrovertical 1", color = "black", cex = 0.75,
  depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "red",

```

```

type = "rotation",
lwd = 2,
width = 0.3,
barblen = 0.02
)
text3d(p1_flux2[1], p1_flux2[2], p1_flux2[3],
       texts = "Flux retrovertical 2", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux3[1], p1_flux3[2], p1_flux3[3],
       texts = "Flux retrovertical 3", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "red",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux4[1], p1_flux4[2], p1_flux4[3],
       texts = "Flux retrovertical 4", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava1
arrow3d(
  p0 = p0_flux0,
  p1 = p1_flux0,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,

```

```

        barblen = 0.02
    )
text3d(p1_flux0[1]+3, p1_flux0[2], p1_flux0[3],
       texts = "Flux vertical 1", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr1[3,] ,
  p1 = coords3Dn_diag_corr1[2,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(coords3Dn_diag_corr1[2,][1]+3, coords3Dn_diag_corr1[2,][2], coords3Dn_diag_c
orr1[2,][3],
       texts = "Flux vertical 2", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = coords3Dn_diag_corr1[3,] ,
  p1 = coords3Dn_diag_corr1[9,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(coords3Dn_diag_corr1[9,][1]+3, coords3Dn_diag_corr1[9,][2], coords3Dn_diag_c
orr1[9,][3],
       texts = "Flux vertical 3", color = "black", cex = 0.75,
       depth_test="always")

# segeta blava4
arrow3d(
  p0 = coords3Dn_diag_corr1[4,] ,
  p1 = coords3Dn_diag_corr1[8,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)

```

```

)
text3d(coords3Dn_diag_corr1[8,][1]+3, coords3Dn_diag_corr1[8,][2], coords3Dn_diag_corr1[8,][3],
       texts = "Flux vertical 4", color = "black", cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[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],
       texts = "FC local", color = "purple", cex = 0.75,
       depth_test="always")

aspect3d(1,1,1)

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]

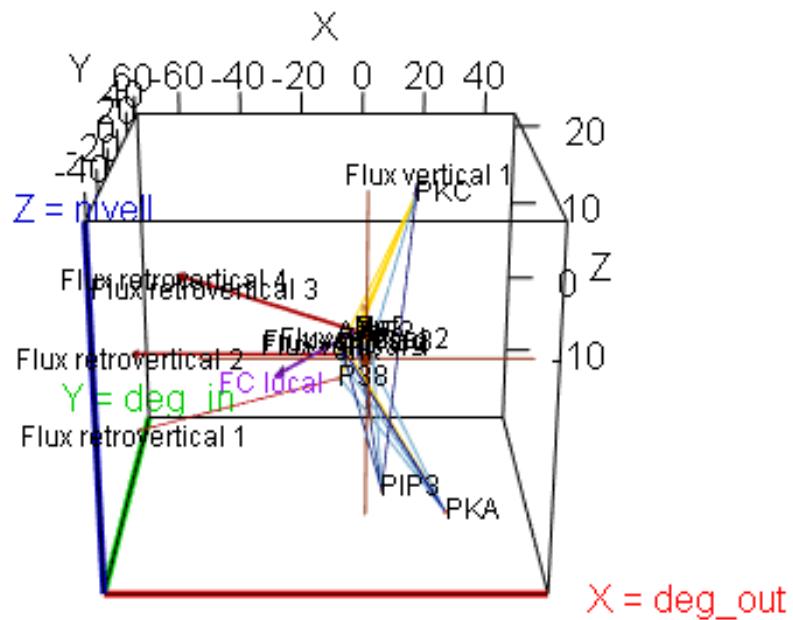
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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

```



## 5.8 Càlcul de l'angle de flux causal vertical (Z amb XY lliures) del model topoempíric: theta\_flux\_vertical\_corregit

FLUX CAUSAL VERTICAL (8 col·lididors fulla) AMB COORDENADES coords3Dn\_diag\_corr1 (SENSE ARESTA Erk->Akt)

taula\_fluxos verticals (i local)

Taula 16

```
# Flux Local cadena
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1)

# Fluxos verticals (1 cadena i 3 arestes verticals)
flux_vertical1 <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1)

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_corr1)

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_corr1)

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_corr1)

# Fluxos retroverticals (4 col·lididors)
(flux_retro_vertical1 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1))

## $r_in
## [1] -52.367486 17.501757 -8.767784
##
## $len_r
## [1] 55.90652
##
## $flux_z
## [1] -8.767784
```

```

## 
## $theta_z
## [1] 80.97709
##
## $signe
## [1] -1

(flux_retro_vertical2 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC, co
ords3Dn_diag_corr1))

## $r_in
## [1] -53.736669 20.610424 -3.474299
##
## $len_r
## [1] 57.65839
##
## $flux_z
## [1] -3.474299
##
## $theta_z
## [1] 86.54546
##
## $signe
## [1] -1

(flux_retro_vertical3 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Mek_PKC, co
ords3Dn_diag_corr1))

## $r_in
## [1] -42.189610 42.924838 -1.520733
##
## $len_r
## [1] 60.20646
##
## $flux_z
## [1] -1.520733
##
## $theta_z
## [1] 88.55263
##
## $signe
## [1] -1

(flux_retro_vertical4 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Raf_PKC, co
ords3Dn_diag_corr1))

## $r_in
## [1] -42.260338 18.011805 2.487328
##
## $len_r
## [1] 46.00596

```

```

## 
## $flux_z
## [1] 2.487328
## 
## $theta_z
## [1] 86.90077
## 
## $signe
## [1] 1

```

*Taula 17*

```

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$nodes[2],
      node_3 = f$nodes[3],
      node_4 = ifelse(length(f$nodes) == 4, f$nodes[4], 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 × 12
##   flux node_1 node_2 node_3 node_4 u_bar_X u_bar_Y u_bar_Z modul flux_z theta_z
##   <chr> <chr>  <chr>  <chr>  <lgl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 flux... Raf     Mek     Erk     NA      -5.37    0.759    -1.35    5.59    -1.35    76.1
## 2 flux... PIP3   Plcg   PKC     NA      9.62     16.3     32.4     37.5     32.4     30.4
## 3 flux... PIP3   PIP2   <NA>    NA      -7.39    18.2     15.7     25.1     15.7     51.4
## 4 flux... PIP3   Akt    <NA>    NA     -12.5    17.6     15.5     26.6     15.5     54.3
## 5 flux... PKA    Erk    <NA>    NA     -31.3    19.6     17.1     40.7     17.1     65.1
## 6 flux... PKA    P38    PKC     NA     -52.4    17.5     -8.77    55.9     -8.77    81.0
## 7 flux... PKA    Jnk    PKC     NA     -53.7    20.6     -3.47    57.7     -3.47    86.5
## 8 flux... PKA    Mek    PKC     NA     -42.2    42.9     -1.52    60.2     -1.52    88.6
## 9 flux... PKA    Raf    PKC     NA     -42.3    18.0     2.49     46.0     2.49     86.9
## # [i] 1 more variable: signe <dbl>

mitjana_vectorial_global <- function(taula_fluxos) {
  #hom n'exclou el flux local objectiu: NO entra en la mitjana, no pot contribuir en la definició del flux causal
  taula2 <- taula_fluxos %>%
    dplyr::filter(flux != "flux_objectiu")

  #Matriu dels vectors direcció (u_bar_X, u_bar_Y, u_bar_Z) dels col·lididors fulla a R^3
  V <- as.matrix(taula2[, c("u_bar_X", "u_bar_Y", "u_bar_Z")])

  #aquí hom recupera el mòdul de cada col·lididor fulla, cada mòdul és interpretat com a pes causal=> mòduls com a pesos que ja vénen del sistema
  w <- sqrt(rowSums(V^2))

  #hom normalitza cada vector: hom decanta direcció pura (U_norm) de la intensitat (w), cosa que permet de controlar explícitament com entra el pes
  U_norm <- V / w

  #suma vectorial ponderada: és la mitjana vectorial ponderada dels col·lididors fulla, JUST AQUÍ NEIX EL FLUX CAUSAL GLOBAL
  v_sum <- colSums(w * U_norm)

  #hom normalitzem la direcció resultant sols a fi de calcular angles o comparar direccions, no redefineix el flux, sols la seva direcció unitària
  v_global <- v_sum / sqrt(sum(v_sum^2))

  # =====
  # PART ORIGINAL (ANGLE VERTICAL respecte de Z)
  # =====

  #angle vertical RESULTANT (respecte Z): mesura l'orientació global del flux causal respecte de l'eix Z. És una projecció vertical, no pas una anàlisi causal nova
  cos_theta_Z <- v_global[3]                      # component Z
  angle_vertical <- acos(cos_theta_Z) * 180 / pi

```

```

# angle del flux local objectiu (ja és a la taula!): hom n'extreu l'angle vertical ja computat
angle_objectiu <- taula_fluxos$theta_z[
  taula_fluxos$flux == "flux_objectiu"
]

# diferència d'angles entre el flux causal i el flux local objectiu
diferencia_angles <- abs(angle_objectiu - angle_vertical)

# =====
# PART AFEGIDA (CON CAUSAL: dispersió angular)
# =====

# angles entre cada fulla i el vector causal global (3D): tots els angles són respecte el flux causal (no pas respecte Z ni d'altres col·lididors fulla )

angles_fulla <- acos(
  (v %*% v_global) /
  (sqrt(rowSums(v^2)) * sqrt(sum(v_global^2)))
) * 180 / pi

# radi angular del con
angle_mean <- sum(w * angles_fulla) / sum(w) # La mitjana ponderada dels angles dels col·lididors fulla respecte el flux causal: és un escalar que descriu la dispersió angular mitjana (no és una mitjana vectorial ni defineix cap flux)==> SERVEIX EXCLUSIVAMENT PER A L'OBERTURA DEL CON(el radi és una conseqüència de l'obertura quan hom en fixa l'alçada ==> radi= alçada con(h) * tan(theta) obertura angular)
angle_max <- max(w * angles_fulla) / max(w) #és descriptiu, no pas estructural (outlier de la dispersió angular)

# =====
# hom retorna tots els valors
# =====

list(
  # original
  vectors_utilitzats = taula2$flux,
  v_sum = v_sum,
  v_global = v_global,
  angle_vertical = angle_vertical,
  angle_objectiu = angle_objectiu,
  diferencia_angles = diferencia_angles,
  
  # afegit (con)
  angles_fulla = as.numeric(angles_fulla),
  angle_mean = angle_mean,
  angle_max = angle_max
)
}

```

```
taula_general_fluxos_fulla8 amb coords3Dn_diag_corr1
```

### Taula 18

```
(taula_general_fluxos_fulla8 <- 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
## -232.20869 170.77021 69.42056
##
## $v_global
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.7832091  0.5759852  0.2341463
##
## $angle_vertical
##   u_bar_Z
## 76.45869
##
## $angle_objectiu
## [1] 76.05867
##
## $diferencia_angles
##   u_bar_Z
## 0.4000253
##
## $angles_fulla
## [1] 75.40442 37.49494 27.53804 12.02962 28.69080 22.82125 17.52730 16.74297
##
## $angle_mean
## [1] 27.84487
##
## $angle_max
## [1] 46.97873
```

Aquesta funció permet determinar l'angle entre el flux local que és en estudi i el flux causal global

```
angle_interseccio_fluxos <- function(p0, v_h, q0, v_l) {
```

```
# Normalitzem per calcular angle
vh_n <- v_h / sqrt(sum(v_h^2))
vl_n <- v_l / sqrt(sum(v_l^2))

# Angle entre fluxos
angle <- acos(sum(vh_n * vl_n)) * 180 / pi
```

```

# Vector entre origens
w0 <- p0 - q0

# Productes interns necessaris
a <- sum(v_h * v_h)
b <- sum(v_h * v_l)
c <- sum(v_l * v_l)
d <- sum(v_h * w0)
e <- sum(v_l * w0)

# Denominador del sistema
denom <- a * c - b * b

if (abs(denom) < 1e-12) {
  # Rectes gairebé paral·leles
  t <- 0
  s <- d / b # aproximació
} else {
  t <- (b * e - c * d) / denom
  s <- (a * e - b * d) / denom
}

# Punts de mínima distància
P <- p0 + t * v_h
Q <- q0 + s * v_l

# Distància mínima
dist_min <- sqrt(sum((P - Q)^2))

return(
  list(
    angle_graus = angle,
    distancia_minima = dist_min,
    punt_holistic = P,
    punt_local = Q,
    vector_PQ = Q - P
  )
)
}

```

interseccio\_fluxos8 amb coords3Dn\_diag\_corr1

Taula 19

```

taula_general_fluxos_fulla8 <- taula_fluxos_general(fluxos)
#resultat8 <- mitjana_vectorial_global(taula_general_fluxos_fulla8)

res_holistic8 <- mitjana_vectorial_global(taula_general_fluxos_fulla8)

v_holos8 <- res_holistic8$v_sum

```

```

p08 <- mean(coords3Dn_diag_corr1)

v_local8 <- as.numeric(res_flux$u_bar)
q08 <- p0_flux

(intersecció_fluxos8 <- angle_interseccio_fluxos(p08, v_holos8, q08, v_local8))

## $angle_graus
## [1] 39.24663
##
## $distancia_minima
## [1] 1.664344
##
## $punt_holistic
##   u_bar_X   u_bar_Y   u_bar_Z
## -4.588844  3.374714  1.371870
##
## $punt_local
##      X          Y          Z
## -5.037511  2.286354  2.548396
##
## $vector_PQ
##      X          Y          Z
## -0.4486667 -1.0883596  1.1765258

```

Matriu de distàncies dels punts de destinació dels vectors directors i la seva distància amb el centre de masses segons aquests 9 punts a la diagonal

```

# taula de puntes dels vectors directors

# hom n'agafa sols amb els fluxos generals (sense el flux Local 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)

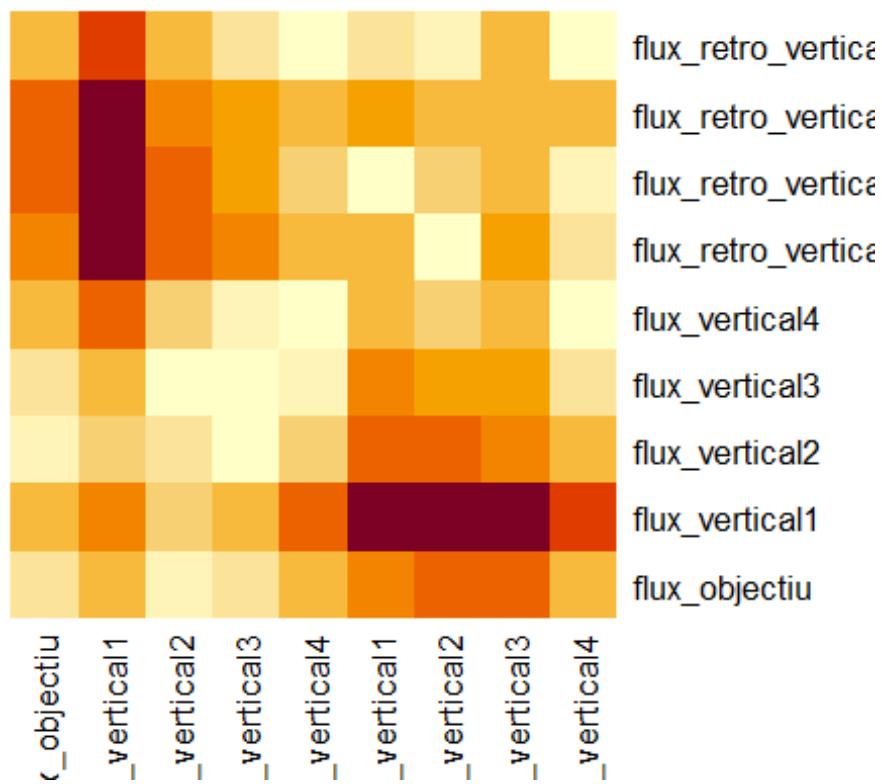
for (i in 1:n) {
  for (j in 1:n) {
    D[i, j] <- sqrt(sum((puntes[i, ] - puntes[j, ])^2))
  }
}

# 5. Substituïm la diagonal per distància al centre de masses
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_objec tiu	flux_v eritical 1	flux_v eritical 2	flux_v eritical 3	flux_v eritical 4	flux_retr o_vertical 1	flux_retr o_vertical 2	flux_retr o_vertical 3	flux_retr o_vertical 4
flux_obje ctiu	26.14 426	40.26 347	19.82 191	22.73 204	37.56 987	53.24078 2	54.82698 7	55.97818 0	37.84905
flux_verti cal1	40.26 347	51.02 992	29.22 931	37.31 119	54.31 232	85.24490 8	83.61157 0	80.33203 9	64.07400
flux_verti cal2	19.82 191	29.22 931	22.77 575	10.37 305	29.38 451	57.89003 1	56.16300 9	52.54829 8	37.15804
flux_verti cal3	22.73 204	37.31 119	10.37 305	13.95 404	19.12 784	49.05733 6	46.70869 8	47.15999 2	27.47694
flux_verti cal4	37.56 987	54.31 232	29.38 451	19.12 784	13.26 342	36.58572 8	31.70226 2	40.42002 9	14.57977
flux_retr o_vertical 1	53.24 078	85.24 491	57.89 003	49.05 734	36.58 573	35.61218 9	9.470083 9	42.61930 5	22.52944
flux_retr o_vertical 2	54.82 699	83.61 157	56.16 301	46.70 870	31.70 226	9.470083 6	33.41871 6	37.91724 6	19.57709
flux_retr o_vertical 3	55.97 818	80.33 203	52.54 829	47.15 999	40.42 002	42.61930 2	37.91723 6	37.56522 6	37.91451
flux_retr o_vertical 4	37.84 905	64.07 400	37.15 804	27.47 694	14.57 977	22.52944 2	19.57709 5	37.91451 5	14.97171

## DEFINICIONS CLAU

→Esfera de direccions: hom considera l'esfera de direccions centrada a l'origen geomètric del sistema, de radi  $R_{esfera}$ , que representa el conjunt complet de direccions possibles a dins de l'espai tridimensional.

→CASQUET ESFÈRIC COM A SUBCONJUNT MESURABLE DE L'ESFERA: la fracció esfèrica associada al casquet causal forneix una mesura invariant d'escala de la concentració angular del flux causal emergent (si és causalment concentrada o nul·la si el casquet depassa la semiesfera)

→Vector causal global: el vector causal global obtingut mitjançant la mitjana vectorial ponderada dels col·lididors fulla. Aquest vector defineix una direcció privilegiada del sistema, que actua com a eix del con causal.

→Con causal: hom el defineix com el conjunt de direccions que formen un angle més petit o igual que  $\theta_{con}$  amb el vector  $v_{causal}$ , on

$$\theta_{con}$$

correspon a l'angle mitjà (mitjana vectorial com a definició de flux causal emergent, perquè inclou tant intensitat com coherència direccional) dels col·lididors fulla respecte del flux causal global.

La interpretació del con causal com a mesura de coherència direccional sols és significativa quan el nombre de col·lididors fulla és més gran que no un. Un angle d'obertura mitjà petit assenyala una concentració direccional alta dels vectors locals (col·lididors fulla) al voltant del vector causal holístic. Això suggereix baixa dispersió angular i, per tant, una alta consistència geomètrica interna del model en termes de direcció del flux. Aquesta conclusió és vàlida sempre que l'angle s'hagi calculat exclusivament a partir de vectors unitats (direccions) i que la tria del centre geomètric ( $p_0$ ) i del vector holístic sigui coherent amb la definició del flux causal.

En el cas límit d'un únic col·lididor, l'angle d'obertura és nul per construcció i no pot esser interpretat com a evidència de fiabilitat del model. Aquí el con causal degenera i perd capacitat informativa. En aquest escenari, la inferència no pot basar-se en dispersió angular, sinó en la comparació directa de direccions entre el flux local i el flux holístic, complementada amb criteris de mòdul.

→Definició del casquet angular: hom el defineix com la intersecció entre l'esfera de direccions i el con causal. Aquest conjunt constitueix un subconjunt mesurable de l'esfera, delimitat per un angle sòlid centrat en la direcció causal global.

→Interpretació geomètrica i causal: el casquet angular representa el conjunt de direccions compatibles amb el flux causal emergent del sistema, tenint en compte la variabilitat angular observada als col·lididors fulla. No depèn de la distribució espacial del núvol de punts, sinó exclusivament de la geometria angular induïda pel flux causal.

→Funció del radi de l'esfera: aquest fixa únicament l'escala geomètrica de representació i garanteix que l'esfera engloba el sistema sencer. La informació causal rellevant resideix en l'obertura/tancament angular del casquet.

### **ASSUMPCIÓ ESFERA DE DIRECCIONS:**

hom assumeix que el radi de l'esfera de direccions és donada pel mòdul del vector director del flux causal global. Aquesta tria garanteix que l'esfera representa el conjunt sencer de direccions possibles, mentre que els vectors causals locals (col·lididors fulla) poden travessar-la sense restricció.

Tanmateix, per tal d'assegurar que tots els nodes i arestes del model romanen continguts a dins de l'esfera geomètrica associada al núvol de punts, hom imposa la condició següent: si el mòdul de flux xausal glabol és més petita que no el radi mínim calgit per contenir el núvol de punts, el radi de l'esfera és establerta com el màxim de distàncies del núvol respecte el centre geomètric (anomenat aquí el centre de masses).

Hom hi afegeix la seteta del flux causal general vertical (amb l'origen del seu vector director al centre de masses). SENSE forçar l'aresta absent Erk->Akt, el flux local i el general vertical no sembla que convergeixin: una és ascendent i l'altre té un sentit descendent respecte de l'eix Z. Cal després assajar

un nou model tot forçant l'aresta Erk->Akt, cosa que implica recalcular les coodenades topoempíriques de nou i tornar a fer el plot.

Hom pot veure com el vector director del flux local (vegeu segeta lila) travessa el con causal però surt fora de l'espai conoïdal; el seu sentit no sembla convergir a dins del casquet de direccions. SENSE forçar l'aresta absent Erk->Akt el resultat sembla, per tant, de non-convergència.

```

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]

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] _____[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_colididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1[c("PKA", "Jnk", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux2 <- longitud_cami_fisic(colididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1)

res_flux2 <- theta_flux_colididor_Z_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

# _____
# Longitud camí
nodes_colididor_PKA_Mek_PKC <- c("PKA", "Mek", "PKC")
collididor3n_PKA_Mek_PKC <- induced_subgraph(g_bosc, vids = nodes_colididor_PKA_Mek_PKC)
coords_colididor_PKA_Mek_PKC <- coords3Dn_diag_corr1[c("PKA", "Mek", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux3 <- longitud_cami_fisic(colididor3n_PKA_Mek_PKC, coords3Dn_diag_corr1)

res_flux3 <- theta_flux_colididor_Z_fisic(colididor3n_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_colididor_PKA_Mek_PKC["Mek",c("X","Y","Z")]
p1_flux3 <- p0_flux3 + len_flux3 * u_dir3

# _____
# Longitud camí
nodes_colididor_PKA_Raf_PKC <- c("PKA", "Raf", "PKC")
collididor3n_PKA_Raf_PKC <- induced_subgraph(g_bosc, vids = nodes_colididor_PKA_Raf_PKC)
coords_colididor_PKA_Raf_PKC <- coords3Dn_diag_corr1[c("PKA", "Raf", "PKC"), c("X", "Y", "Z")]

# Longitud camí
len_flux4 <- longitud_cami_fisic(colididor3n_PKA_Raf_PKC, coords3Dn_diag_corr1)

res_flux4 <- theta_flux_colididor_Z_fisic(colididor3n_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]
```

Figura 21 <Plot coords3Dn\_diag\_corr1 amb 8 vectors directors de flux SENSE ARESTA Erk->Akt>

```
open3d()
```

```
## wg1
## 52
```

```
plot3d(coords3Dn_diag_corr1, 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)
  } else {
    return("gray70")
  }
})
```

```
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_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"],
       coords3Dn_diag_corr1[, "Y"],
       coords3Dn_diag_corr1[, "Z"],
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2), cex = 0.75,
```

```

    depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),col = "blue",
#"lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[8, ]),col = "blue",
#"lwd = 1.5)

# segeta vermella1
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]-2,
       texts = "Flux retrovertical 1", color = "green", cex = 0.6,
       depth_test="always")

# segeta vermella2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux2[1], p1_flux2[2], p1_flux2[3]-2,
       texts = "Flux retrovertical 2", color = "green", cex = 0.6,
       depth_test="always")

# segeta vermella3
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux3[1], p1_flux3[2], p1_flux3[3]+2,
       texts = "Flux retrovertical 3", color = "green", cex = 0.6,
       depth_test="always")

```

```

# segeta vermella4
arrow3d(
  p0 = p0_flux4,
  p1 = p1_flux4,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(p1_flux4[1], p1_flux4[2], p1_flux4[3]+2,
       texts = "Flux retrovertical 4", color = "green", cex = 0.6,
       depth_test="always")

# 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]+3, p1_flux0[2], p1_flux0[3]+2,
       texts = "Flux vertical1", color = "blue", cex = 0.6,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = coords3Dn_diag_corr1[3,] ,
  p1 = coords3Dn_diag_corr1[2,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barrlen = 0.02
)
text3d(coords3Dn_diag_corr1[2,][1]+2, coords3Dn_diag_corr1[2,][2], coords3Dn_diag_c
orr1[2,][3]-3,
       texts = "Flux vertical2", color = "blue", cex = 0.6,
       depth_test="always")

# segeta blava3
arrow3d(
  p0 = coords3Dn_diag_corr1[3,] ,
  p1 = coords3Dn_diag_corr1[9,] ,
  col = "blue",

```

```

type = "rotation",
lwd = 2,
width = 0.3,
barblen = 0.02
)
text3d(coords3Dn_diag_corr1[9,][1]-11, coords3Dn_diag_corr1[9,][2]+10, coords3Dn_di
ag_corr1[9,][3]-10,
       texts = "Flux vertical3", color = "blue", cex = 0.6,
       depth_test="always")

# segeta blava4
arrow3d(
  p0 = coords3Dn_diag_corr1[4,] ,
  p1 = coords3Dn_diag_corr1[8,] ,
  col = "blue",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(coords3Dn_diag_corr1[8,][1]-9, coords3Dn_diag_corr1[8,][2]+15, coords3Dn_dia
g_corr1[8,][3]-10,
       texts = "Flux vertical4", color = "blue", cex = 0.6,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[8, ]),
            col = "purple", lwd = 1.5)

# segeta Lila
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",
  type = "rotation",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux[1]-5, p1_flux[2], p1_flux[3]-3,
       texts = paste0("Flux local versus causal | (Angle; Distancia) = (",
                     round(int
ersecció_fluxos8$angle_graus,0), ";",
                     round(intercecció_fluxos8$distancia_minima,1),
                     ")"),
       color = "purple4", cex = 0.6,
       depth_test="always")

```

```

#p0 <- colMeans(coords3Dn_diag_corr1)
# el flux causal global
#res_global <- mitjana_vectorial_global(taula_general_fluxos)
#v_geom111 <- res_global$v_sum # components reals
#h222 <- sqrt(sum(v_geom111^2))
#v_global <- v_geom111 / h222 # direcció normalitzada
# EL CODI CORRECTE ÉS AQUEST p1 <- p0 + v_global * h222 (h222 és el punt final del
# flux vertical (direcció + mòdul real)) PERÒ HOM MULTIPLICA PER max(taula_general_fluxos$modul)
# A FI DE NO FER ELS EIXOS DEL PLOT MASSA LLARGS
#p1 <- p0 + v_global * max(taula_general_fluxos$modul) # direcció

#arrow3d(
# p0 = p0,
# p1 = p1,
# type = "rotation",
# col = "orange",
# lwd = 3,
# width = 0.3,
# barrlen = 0.02
#)

#text3d(p1,
#        texts = "Flux causal vertical",
#        col = "black", cex = 0.75,
#        depth_test="always")

#aspect3d(1,1,1)

# el flux causal vertical: aplica la funció que implementa la mitjana vectorial ponderada dels col·lididors fulla (retorna v_sum com a vector causal amb direcció i mòdul, v_global com a direcció normalitzada). AQUÍ HOM CONSTREIX EL FLUX CAUSAL
res_vertical1 <- mitjana_vectorial_global(taula_general_fluxos)

## con causal
p0 <- colMeans(coords3Dn_diag_corr1) # mitjana de cada eix X,Y,Z, punt d'origen (hom defineix el centre geomètric del sistema a dins de l'espai 3D com a centre de masses, on neix el con causal que és al centre del model, no pas a un node concret. És tant el vèrtex del con com l'origen de la segeta causal)

v_geom1 <- res_vertical1$v_sum #suma ponderada de tots els fluxos fulla (vector geomètric del flux holòstic): conté la direcció emergent i la intenistat causal per mitjà dels pesos. ÉS EL VECTOR FÍSIC DEL FLUX CAUSAL, NO SOLS UNA DIRECCIÓ

```

```

v2 <- sqrt(sum(v_geom1^2))/8# MÒDUL!!!! Llargària del con (la mateixa que la del seu eix, el flux causal holístic): interpretable com a intensitat global del flux i a lçada real del con causal. VALOR QUE DEFINEIX L'EIX DEL CON

v_vertical_norm1 <- res_vertical1$v_global # vector holístic normalitzat: direcció normalitzada del flux holístic

# EL CODI CORRECTE ÉS AQUEST
p1 <- p0 + v_vertical_norm1 * v2 #(v2 és el punt final del flux vertical (direcció + mòdul real)) #PERÒ HOM MULTIPLICA PER max(taula_general_fluxos_fulla111$modul) A FI DE NO FER ELS EIXOS DEL PLOT MASSA LLARGS
#p1 <- p0 + v_vertical_norm1 * max(taula_general_fluxos$modul) # EL CORRECTE ÉS v2 NO PAS max(taula_general_fluxos$modul)... intensitat del flux (per mitjà del mòdul) : n'és l'eix

# paràmetres del con
theta_con111 <- res_vertical1$angle_mean * pi / 180 # en radians
# EL RADI CORRECTE :
rad111 <- v2 * tan(theta_con111) # radi base del con
#rad111 <- max(taula_general_fluxos$modul) * tan(theta_con111) # radi base

## ANGLES DE LES 4 FULLES RESPECTE AL VECTOR HOLISTIC
#hom extreu els vectors dels 8 col·lididors fulla: La matriu de vectors 3D dels col·lididors fulla, són les seves direccions causals locals
fulles111 <- taula_general_fluxos %>%
  filter(flux != "flux_objectiu") %>%
  select(u_bar_X, u_bar_Y, u_bar_Z) %>%
  as.matrix()

fulles_norm111 <- fulles111 / sqrt(rowSums(fulles111^2)) #hom nomralitza cada vector fulla (obligat per tal de calcular angles, els angles sols tenen trellat entre vectors unitat)

#hom calcula l'angle en graus entre cada col·lididor fulla(direcció local) i el flux causal global(v_vertical_norm1): són els angles individuals que defineixen la dis persió angular
angles111 <- acos(fulles_norm111 %% v_vertical_norm1) * 180/pi

#theta_con <- max(angles111)# amplada del con = màxima variació angular
#theta_con# aquest ÉS l'angle correcte del con causal
arrow3d(
  p0 = p0,
  p1 = p1,

```

```

type = "rotation", #segeta orientada segons vector
col = "red",
lwd = 3,
width = 0.3,
barblen = 0.02
)

text3d(p1+5,
  texts = paste0("Flux causal vertical | (Angle; Modul) = (",round(res_vertical
1$angle_mean,0), ";", round(v2,0), ")"),
  col = "red",
  cex = 0.75,
  depth_test="always")

#CON CAUSAL
# hom gira el con perquè apunti com el flux objectiu
z_axis <- c(0, 0, 1) # vector unitari Z
v_obj111 <- v_vertical_norm1 # <<< el vector direcció normalitzat

# Producte vectorial per trobar eix de rotació
rot_axis111 <- pracma::cross(z_axis, v_obj111)
rot_angle111 <- acos(sum(z_axis * v_obj111))

# Matriu de rotació (Rodrigues)
if (sqrt(sum(rot_axis111^2)) > 0) {
  k_vec <- rot_axis111 / sqrt(sum(rot_axis111^2))

  K_mat <- matrix(c(0, -k_vec[3], k_vec[2],
                    k_vec[3], 0, -k_vec[1],
                    -k_vec[2], k_vec[1], 0),
                  nrow = 3,
                  byrow = TRUE)

  R <- diag(3) * cos(rot_angle111) +
    sin(rot_angle111) * K_mat +
    (1 - cos(rot_angle111)) * (k_vec %*% t(k_vec))
} else {

```

```

R <- diag(3)
}

# hom renderitza el con
n_cercle <- 20 # nombre de punts que aproximen el cercle
theta <- seq(0, 2 * pi, length.out = n_cercle)
x <- rad111 * cos(theta)
y <- rad111 * sin(theta)
# EL CORRECTE z <- rep(max(h2, n_cercle)
z <- rep(v2, n_cercle)

# con alineat amb l'eix Z i vèrtex a l'origen
cone_base <- rbind(x, y, z) # 3 x n_cercle
#vertex <- matrix(c(0,0,0), nrow = 3, ncol = 1) # 3x1
# Apliquem la rotació correcta
cone_rot <- R %*% cone_base

#origin <- c(B["X"], B["Y"], A["Z"])
origin <- colMeans(coords3Dn_diag_corr1) # mitjana de cada eix X,Y,Z
cone_pts <- sweep(t(cone_rot), 2, origin, "+")

triangles <- c()
for (i in 1:(n_cercle - 1)) {
  triangles <- rbind(triangles, cbind(i, i+1, n_cercle+1))
}

cone_mesh <- tmesh3d(
  vertices = t(rbind(cone_pts, origin)),
  indices = t(triangles),
  homogeneous = FALSE
)

shade3d(cone_mesh, alpha = 0.18, col = "red")
#spheres3d(0,0,0, radius = h*0.03) # punt del vèrtex a fi d'orientar-se
text3d(origin, texts = "Con causal vertical", col = "red", cex = 0.6)

aspect3d(1,1,1)

## =====
## ESFERA TANGENT A LA BASE DEL CON (CON SAGRAT, NO ES TOCA)
## =====

par3d(ignoreExtent = TRUE)

```

```

par3d(scale = c(1,1,1))

# --- centre sagrat (centre de masses) ---
centre <- colMeans(coords3Dn_diag_corr1)

# --- paràmetres sagrats del con (JA DEFINITS ABANS) ---
L <- v2
theta_con <- res_vertical1$angle_mean * pi/180

# --- radi CORRECTE de l'esfera: base del con en contacte ---
r <- L / cos(theta_con)

# --- malla esfèrica ---
nu <- 20
nv <- 20

u <- seq(0, pi, length.out = nu)
v <- seq(0, 2*pi, length.out = nv)

Xs <- centre[1] + r * outer(sin(u), cos(v))
Ys <- centre[2] + r * outer(sin(u), sin(v))
Zs <- centre[3] + r * outer(cos(u), rep(1, nv))

# --- render de l'esfera ---
surface3d(
  Xs, Ys, Zs,
  color = "gray77",
  alpha = 0.20,
  lit = FALSE
)

# --- centre (orientació visual) ---
points3d(
  centre[1], centre[2], centre[3],
  col = "orangered4",
  size = 6
)

## =====
## CASQUET ESFÈRIC ADMISSIBLE (OMBRA DEL CON)
## =====

# direcció unitària del con
v_dir <- as.numeric(v_vertical_norm1)

# resolució del casquet
nu_cap <- 20

```

```

nv_cap <- 20

# angle polar limitat al con
u_cap <- seq(0, theta_con, length.out = nu_cap)
v_cap <- seq(0, 2*pi, length.out = nv_cap)

# casquet en coordenades esfèriques (eix Z)
Xs0 <- outer(sin(u_cap), cos(v_cap))
Ys0 <- outer(sin(u_cap), sin(v_cap))
Zs0 <- outer(cos(u_cap), rep(1, nv_cap))

# rotació Z -> v_h_norm (Rodrigues, reutilitzant criteri)
z_axis <- c(0,0,1)
rot_axis <- pracma::cross(z_axis, v_dir)
rot_angle <- acos(sum(z_axis * v_dir))

if (sqrt(sum(rot_axis^2)) > 0) {
  k_vec <- rot_axis / sqrt(sum(rot_axis^2))
  k_mat <- matrix(c(0,-k_vec[3],k_vec[2],
                    k_vec[3],0,-k_vec[1],
                    -k_vec[2],k_vec[1],0), 3,3, byrow=TRUE)
  Rcap <- diag(3)*cos(rot_angle) +
    sin(rot_angle)*k_mat +
    (1-cos(rot_angle))*(k_vec %*% t(k_vec))
} else {
  Rcap <- diag(3)
}

# aplicar rotació, escala i trasllat
cap_rot <- Rcap %*% rbind(
  as.vector(Xs0),
  as.vector(Ys0),
  as.vector(Zs0)
)

Xs_cap <- matrix(cap_rot[1,], nu_cap, nv_cap) * r + centre[1]
Ys_cap <- matrix(cap_rot[2,], nu_cap, nv_cap) * r + centre[2]
Zs_cap <- matrix(cap_rot[3,], nu_cap, nv_cap) * r + centre[3]

# render del casquet
surface3d(
  Xs_cap, Ys_cap, Zs_cap,
  color = "red",
  alpha = 0.35,
  lit = FALSE
)

```

```

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))

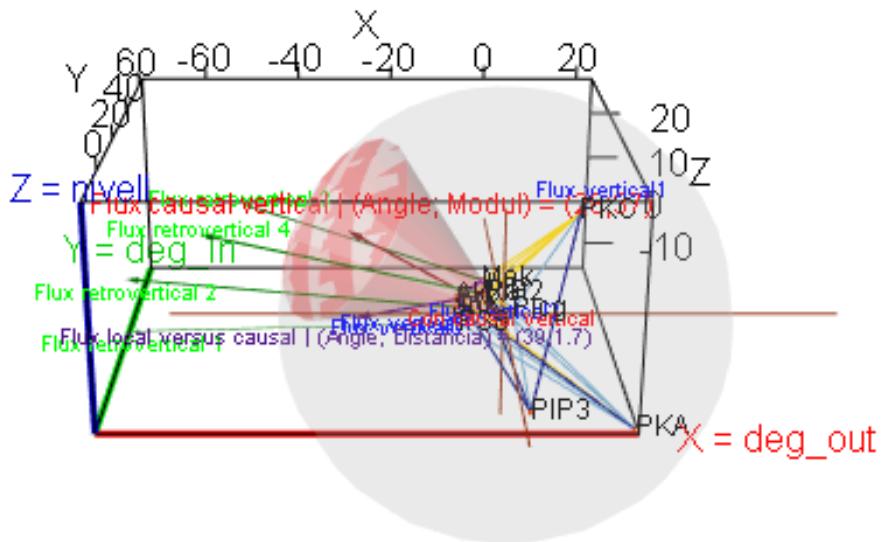
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

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
```



## 5.8 Càlcul de l'angle de flux causal holòstic (ZYX) del model topoempíric: flux\_fulla\_corregit

taula\_general\_fluxos\_fulla4 amb coords3Dn\_diag\_corr1

Taula 20

#TALLAFOC1

```
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("PI
P3", "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_collididor_Z_fisic(collididor3n_PIP3_Akt_PK, coords3Dn_diag_corr1)

# Fluxos fulles retroverticals (2 col·lididors fulla retro-verticals)
flux2_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PK, coords3Dn_diag_corr1)
flux3_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PK, coords3Dn_diag_corr1)

# Flux fulla laterovertical (1 col·lididor fulla latero-vertical)
flux4_coll_laterover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1)

fluxos4 <- 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")
  )
)

```

```

## ---- Diferències angulars detallades ----
taula_general_fluxos_fulla4 <- taula_fluxos_general(fluxos4)
resultat4 <- mitjana_vectorial_global(taula_general_fluxos_fulla4)

v_obj4 <- as.numeric(t(taula_general_fluxos_fulla4[taula_general_fluxos_fulla4$flux
== "flux_objectiu", c("u_bar_X", "u_bar_Y", "u_bar_Z")])))

# normalitza vectors
ng4 <- resultat4$v_global
no4 <- v_obj4 / sqrt(sum(v_obj4^2))

# angles per eix
ang_X4 <- acos(ng4[1] * no4[1]) * 180 / pi
ang_Y4 <- acos(ng4[2] * no4[2]) * 180 / pi
ang_Z4 <- acos(ng4[3] * no4[3]) * 180 / pi

# angle total 3D
ang_total4 <- acos(sum(ng4 * no4)) * 180 / pi

resultat_angles4 <- list(
  angle_X = ang_X4,
  angle_Y = ang_Y4,
  angle_Z = ang_Z4,
  angle_total_3D = ang_total4,
  angles_fulla = as.numeric(resultat4$angles_fulla),
  angle_max = resultat4$angle_max,
  angle_mean = resultat4$angle_mean
)

resultat4

## $vectors_utilitzats
## [1] "flux1_coll_ver"      "flux2_coll_retrover"  "flux3_coll_retrover"
## [4] "flux4_coll_laterover"
##
## $v_sum
##   u_bar_X   u_bar_Y   u_bar_Z
## -181.89736  83.09035  36.35038
##
## $v_global
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.8949283  0.4088015  0.1788425
##
## $angle_vertical
##   u_bar_Z
## 79.69765

```

```

## 
## $angle_objectiu
## [1] 76.05867
##
## $diferencia_angles
## u_bar_Z
## 3.638985
##
## $angles_fulla
## [1] 24.86241 20.24756 14.20798 19.14151
##
## $angle_mean
## [1] 19.87493
##
## $angle_max
## [1] 24.86241

```

Per tal de crear nodes col·lididors amb 2 o més de parents

```

crea_col·lididor_k <- function(graph, centre, parents) {
  # <centre> és el node col·lididor B (a->B<-c)
  # <parents> és el vector de nodes (a i c) que van cap a B
  stopifnot(length(parents) >= 2) #guaita perquè hi hagin mínim dos parents

  g_col <- make_empty_graph(directed = TRUE) %>% add_vertices(length(parents) + 1,
  name = c(centre, parents))

  for (p in parents) {
    g_col <- add_edges(g_col, c(p, centre)) #hom afegeix una aresta per a casacun dels parents amb el centre centre
  }
  return(g_col) #un DAG amb tantes d; a restes com length(parents)
}

theta_flux_col·lididor_k <- function(g_col, coords3D) {
  # Llista d'arestes entrants
  E <- as_edgelist(g_col)

  # node col·lididor (destí comú)
  centres <- E[, 2]
  stopifnot(length(unique(centres)) == 1)
  B <- unique(centres)

  parents <- E[, 1]
  k <- length(parents)
  stopifnot(k >= 2)

  # coordenades del node central

```

```

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

# vectors entrants v_i = B - A_i (sense normalitzar!)
v_list <- lapply(parents, function(P) {
  pA <- as.numeric(coords3D[P, c("X", "Y", "Z")])
  pB - pA
})

# resultant
r_in <- Reduce("+", v_list)

# mòdul geomètric del resultant
mods <- sapply(v_list, function(v) sqrt(sum(v^2)))
len_r <- sum(mods)

if (len_r == 0) {
  return(list(
    r_in = r_in,
    len_r = 0,
    flux_z = 0,
    theta_z = NA_real_,
    signe = 0
  ))
}

flux_z <- r_in[3]
theta_z <- acos(abs(flux_z) / len_r) * 180 / pi
signe <- sign(flux_z)

return(list(
  r_in = r_in,
  # vector que ja porta TOT el pes dels mòduls
  len_r = len_r,
  # mòdul del resultant
  flux_z = flux_z,
  theta_z = theta_z,
  signe = signe
))
}

```

### 5.8.1 FLUX CAUSAL holístic del model geomètric topoempíric general: FLUX CAUSAL ÚNIC EN L'ESFERA DE DIRECCIONS

`taula_fluxos_general`

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 va  
lors 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 <- 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, coor  
ds3Dn_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, coor  
ds3Dn_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_collididor_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")
colididor3n_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_corr1_MOD <- 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]
# _____

```

### Taula 21

```

res_holistic4 <- mitjana_vectorial_global(taula_general_fluxos_fulla4)

v_holos4 <- res_holistic4$v_sum
p04 <- mean(coords3Dn_diag_corr1)

v_local4 <- as.numeric(res_flux$u_bar)
# _____

```

```

q04 <- p0_flux

(intersecció_fluxos4 <- angle_interseccio_fluxos(p04, v_holos4, q04, v_local4))

## $angle_graus
## [1] 29.25813
##
## $distancia_minima
## [1] 0.8678162
##
## $punt_holistic
##   u_bar_X   u_bar_Y   u_bar_Z
## -7.268485  3.320229  1.452534
##
## $punt_local
##      X          Y          Z
## -7.486466  2.632215  1.934433
##
## $vector_PQ
##      X          Y          Z
## -0.2179808 -0.6880140  0.4818985

```

### *FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn\_diag\_corr1 (SENSE ARESTA Erk->Akt)*

El flux local —tot i atravesar el con— cau a fora del con causal i, per tant, no sembla que sigui a dins del casquet de l'esfera de direccions. Un altre fet a ressaltar és que l'angle del con ara és més petit: abans amb el flux causal vertical hi ha un angle de 19.58 i ara l'angle del con és de 18.37 (hom recorda que l'angle del con és definit com la mitjana angular dels col·lididors fulla); sembla que el càlcul del flux causal per mitjà del subgraf retro lateral dóna un con més òptim en termes de model: *angle mitjà més petit i mòdul més llarg.*

El gràfic mostra quin flux causal hom obté a partir del subgraf retro lateral (encara hom no força l'aresta absent com a present Erk->Akt amb el recàlcul consegüent de les coordenades topoempíriques) ##### Figura 22 <Plot coords3Dn\_diag\_corr1 amb 4 vectors directors de flux SENSE ARESTA Erk->Akt>

```

open3d()

## wgl
## 55

plot3d(coords3Dn_diag_corr1, 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_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"],
       coords3Dn_diag_corr1[, "Y"],
       coords3Dn_diag_corr1[, "Z"],
       texts = rownames(coords3Dn_diag_corr1),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),col = "blue",
#            lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[8, ]),col = "blue",
#            lwd = 1.5)

# segeta daurada1
arrow3d(
  p0 = p0_flux1,
  p1 = p1_flux1,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux1#[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical1 ", color = "green4",cex = 0.6,
       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 retrovertical2 ", color = "green4",cex = 0.6,
       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 = 0.6,
       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 = 0.6,
       depth_test="always")

```

```

#edges1 <- E(g_bosc)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr1[7, ], coords3Dn_diag_corr1[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr1[1, ], coords3Dn_diag_corr1[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],
       texts = paste0("Flux local versus causal | (Angle; Distancia) = (",
                     round(int
ersecció_fluxos4$angle_graus,0), ";", round(int
ersecció_fluxos4$distancia_minima,1),
                     ")"),
       color = "purple4",
       cex = 0.6,
       depth_test="always")
}

## CON CAUSAL HOLÍSTIC 3D

# 0) Centre geomètric del model
centre <- colMeans(coords3Dn_diag_corr1)

# 1) Dimensions reals del model (alçada del con)
#distancies <- sqrt(rowSums((
#  coords3Dn_diag_corr1 -
#  matrix(centre, nrow(coords3Dn_diag_corr1), 3, byrow =
#         TRUE))^2))

v_axis <- resultat4$v_sum
h4 <- sqrt(sum(v_axis^2))/4
v_axis_n <- resultat4$v_global

# el flux causal global
res_holistic4 <- mitjana_vectorial_global(taula_general_fluxos_fulla4)

v_sum4 <- res_holistic4$v_sum # components reals
v_holistic4 <- res_holistic4$v_global # direcció normalitzada
pfinal <- centre + v_axis_n * h4

```

```

# paràmetres del con
theta_con4 <- res_holistic4$angle_mean * pi / 180 # en radians
# EL RADI CORRECTE :
rad4 <- h4 * tan(theta_con4) # radi base del con
#rad111 <- max(taula_general_fluxos$modul) * tan(theta_con111) # radi base

## ANGLES DE LES 4 FULLES RESPECTE AL VECTOR HOLISTIC
#fulles4 <- taula_general_fluxos_fulla4 %>%
# filter(flux != "flux_objectiu") %>%
# select(u_bar_X, u_bar_Y, u_bar_Z) %>%
# as.matrix()
#fulles_norm4 <- fulles4 / sqrt(rowSums(fulles4^2))
#angles4 <- acos(fulles_norm4 %*% v_axis_n) * 180/pi

# hom escala la segeta perquè sigui comparable amb el model

#p0 <- coords3Dn_diag_corr1["Erk",]
#p1 <- p0 + v_holistic4 * escala # direcció
# 2) Angle causal en radians i radi del con

#theta_con_gra <- mean(angles4)
#theta_con_rad <- theta_con_gra*pi/180

# aquest és el correcte rad <- h * tan(theta_con_rad) però hom posa un h adaptat per
# rquè l'escala dels eixos no siguin desmesuradament grans

#rad_correcte <- h * tan(theta_con_rad) #radi=(alçada del con).tan(angle d'obertura)
#h_plot <- max(taula_general_fluxos_fulla4$modul)
#scale_factor <- h_plot / h
#rad_eixos <- scale_factor * rad_correcte

arrow3d(
  p0 = centre,
  p1 = pfinal,
  type = "rotation",
  col = "red",
  lwd = 3,

```

```

width = 0.3,
barblen = 0.02
)

text3d(p1,
  texts = paste0("Flux causal holistic (4) | (Angle; Modul) = (",
    round(res_holistic4$angle_mean, 0), ";", round(h4, 0), ")"),
  col = "red",
  cex = 0.75,
  depth_test="always")

# 4) Orientació: hom gira el con perquè apunti com el flux causal

z_axis <- c(0, 0, 1)
rot_angle4 <- v_axis_n

rot_axis4 <- pracma::cross(z_axis, rot_angle4)
rot_angle4 <- acos(sum(z_axis * rot_axis4))

if (sqrt(sum(rot_axis4^2)) > 0) {
  k4 <- rot_axis4 / sqrt(sum(rot_axis4^2))

  K <- matrix(
    c(0, -k4[3], k4[2],
      k4[3], 0, -k4[1],
      -k4[2], k4[1], 0),
    3,
    byrow = TRUE)

  R <- diag(3)*cos(rot_angle4) + sin(rot_angle4) * K +
    (1-cos(rot_angle4)) * (k4 %*
% t(k4))
} else {
  R <- diag(3)
}

# 6) Renderitzat del con
# Base d'un con alineat amb l'eix Z
n_cercle <- 20
theta <- seq(0, 2 * pi, length.out = n_cercle)
x4 <- rad4 * cos(theta)
y4 <- rad4 * sin(theta)
z4 <- rep(h4, n_cercle)

```

```

cone_base <- rbind(x4, y4, z4)
cone_pts <- t(R %*% cone_base)

# 5) El con comença al centre geomètric
origin <- centre
cone_pts <- sweep(cone_pts, 2, origin, "+")

triangles <- c()
for (i in 1:(n_cercle - 1)){
  triangles <- rbind(triangles, cbind(i, i+1, n_cercle+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 (4)",
       col = "red",
       cex = 0.6)

aspect3d(1,1,1)

## =====
## ESFERA TANGENT A LA BASE DEL CON (CON SAGRAT, NO ES TOCA)
## =====

par3d(ignoreExtent = TRUE)
par3d(scale = c(1,1,1))

# --- centre sagrat (centre de masses) ---
centre <- colMeans(coords3Dn_diag_corr1)

# --- paràmetres sagrats del con (JA DEFINITS ABANS) ---
L <- h4

```

```

theta_con <- res_holistic4$angle_mean * pi/180

# --- radi CORRECTE de l'esfera: base del con en contacte ---
r <- L / cos(theta_con)

# --- malla esfèrica ---
nu <- 20
nv <- 20

u <- seq(0, pi, length.out = nu)
v <- seq(0, 2*pi, length.out = nv)

Xs <- centre[1] + r * outer(sin(u), cos(v))
Ys <- centre[2] + r * outer(sin(u), sin(v))
Zs <- centre[3] + r * outer(cos(u), rep(1, nv))

# --- render de l'esfera ---
surface3d(
  Xs, Ys, Zs,
  color = "gray77",
  alpha = 0.20,
  lit = FALSE
)

# --- centre (orientació visual) ---
points3d(
  centre[1], centre[2], centre[3],
  col = "orangered4",
  size = 6
)

## =====
## CASQUET ESFÈRIC ADMISSIBLE (OMBRA DEL CON)
## =====

# direcció unitària del con
v_dir <- as.numeric(res_holistic4$v_global)

# resolució del casquet
nu_cap <- 20
nv_cap <- 20

# angle polar limitat al con
u_cap <- seq(0, theta_con, length.out = nu_cap)
v_cap <- seq(0, 2*pi, length.out = nv_cap)

# casquet en coordenades esfèriques (eix Z)

```

```

Xs0 <- outer(sin(u_cap), cos(v_cap))
Ys0 <- outer(sin(u_cap), sin(v_cap))
Zs0 <- outer(cos(u_cap), rep(1, nv_cap))

# rotació Z -> v_h_norm (Rodrigues, reutilitzant criteri)
z_axis <- c(0,0,1)
rot_axis <- pracma::cross(z_axis, v_dir)
rot_angle <- acos(sum(z_axis * v_dir))

if (sqrt(sum(rot_axis^2)) > 0) {
  k_vec <- rot_axis / sqrt(sum(rot_axis^2))
  k_mat <- matrix(c(0,-k_vec[3],k_vec[2],
                    k_vec[3],0,-k_vec[1],
                    -k_vec[2],k_vec[1],0), 3,3, byrow=TRUE)
  Rcap <- diag(3)*cos(rot_angle) +
    sin(rot_angle)*k_mat +
    (1-cos(rot_angle))*(k_vec %*% t(k_vec))
} else {
  Rcap <- diag(3)
}

# aplicar rotació, escala i trasllat
cap_rot <- Rcap %*% rbind(
  as.vector(Xs0),
  as.vector(Ys0),
  as.vector(Zs0)
)

Xs_cap <- matrix(cap_rot[1,], nu_cap, nv_cap) * r + centre[1]
Ys_cap <- matrix(cap_rot[2,], nu_cap, nv_cap) * r + centre[2]
Zs_cap <- matrix(cap_rot[3,], nu_cap, nv_cap) * r + centre[3]

# render del casquet
surface3d(
  Xs_cap, Ys_cap, Zs_cap,
  color = "red",
  alpha = 0.35,
  lit = FALSE
)

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))

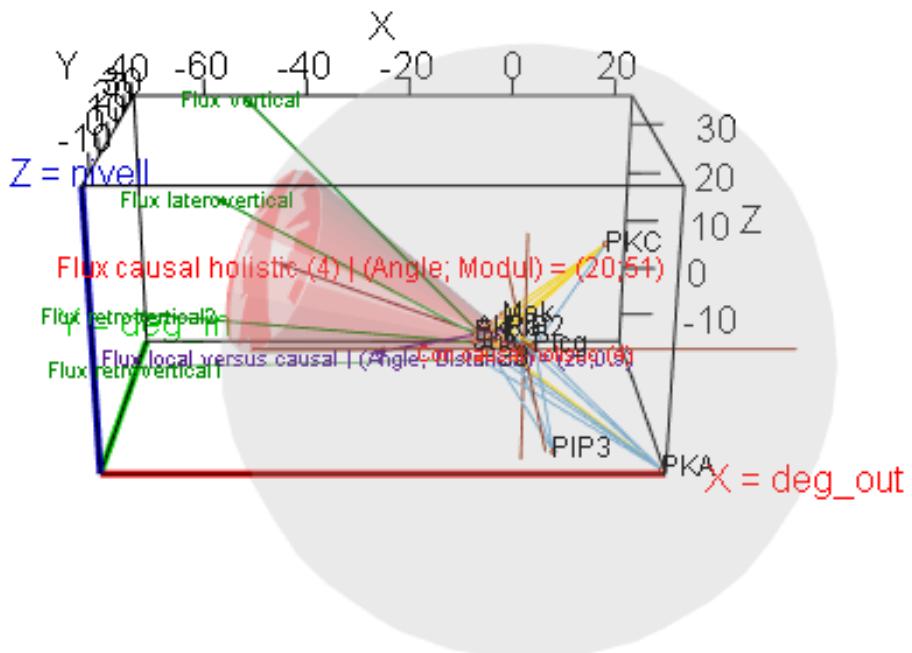
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

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

```



### *Hom força la inclusió de l'aresta absent Erk -> Akt*

El cas d'estudi és l'aresta absent Erk -> Akt, si hom força l'aresta a dins del model causal geomètric, causa la reducció de nodes col·lididors fulla de 4 a 3.

A un col·lididor k-éssim, l'efecte geomètric d'una aresta afegida depèn tant del seu mòdul (força) com de la seva direcció angular. Així, fins i tot una correlació feble pot alterar en certa mesura el vector resultant de la mitjana vectorial si la seva direcció és molt diferent dels restants.

Cal reprendre el fil a la 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), aquí hom força una aresta absent a present, cosa que implica tornar a refer les coordenades topoempíriques finals.

El DAG\_combinatori no inclou les dades empíriques en qüestió, el model causal geomètric sí, d'aquí que, per tal de plantejar els mateixos tests locals, cal treballar a dins del mateix terreny de joc: cal forçar l'aresta absent a dins del model o treure'n-la (segons el cas) i calcular el con causal modificant i comparar-ne el radi amb el con causal holístic. Si el radi és més petit, l'aresta absent és probable en termes de causalitat; altrament, si el radi del con modificant és més gran, l'aresta absent no sembla que sigui probable en el marc d'un test geomètric que força una única aresta (forçar o bé la inclusió d'una aresta absent o bé l'exclusió d'una aresta present).

Ara el subgraf retrolateral ja és connex: dos nodes fulla i un collididor fulla, en total 3 fulles. Que l'aresta absent proposada per l'article d'standard gold com a present sigui just aquesta és molt interessant perquè és just la que uneix els dos components del subgraf retrolateral. ##### Figura 23

```
set.seed(777)
par(mar=c(0,0,1,0))
g_retrolateral3 <- add_edges(g_retrolateral1, c("Erk", "Akt"))

arrels <- V(g_retrolateral3)$name[degree(g_retrolateral3, mode="in") == 0]
#nodes fulla (sense fills)
fulles <- V(g_retrolateral3)$name[degree(g_retrolateral3, mode="out") == 0]
#nodes col·lididors (>= 2 pares)
collididores <- V(g_retrolateral3)$name[degree(g_retrolateral3, mode="in") >= 2]

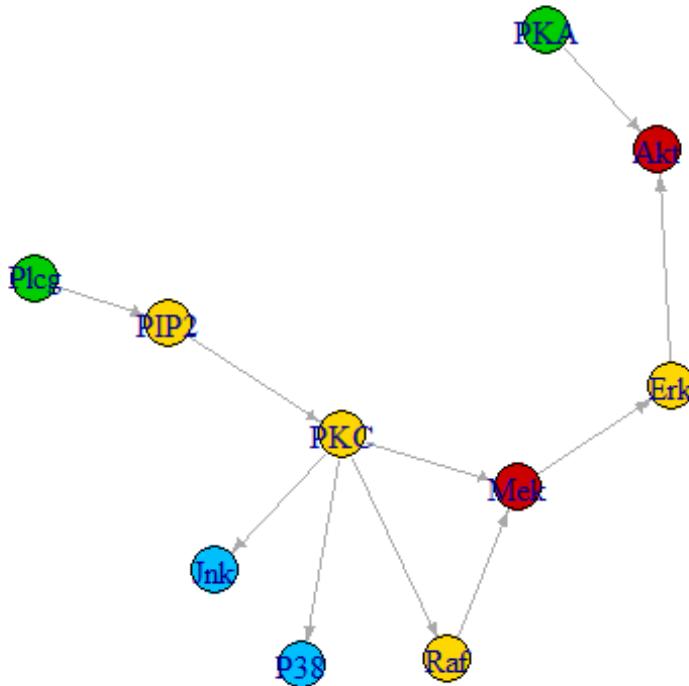
noms_nodes <- V(g_retrolateral3)$name

V(g_retrolateral3)$role <- ifelse(!is.na(match(noms_nodes,collididores )), "col·lididor",
                                     ifelse(!is.na(match(noms_nodes, arrels)), "arrel",
                                           ifelse(!is.na(match(noms_nodes,fulles )), "fulla", "mitjancer"
                                         )))
                                         )))

V(g_retrolateral3)$color <- ifelse(V(g_retrolateral3)$role == "col·lididor", "red3"
,
                                     ifelse(V(g_retrolateral3)$role == "arrel" , "green3",
                                           ifelse(V(g_retrolateral3)$role == "fulla", "deepskyblue", "gold"
                                         )))

plot(g_retrolateral3,
      layout=layout_with_gem(g_retrolateral3),
      edge.arrow.size=0.4,
      vertex.size=15,
      main ="subgraf retrolateral connex dirigit")
```

## subgraf retrolateral connex dirigit



Una nova cadena (Mek -> Erk-> Akt) i un nou col·lididor (Erk -> Akt <- PKA), aquest darrer canvia del tot el paper d'Akt dins del subgraf, la qual cosa no altera l'estructura retrolateral basada en PKC PERÒ transforma Erk en medial i transmforma Akt en un nou punt de confluència causal. Tots dos canvis transformen qualitativament la geometria del flux causal holístic.

Hi han els mateixos nodes que a g\_retrolateral1 i sols una única aresta més, però el sorgiment de nous col·lididors sembla que acreix la probabilitat de transició de non-connexitat a connexitat del subgraf tal com en aquest cas en efecte s'ha esdevingut. El col·lididor és una estructura de convergència (pot reduir el nombre de fulles, cosa que tanca el con causal i reorienta el flux holístic).

#### Taula 22

triades contigues(g_retrolateral3)				
## # A tibble: 15 × 5				
n1	n2	n3	tipus	edges
1	Mek	Erk	Akt	Cadena
2	Mek	PIP2	PKC	Cadena
3	Mek	PKC	Erk	Cadena
4	Mek	Raf	Erk	Cadena
5	PIP2	PKC	Jnk	Cadena
6	PIP2	PKC	P38	Cadena
7	PIP2	PKC	Plcg	Cadena
8	PIP2	PKC	Raf	Cadena
9	Erk	PKA	Akt	Col·lididor
10	Mek	PKC	Raf	Col·lididor
				Erk->Akt, PKA->Akt
				PKC->Mek, PKC->Raf, Raf->Mek

```

## 11 Mek   PKC   Jnk   Forca      PKC->Mek, PKC->Jnk
## 12 Mek   PKC   P38   Forca      PKC->Mek, PKC->P38
## 13 PKC   Jnk   P38   Forca      PKC->Jnk, PKC->P38
## 14 PKC   Raf    Jnk   Forca      PKC->Raf, PKC->Jnk
## 15 PKC   Raf    P38   Forca      PKC->Raf, PKC->P38

```

*Taula 23 <matriu d'adjacència (dirigida)>*

```

# matriu d'adjacència (dirigida) seguint l'ordre de coords3D
nodes <- rownames(coords3Dn_diag)
A1 <- as.matrix(as_adjacency_matrix(g_bosc, type = "both", attr = NULL))
A1["Erk","Akt"] <- 1
A1["Akt","Erk"] <- 0
# 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_filt1 <- R * A1 # zeros on no hi ha aresta

R_filt_df1 <- as.data.frame(as.table(R_filt1))
colnames(R_filt_df1) <- c("des", "fins", "corr")
R_filt_df1 <- R_filt_df1[R_filt_df1$corr != 0,]
R_filt_df1

##      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
## 96     Erk  Akt  0.902993735
## 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 #### Taula 24 Akt>

```

g_bosc1 <- g_bosc  # recuperar graf original
g_bosc1 <- add_edges(g_bosc1, c("Erk", "Akt"))
edges_mod <- as_edgelist(g_bosc1, names = TRUE)

```

```

# funció d'utilitat per a longituds
edge_lengths <- function(coords3Dn_diag, edges_mod){
  des <- edges_mod[,1]
  fins <- edges_mod[,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 DIRECCIÓ I MAGNITUD), no
    o pas una coordenada
    dy = vec[,2],
    dz = vec[,3],
    length = lon,
    row.names = NULL
  )
}

(edges_df_mod <- edge_lengths(coords3Dn_diag, edges_mod))

##      des fins        dx        dy        dz    length
## 1    Mek  Erk -5.5735250 -11.7571755  0.6879384 13.029529
## 2   PIP2  PKC 17.0131484 -1.8583116 16.6791133 23.897560
## 3   PIP3  Akt -12.5433772 17.6154611 15.5274458 26.622215
## 4   PIP3  PIP2 -7.3917630 18.1947023 15.6868359 25.134880
## 5   PIP3  Plcg -2.8864444  9.6976674 15.8622414 18.814543
## 6    PKA  Akt -31.0694241 20.5702235 17.5538571 41.189575
## 7    PKA  Erk -31.3408343 19.5748302 17.1158161 40.723126
## 8    PKA  Jnk -31.4729774 20.0092071 15.4491832 40.368229
## 9    PKA  Mek -25.7673093 31.3320057 16.4278777 43.766699
## 10   PKA  P38 -30.7121359 18.4218761 12.7995533 38.031952
## 11   PKA  Raf -25.9688852 18.8161605 18.4625857 37.004026
## 12   PKC  Jnk -22.5683160  0.7180540 -18.9431773 29.473521
## 13   PKC  Mek -16.8626478 12.0408526 -17.9644828 27.423597
## 14   PKC  P38 -21.8074745 -0.8692771 -21.5928073 30.701318
## 15   PKC  Raf -17.0642238 -0.4749927 -15.9297749 23.348899
## 16  Plcg  PIP2 -4.5053186  8.4970349 -0.1754054  9.619161
## 17  Plcg  PKC 12.5078298  6.6387233 16.5037079 21.746053
## 18    Raf  Mek  0.2015760 12.5158453 -2.0347079 12.681761
## 19    Erk  Akt  0.2714102  0.9953933  0.4380410  1.120871

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

```

```

df_fos1 <- R_filt_df1 %>% left_join(edges_df_mod, by=c("des", "fins"))
df_fos1$llarg_nova <- (df_fos1$corr*df_fos1$length) + df_fos1$length
df_fos1$prop <- df_fos1$llarg_nova/ df_fos1$length
(df_fos1 <- df_fos1[order(df_fos1$llarg_nova, decreasing = TRUE),])

##      des fins      corr        dx        dy        dz    length
## 12  PKA  Erk  0.259596487 -31.3408343 19.5748302 17.1158161 40.723126
## 14  PKA  Akt  0.226368066 -31.0694241 20.5702235 17.5538571 41.189575
##  1  PKA  Mek  0.048366457 -25.7673093 31.3320057 16.4278777 43.766699
## 19  PKC  P38  0.488494531 -21.8074745 -0.8692771 -21.5928073 30.701318
## 16  PKA  Jnk  0.087830114 -31.4729774 20.0092071 15.4491832 40.368229
##  9  PKA  Raf  0.083669411 -25.9688852 18.8161605 18.4625857 37.004026
## 18  PKA  P38 -0.059613508 -30.7121359 18.4218761 12.7995533 38.031952
## 17  PKC  Jnk  0.210998302 -22.5683160 0.7180540 -18.9431773 29.473521
##  4  PIP3  PIP2  0.225912342 -7.3917630 18.1947023 15.6868359 25.134880
##  2  PKC  Mek  0.068699870 -16.8626478 12.0408526 -17.9644828 27.423597
## 13  PIP3  Akt -0.002099523 -12.5433772 17.6154611 15.5274458 26.622215
## 10  PKC  Raf  0.021805821 -17.0642238 -0.4749927 -15.9297749 23.348899
##  6  PIP2  PKC -0.007702225 17.0131484 -1.8583116 16.6791133 23.897560
##  7  Plcg  PKC  0.011182880 12.5078298 6.6387233 16.5037079 21.746053
##  3  Raf  Mek  0.732754099  0.2015760 12.5158453 -2.0347079 12.681761
##  8  PIP3  Plcg  0.135894703 -2.8864444 9.6976674 15.8622414 18.814543
## 11  Mek  Erk  0.083634460 -5.5735250 -11.7571755 0.6879384 13.029529
##  5  Plcg  PIP2  0.122102384 -4.5053186 8.4970349 -0.1754054 9.619161
## 15  Erk  Akt  0.902993735  0.2714102 0.9953933 0.4380410 1.120871

##      llarg_nova      prop
## 12     51.29471 1.2595965
## 14     50.51358 1.2263681
##  1     45.88354 1.0483665
## 19     45.69874 1.4884945
## 16     43.91378 1.0878301
##  9     40.10013 1.0836694
## 18     35.76473 0.9403865
## 17     35.69238 1.2109983
##  4     30.81316 1.2259123
##  2     29.30759 1.0686999
## 13     26.56632 0.9979005
## 10     23.85804 1.0218058
##  6     23.71350 0.9922978
##  7     21.98924 1.0111829
##  3     21.97437 1.7327541
##  8     21.37134 1.1358947
## 11     14.11925 1.0836345
##  5     10.79368 1.1221024
## 15     2.13301 1.9029937

```

### Taula 25

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

```
coords3Dn_diag_MOD <- coords3Dn_diag
```

```

# hom itera per cada aresta
for (i in seq_len(nrow(df_fos1))) {
  des <- df_fos1$des[i]
  fins <- df_fos1$fins[i]
  # indexes dels nodes
  v1 <- which(rownames(coords3Dn_diag_MOD) == des)
  v2 <- which(rownames(coords3Dn_diag_MOD) == fins)

  # posicions originals
  p1 <- coords3Dn_diag_MOD[v1, ]
  p2 <- coords3Dn_diag_MOD[v2, ]

  # vector direccional (de <des> cap a <fins>)
  v <- p2 - p1
  len_v <- sqrt(sum(v^2)) # Llargada actual de l'aresta
  v_unit <- v / len_v # direcció unitària

  # Factor d'ajust
  f <- df_fos1$prop[i]

  # increment de llargada que hom vol afegir o treure (repartit entre els 2 nodes)
  delta_L <- (f - 1) * len_v / 2

  # desplaçament geomètric al llarg de la direcció de l'aresta
  delta <- v_unit * delta_L

  # a partir de les POSICIONS ACTUALS (p1, p2), movem simètricament
  coords3Dn_diag_MOD[v1, ] <- p1 - delta
  coords3Dn_diag_MOD[v2, ] <- p2 + delta
}

(coords3Dn_diag_corr1_MOD <- normalitza_coords(coords3Dn_diag_MOD))

##          X          Y          Z
## Mek    -4.201799  21.1639981 -0.7128947
## PIP2   -4.116384   6.9873594  4.1768396
## PIP3    5.377279 -16.6691270 -16.3512546
## PKA    33.555081 -24.2787360 -21.8887767
## PKC    23.598034   2.1298462  27.8591251
## Plcg   1.326912  -3.9821363  3.4563165
## Raf    -5.096299  -1.5509680  4.9411446
## Erk    -13.242325  3.9222203  3.9709900
## Akt    -11.877878  7.1250507  4.4450338
## Jnk    -13.547468  4.5296862 -1.5536872
## P38   -11.775154  0.6228064 -8.3428363

```

Figura 24

```
open3d()

## wg1
## 58

plot3d(coords3Dn_diag_corr1_MOD,
        col = "red",
        size = 0.5,
        type = "s")
edges_mod <- df_fos1[, c("des", "fins")]

for (i in seq_len(nrow(edges_mod))) {
  v1 <- which(rownames(coords3Dn_diag_corr1_MOD) == edges_mod$des[i])
  v2 <- which(rownames(coords3Dn_diag_corr1_MOD) == edges_mod$fins[i])
  segments3d(rbind(coords3Dn_diag_corr1_MOD[v1, ], coords3Dn_diag_corr1_MOD[v2, ]),
             col = "grey60",
             lwd = 0.9)
}

offset <- seq(-0.04, 0.04, length.out=nrow(coords3Dn_diag_corr1_MOD))
text3d(coords3Dn_diag_corr1_MOD[, "X"]+0.02,
       coords3Dn_diag_corr1_MOD[, "Y"],
       coords3Dn_diag_corr1_MOD[, "Z"]+offset,
       texts = rownames(coords3Dn_diag_corr1_MOD),
       adj = c(0, 0.2),
       cex=0.95, depth_test="always")

segments3d(rbind(coords3Dn_diag_corr1_MOD[8, ], coords3Dn_diag_corr1_MOD[9, ]),
           col = "gold", lwd = 4)

centre <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z
# punt del centre
points3d(centre[1], centre[2], centre[3], color = "orange3", size = 6)
# 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]

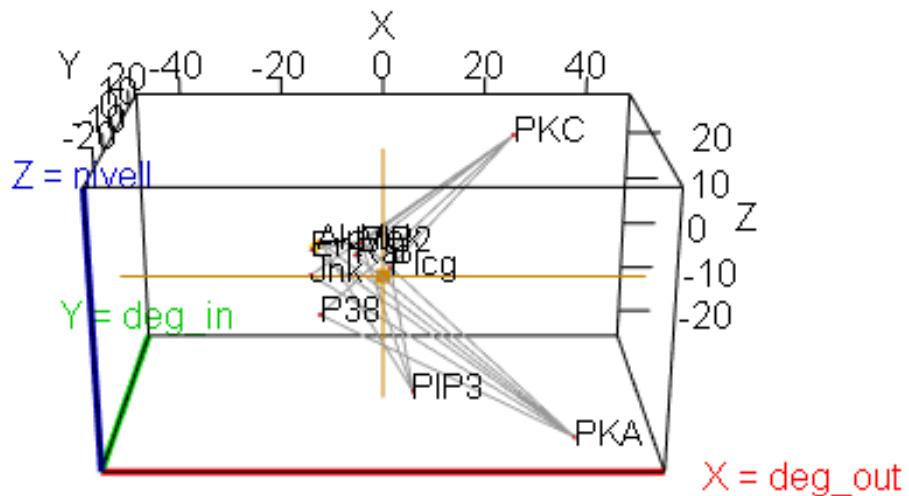
segments3d(rbind(c(xmin, ymin, zmin), c(xmax, ymin, zmin)), col="red", lwd=3)
text3d(xmax, ymin, zmin, texts="X = deg_out", color="red", adj=c(-0.2,0.5))

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

```



## 5.8.2 Càlcul de les coordenades del flux causal holístic (ZYX) del model topoempíric: flux\_fulla\_corregit

Caldria escatir quin és l'origen òptim en termes causals del vector director del flux causal holístic. Tot i que hom pot tenir en compte tant la bisectriu dels col·lididors fulla com la bisectriu dels nodes arrel, hom en pren en aquest TFM el centre de masses del model com a punt d'origen, no sols com a opció neutral sinó també conservadora: aquesta tria evitar moure artificalment el centre del con causal ni cap avant ni cap rere al llarg del flux.

El centre de masses sembla esdevé el punt de referència més robust a fi d'avaluar la coherència direccional global del model, atès que minimitza biaixos geomètrics induïts per la mateixa tria de l'origen.

*FLUX CAUSAL RETROLATERAL (4 col·lididors fulla) AMB COORDENADES coords3Dn\_diag\_corr1\_MOD (AMB ARESTA Erk->Akt)*

taula\_general\_fluxos\_fulla4MOD amb coords3Dn\_diag\_corr1\_MOD

Taula 26

```
collididor3n_PKA_P38_PKC <- crea_collididor_artificial(g_bosc, chain_nodes = c("PKA", "P38", "PKC"))
collididor3n_PKA_Jnk_PKC <- crea_collididor_artificial(g_bosc, chain_nodes = c("PKA", "Jnk", "PKC"))
collididor3n_PIP3_Akt_PKA <- crea_collididor_artificial(g_bosc, chain_nodes = c("PIP3", "Akt", "PKA"))
collididor3n_PKA_Erk_Mek <- crea_collididor_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_MOD)

# Flux fulla vertical (1 col·lididor fulla vertical)
flux1_coll_ver <- theta_flux_collididor_Z_fisic(collididor3n_PIP3_Akt_PKA, coords3Dn_diag_corr1_MOD)

# Fluxos fulles retroverticals (2 col·lididors fulla retro-verticals)
flux2_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1_MOD)
flux3_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1_MOD)

# Flux fulla laterovertical (1 col·lididor fulla Latero-vertical)
flux4_coll_laterover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1_MOD)
```

```

fluxos4MOD <- 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")
  )
)

## ---- Diferències angulars detallades ----
taula_general_fluxos_fulla4MOD <- taula_fluxos_general(fluxos4MOD)
resultat4MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla4MOD)

v_obj4MOD <- as.numeric(t(taula_general_fluxos_fulla4MOD[taula_general_fluxos_fulla4MOD$flux == "flux_objectiu", c("u_bar_X", "u_bar_Y", "u_bar_Z")])))

# normalitza vectors
ng4MOD <- resultat4MOD$v_global / sqrt(sum(resultat4MOD$v_global^2))
no4MOD <- v_obj4MOD / sqrt(sum(v_obj4MOD^2))

# angles per eix
ang_X4MOD <- acos(ng4MOD[1] * no4MOD[1]) * 180 / pi
ang_Y4MOD <- acos(ng4MOD[2] * no4MOD[2]) * 180 / pi
ang_Z4MOD <- acos(ng4MOD[3] * no4MOD[3]) * 180 / pi

```

```

# angle total 3D
ang_total4MOD <- acos(sum(ng4MOD * no4MOD)) * 180 / pi

resultat_angles4MOD <- list(
  angle_X = ang_X4MOD,
  angle_Y = ang_Y4MOD,
  angle_Z = ang_Z4MOD,
  angle_total_3D = ang_total4MOD,
  angles_fulla = as.numeric(resultat4MOD$angles_fulla),
  angle_max = resultat4MOD$angle_max,
  angle_mean = resultat4MOD$angle_mean
)

resultat4MOD

## $vectors_utilitzats
## [1] "flux1_coll_ver"      "flux2_coll_retrover"  "flux3_coll_retrover"
## [4] "flux4_coll_laterover"
##
## $v_sum
##   u_bar_X   u_bar_Y   u_bar_Z
## -277.73874 119.55773 42.92278
##
## $v_global
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.9093966  0.3914664  0.1405415
##
## $angle_vertical
##   u_bar_Z
## 81.92082
##
## $angle_objectiu
## [1] 84.35439
##
## $diferencia_angles
##   u_bar_Z
## 2.433569
##
## $angles_fulla
## [1] 27.27787 24.22136 14.16079 23.19637
##
## $angle_mean
## [1] 22.18832
##
## $angle_max
## [1] 27.27787

# bloc de codi fins a [1] per tal de dibuixar el vector director al plot3d sense va
Lors 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_bosc1, chain_nodes = c("PKA", "P38", "PKC"))
coords_colididor_PKA_P38_PKC <- coords3Dn_diag_corr1_MOD[c("PKA", "P38", "PKC"), c("X", "Y", "Z")]

flux2_coll_retrover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_P38_PKC, coords3Dn_diag_corr1_MOD)
res_flux1 <- flux2_coll_retrover
# Longitud camí
len_flux1 <- longitud_cami_fisic(colididor3n_PKA_P38_PKC, coords3Dn_diag_corr1_MOD)

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_bosc1, chain_nodes = c("PKA", "Jnk", "PKC"))
coords_colididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1_MOD[c("PKA", "Jnk", "PKC"), c("X", "Y", "Z")]

flux3_coll_retrover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1_MOD)
res_flux2 <- flux3_coll_retrover
# Longitud camí
len_flux2 <- longitud_cami_fisic(colididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1_MOD)

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_bosc1, chain_nodes = c("PKA", "Erk", "Mek"))

```

```

coords_colididor_PKA_Erk_Mek <- coords3Dn_diag_corr1_MOD[c("PKA", "Erk", "Mek"),
c("X", "Y", "Z")]

flux4_coll_laterover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1_MOD)
res_flux3 <- flux4_coll_laterover
# Longitud camí
len_flux3 <- longitud_cami_fisic(colididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1_MOD)

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")
colididor3n_PIP3_Akt_PKA <- crea_colididor_artificial(g_bosc1, chain_nodes = c("PIP3", "Akt", "PKA"))

coords_colididor_PIP3_Akt_PKA <- coords3Dn_diag_corr1_MOD[c("PIP3", "Akt", "PKA"),
c("X", "Y", "Z")]

flux1_coll_ver <- theta_flux_colididor_Z_fisic(colididor3n_PIP3_Akt_PKA, coords3Dn_diag_corr1_MOD)
res_flux4 <- flux1_coll_ver
# Longitud camí
len_flux4 <- longitud_cami_fisic(colididor3n_PIP3_Akt_PKA, coords3Dn_diag_corr1_MOD)

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_MOD)
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc1, vids = nodes_cadena_Raf_Mek_Erk)

# Longitud camí
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_

```

```

corr1_MOD)
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1_MOD)

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_MOD[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_MOD,
#                      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_corr1_MOD <- sweep(coords3Dn_diag_corr1_MOD, 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]
```

interseccio\_fluxos4MOD amb coords3Dn\_diag\_corr1\_MOD

Taula 27

```

res_holistic4MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla4MOD)

v_holos4MOD <- res_holistic4MOD$v_sum
p04MOD <- mean(coords3Dn_diag_corr1_MOD)
```

```

v_local4MOD <- as.numeric(res_flux$u_bar)
q04MOD <- p0_flux

(intersecció_fluxos4MOD <- angle_interseccio_fluxos(p04MOD, v_holos4MOD, q04MOD, v_
local4MOD))

## $angle_graus
## [1] 17.32834
##
## $distancia_minima
## [1] 0.055438
##
## $punt_holistic
##   u_bar_X   u_bar_Y   u_bar_Z
## -20.389119   8.776870   3.151011
##
## $punt_local
##      X          Y          Z
## -20.410805   8.738611   3.117257
##
## $vector_PQ
##      X          Y          Z
## -0.02168584 -0.03825923 -0.03375392

```

Plot casquet esfèric causal MODEL \*FLUX CAUSAL RETROLATERAL (4 col·lididors fulla)

*Figura 26 <Plot coords3Dn\_diag\_corr1\_MOD amb 4 vectors directors de flux AMB ARESTA Erk->Akt>*  
`open3d()`

```

## wgl
## 61

plot3d(coords3Dn_diag_corr1_MOD, col = "orangered", size = 1, type = "s")

edges1 <- as_edgelist(g_bosc1)
#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 = "_")
g_retrolateral3 <- add_edges(g_retrolateral1, c("Erk", "Akt"))
edges1_lat <- apply(as_edgelist(g_retrolateral3), 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_bosc1)$name == edges1[i, 1])
  v2 <- which(V(g_bosc1)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr1_MOD[v1, ], coords3Dn_diag_corr1_MOD[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

#offset <- seq(-0.04, 0.04, Length.out=nrow(coords3Dn_diag_corr1_MOD))
text3d(coords3Dn_diag_corr1_MOD[, "X"],
       coords3Dn_diag_corr1_MOD[, "Y"],
       coords3Dn_diag_corr1_MOD[, "Z"],
       texts = rownames(coords3Dn_diag_corr1_MOD),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc1)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1_MOD[7, ], coords3Dn_diag_corr1_MOD[1, ]),col
# = "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1_MOD[1, ], coords3Dn_diag_corr1_MOD[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,#[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical1 ", color = "green4",cex = 0.6,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux2,
  p1 = p1_flux2,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barlen = 0.02
)

```

```

)
text3d(p1_flux2-1,
       texts = "Flux retrovertical2 ", color = "green4",cex = 0.6,
       depth_test="always")

# segeta blava2
arrow3d(
  p0 = p0_flux3,
  p1 = p1_flux3,
  col = "green",
  type = "rotation",
  lwd = 2,
  width = 0.3,
  barblen = 0.02
)
text3d(p1_flux3+1,
       texts = "Flux laterovertical ", color = "green4",cex = 0.6,
       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+1,
       texts = "Flux vertical", color = "green4",cex = 0.6,
       depth_test="always")

#edges1 <- E(g_bosc1)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr1_MOD[7, ], coords3Dn_diag_corr1_MOD[1, ]),
            col = "purple", lwd = 1.5)
segments3d(rbind(coords3Dn_diag_corr1_MOD[1, ], coords3Dn_diag_corr1_MOD[8, ]),
            col = "purple", lwd = 1.5)

# segeta lila
arrow3d(
  p0 = p0_flux,
  p1 = p1_flux,
  col = "purple",
  type = "rotation",

```

```

lwd = 3,
width = 0.3,
barblen = 0.02
)
text3d(p1_flux[1]-6, p1_flux[2], p1_flux[3]-6,
       texts = paste0("Flux local versus causal | (Angle; Distancia) = (",
intersecció_fluxos4MOD$angle_graus,0), ";", round(intersecció_fluxos4MOD$distancia_mi
nima,1), ")"),
       color = "purple4",
       cex = 0.6,
       depth_test="always")

# el flux causal holístic

# hom escala la segeta perquè sigui comparable amb el model
#escala <- 40

#nodes_arrel <-c("PIP3", "PKA")

#p0 <- colMeans(coords3Dn_diag_corr1_MOD[nodes_arrel, , drop = FALSE])
#p0 <- coords3Dn_diag_corr5["Erk",]

#coords_fulls <- coords3Dn_diag_corr1_MOD[c("P38", "Jnk", "Erk", "Akt"), c("X", "Y", "Z
")]
#p0 <- colMeans(coords_fulls) # barycentre complet
#escala <- 40

taula_general_fluxos_fulla4MOD <- taula_fluxos_general(fluxos4MOD)
#resultat3MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla3MOD)

res_holistic4MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla4MOD)

## con causal
p04MOD <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z, punt d'or
igen
v_geom4MOD <- resultat4MOD$v_sum #suma de tots els fluxos fulla (vector geomètric d
el flux holístic)
h4MOD <- sqrt(sum(v_geom4MOD^2))/4# MÒDUL!!!! Llargària del con (la mateixa que la
del seu eix, el flux causal holístic)
v_h_norm4MOD <- res_holistic4MOD$v_global # vector holístic normalitzat: direcció n
ormalitzada del flux holístic

# EL CODI CORRECTE ÉS AQUEST p1 <- p0 + v_h_norm1 * h2 (h2 és el punt final del fl
ux holístic (direcció + mòdul real)) PERÒ HOM MULTIPLICA PER max(taula_general_flux
os_fulla111$modul) A FI DE NO FER ELS EIXOS DEL PLOT MASSA LLARGS

```

```

p14MOD <- p04MOD + v_h_norm4MOD * h4MOD
#p1 <- p0 + v_h_norm3MOD * max(taula_general_fluxos_fulla3MOD$modul) # EL CORRECTE
ÉS p1 NO PAS max(taula_general_fluxos_fulla000$modul)... intensitat del flux (per mitjà del mòdul): n'és l'eix

# paràmetres del con
theta_con4MOD <- res_holistic4MOD$angle_mean * pi / 180 # en radians
# EL CORRECTE : rad1 <- h2 * tan(theta_con1) # radi base
#rad3MOD <- max(taula_general_fluxos_fulla3MOD$modul) * tan(theta_con3MOD) # radi base
rad4MOD <- h4MOD * tan(theta_con4MOD)

arrow3d(
  p0 = p04MOD,
  p1 = p14MOD,
  type = "rotation",
  col = "red",
  lwd = 3,
  width = 0.3,
  barblen = 0.02
)
text3d(p1+4,
texts = paste0("Flux causal holistic | (Angle; Modul) = (",
  round(res_holistic4MOD$angle_mean,0), ";",
  round(h4MOD,0), ")"),
  col = "red",
  cex = 0.75,
  depth_test="always")

## ANGLES DE LES 4 FULLES RESPECTE AL VECTOR HOLISTIC
# 2. Extreiem els vectors de les 4 fulles
fulles4MOD <- taula_general_fluxos_fulla4MOD %>%
  filter(flux != "flux_objectiu") %>%
  select(u_bar_X, u_bar_Y, u_bar_Z) %>%
  as.matrix()

fulles_norm4MOD <- fulles4MOD / sqrt(rowSums(fulles4MOD^2))

angles4MOD <- acos(fulles_norm4MOD %*% v_h_norm4MOD) * 180/pi
#theta_con <- max(angles)# amplada del con = màxima variació angular

```

```
#theta_con# aquest ÉS l'angle correcte del con causal
```

```
#CON CAUSAL
# hom gira el con perquè apunti com el flux objectiu
z_axis <- c(0, 0, 1) # vector unitari Z
v_obj4MOD <- v_h_norm4MOD # <<<< el vector direcció normalitzat

# Producte vectorial per trobar eix de rotació
rot_axis4MOD <- pracma::cross(z_axis, v_obj4MOD)
rot_angle4MOD <- acos(sum(z_axis * v_obj4MOD))

# Matriu de rotació (Rodrigues)
if (sqrt(sum(rot_axis4MOD^2)) > 0) {
  k_vec4MOD <- rot_axis4MOD / sqrt(sum(rot_axis4MOD^2))

  k_mat4MOD <- matrix(c(0, -k_vec4MOD[3], k_vec4MOD[2],
                        k_vec4MOD[3], 0, -k_vec4MOD[1],
                        -k_vec4MOD[2], k_vec4MOD[1], 0),
                        nrow = 3,
                        byrow = TRUE)

  RRR <- diag(3) * cos(rot_angle4MOD) +
    sin(rot_angle4MOD) * k_mat4MOD +
    (1 - cos(rot_angle4MOD)) * (k_vec4MOD %*% t(k_vec4MOD))
} else {
  RRR <- diag(3)
}

# hom renderitza el con
n_cercle <- 20 # nombre de punts que aproximen el cercle
theta <- seq(0, 2 * pi, length.out = n_cercle)
x4MOD <- rad4MOD * cos(theta)
y4MOD <- rad4MOD * sin(theta)
# EL CORRECTE z <- rep(max(h2, n_cercle),
z4MOD <- rep(h4MOD, n_cercle)

# con alineat amb l'eix Z i vèrtex a l'origen
cone_base4MOD <- rbind(x4MOD, y4MOD, z4MOD) # 3 x n_cercle
#vertex <- matrix(c(0,0,0), nrow = 3, ncol = 1) # 3x1
# Apliquem la rotació correcta
cone_rot4MOD <- RRR %*% cone_base4MOD

#origin <- c(B["X"], B["Y"], A["Z"])
```

```

origin4MOD <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z
cone_pts4MOD <- sweep(t(cone_rot4MOD), 2, origin4MOD, "+")

triangles <- c()
for (i in 1:(n_cercle - 1)) {
  triangles <- rbind(triangles, cbind(i, i + 1, n_cercle + 1))
}

cone_mesh <- tmesh3d(
  vertices = t(rbind(cone_pts4MOD, origin4MOD)),
  indices = t(triangles),
  homogeneous = FALSE
)

# 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_corr1_MOD[c("P38", "Jnk", "Erk", "Akt"), c("X", "Y", "Z")]
#B <- colMeans(coords_fulles) # barycentre complet
#B <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z

## --- 2) mitjana del nivell Z dels nodes arrel ---

#coords_arrel <- coords3Dn_diag_corr1_MOD[c("PIP3", "PKA"), ]
#A <- colMeans(coords_arrel)

## --- 3) punt híbrid ---

shade3d(cone_mesh, alpha = 0.18, col = "red")
#spheres3d(0,0,0, radius = h*0.03) # punt del vèrtex a fi d'orientar-se
text3d(origin4MOD, texts = "Con causal holistic (4MOD)", col = "red", cex = 0.6)

aspect3d(1,1,1)

## =====
## ESFERA TANGENT A LA BASE DEL CON (CON SAGRAT, NO ES TOCA)

```

```

## =====

par3d(ignoreExtent = TRUE)
par3d(scale = c(1,1,1))

# --- centre sagrat (centre de masses) ---
centre <- colMeans(coords3Dn_diag_corr1_MOD)

# --- paràmetres sagrats del con (JA DEFINITS ABANS) ---
L4MOD <- h4MOD
theta_con4MOD <- res_holistic4MOD$angle_mean * pi/180

# --- radi CORRECTE de l'esfera: base del con en contacte ---
r4MOD <- L4MOD / cos(theta_con4MOD)

# --- malla esfèrica ---
nu <- 20
nv <- 20

u <- seq(0, pi, length.out = nu)
v <- seq(0, 2*pi, length.out = nv)

Xs4MOD <- centre[1] + r4MOD * outer(sin(u), cos(v))
Ys4MOD <- centre[2] + r4MOD * outer(sin(u), sin(v))
Zs4MOD <- centre[3] + r4MOD * outer(cos(u), rep(1, nv))

# --- render de l'esfera ---
surface3d(
  Xs4MOD, Ys4MOD, Zs4MOD,
  color = "gray77",
  alpha = 0.20,
  lit = FALSE
)

# --- centre (orientació visual) ---
points3d(
  centre[1], centre[2], centre[3],
  col = "orangered4",
  size = 6
)

## =====
## CASQUET ESFÈRIC ADMISSIBLE (OMBRA DEL CON)
## =====

# direcció unitària del con
v_dir4MOD <- as.numeric(res_holistic4MOD$v_global)

```

```

# resolució del casquet
nu_cap <- 20
nv_cap <- 20

# angle polar limitat al con
u_cap4MOD <- seq(0, theta_con4MOD, length.out = nu_cap)
v_cap4MOD <- seq(0, 2*pi, length.out = nv_cap)

# casket en coordenades esfèriques (eix Z)
Xs04MOD <- outer(sin(u_cap4MOD), cos(v_cap4MOD))
Ys04MOD <- outer(sin(u_cap4MOD), sin(v_cap4MOD))
Zs04MOD <- outer(cos(u_cap4MOD), rep(1, nv_cap))

# rotació Z -> v_h_norm (Rodrigues, reutilitzant criteri)
z_axis <- c(0,0,1)
rot_axis4MOD <- pracma::cross(z_axis, v_dir4MOD)
rot_angle4MOD <- acos(sum(z_axis * v_dir4MOD))

if (sqrt(sum(rot_axis^2)) > 0) {
  kkk_vec <- rot_axis4MOD / sqrt(sum(rot_axis4MOD^2))
  kkk_mat <- matrix(c(0,-kkk_vec[3],kkk_vec[2],
                      kkk_vec[3],0,-kkk_vec[1],
                      -kkk_vec[2],kkk_vec[1],0), 3,3, byrow=TRUE)
  RRRcap <- diag(3)*cos(rot_angle4MOD) +
    sin(rot_angle4MOD)*kkk_mat +
    (1-cos(rot_angle4MOD))*(kkk_vec %*% t(kkk_vec))
} else {
  RRRcap <- diag(3)
}

# aplicar rotació, escala i trasllat
cap_rot <- RRRcap %*% rbind(
  as.vector(Xs04MOD),
  as.vector(Ys04MOD),
  as.vector(Zs04MOD)
)

Xs_cap4MOD <- matrix(cap_rot[1,], nu_cap, nv_cap) * r4MOD + centre[1]
Ys_cap4MOD <- matrix(cap_rot[2,], nu_cap, nv_cap) * r4MOD + centre[2]
Zs_cap4MOD <- matrix(cap_rot[3,], nu_cap, nv_cap) * r4MOD + centre[3]

# render del casquet
surface3d(
  Xs_cap4MOD, Ys_cap4MOD, Zs_cap4MOD,
  color = "red",
  alpha = 0.35,
  lit = FALSE
)

```

```

bbox <- par3d("bbox")
xmin <- bbox[1]; xmax <- bbox[2]
ymin <- bbox[3]; ymax <- bbox[4]
zmin <- bbox[5]; zmax <- bbox[6]
centre <- colMeans(coords3Dn_diag_corr1_MOD) # 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)

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))

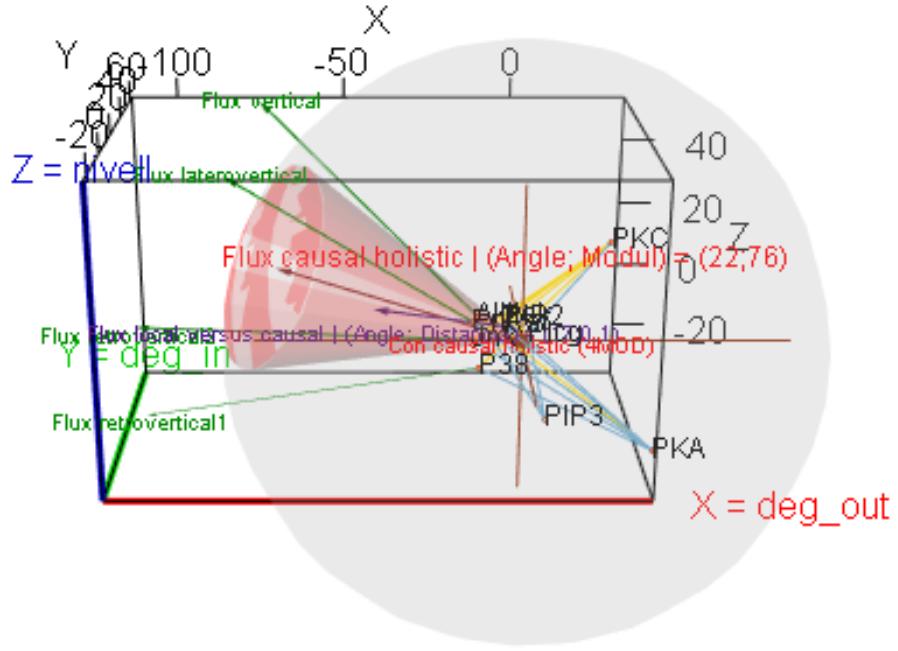
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

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

```



*FLUX CAUSAL RETROLATERAL (3 col·lididors fulla) AMB COORDENADES  
coords3Dn\_diag\_corr1\_MOD (AMB ARESTA Erk->Akt)*

*taula\_general\_fluxos\_fulla3MOD amb coords3Dn\_diag\_corr1\_MOD*

```
# 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_colididor_PKA_P38_PKC <- c("PKA", "P38", "PKC")
collididor3n_PKA_P38_PKC <- crea_colididor_artificial(g_bosc11, chain_nodes = c("PKA", "P38", "PKC"))
coords_colididor_PKA_P38_PKC <- coords3Dn_diag_corr1_MOD[c("PKA", "P38", "PKC"), c("X", "Y", "Z")]

flux2_coll_retrover <- theta_flux_colididor_Z_fisic(colididor3n_PKA_P38_PKC, coords3Dn_diag_corr1_MOD)
res_flux1 <- flux2_coll_retrover
# Longitud camí
len_flux1 <- longitud_cami_fisic(colididor3n_PKA_P38_PKC, coords3Dn_diag_corr1_MOD)
```

```

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_bosc1, chain_nodes = c("PKA", "Jnk", "PKC"))
coords_collididor_PKA_Jnk_PKC <- coords3Dn_diag_corr1_MOD[c("PKA", "Jnk", "PKC"),
c("X", "Y", "Z")]

flux3_coll_retrover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1_MOD)
res_flux2 <- flux3_coll_retrover
# Longitud camí
len_flux2 <- longitud_cami_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1_MOD)

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_Erk_Mek <- c("PKA", "Erk", "Mek")
#collididor3n_PKA_Erk_Mek <- crea_collididor_artificial(g_bosc1, chain_nodes = c("PKA", "Erk", "Mek"))

#coords_collididor_PKA_Erk_Mek <- coords3Dn_diag_corr1_MOD[c("PKA", "Erk", "Mek"),
c("X", "Y", "Z")]

#flux4_coll_laterover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1_MOD)
#res_flux3 <- flux4_coll_laterover
# Longitud camí
#Len_flux3 <- longitud_cami_fisic(collididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1_MOD)

#r_in3 <- as.numeric(res_flux3$r_in)
#u_dir3 <- r_in3 / sqrt(sum(r_in3^2))

#p0_flux3 <- coords_collididor_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_bosc1, chain_nodes = c(
  "PIP3", "Akt", "PKA"))
collididor3n_PIP3_Akt_PKA_Erk <- crea_colididor_k(g_bosc1, centre = "Akt", parents = c("PKA", "PIP3", "Erk"))

coords_colididor_PIP3_Akt_PKA_Erk <- coords3Dn_diag_corr1_MOD[c("PIP3", "Akt", "PKA", "Erk"), c("X", "Y", "Z")]

#fLux1_coll_ver <- theta_flux_colididor_Z_fisic(collididor3n_PIP3_Akt_PKA_Erk, coords3Dn_diag_corr1_MOD)
flux1_coll_ver <- theta_flux_colididor_k(collididor3n_PIP3_Akt_PKA_Erk, coords3Dn_diag_corr1_MOD)

res_flux4 <- flux1_coll_ver
# Longitud camí
len_flux4 <- longitud_cami_fisic(collididor3n_PIP3_Akt_PKA_Erk, coords3Dn_diag_corr1_MOD)

r_in4 <- as.numeric(res_flux4$r_in)
u_dir4 <- r_in4 / sqrt(sum(r_in4^2))

p0_flux4 <- coords_colididor_PIP3_Akt_PKA_Erk["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_MOD)
#nodes_cadena_PIP3_Plcg_PKC <- c("PIP3", "Plcg", "PKC")
#cadena3n_PIP3_Plcg_PKC <- induced_subgraph(g_bosc1, vids = nodes_cadena_PIP3_Plcg_PKC)

# Longitud camí
#fLux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1_MOD)
#Len_flux <- longitud_cami_fisic(cadena3n_PIP3_Plcg_PKC, coords3Dn_diag_corr1_MOD)

#res_flux <- fLux_objectiu
#u_bar <- as.numeric(res_flux$u_bar)
#u_dir <- u_bar / sqrt(sum(u_bar^2))

#coords_cadena3n_PIP3_Plcg_PKC <- coords3Dn_diag_corr1_MOD[ c("PIP3", "Plcg", "PKC"),
  c("X", "Y", "Z")]
#p0_flux <- coords_cadena3n_PIP3_Plcg_PKC["Plcg",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_MOD,
#
#           p0_flux1, p1_flux1,
#           p0_flux2, p1_flux2,
#
#           p0_flux4, p1_flux4)
# hom calcula quant s'ha de desplaçar cada eix perquè el mínim sigui 000
#offset6_MOD <- apply(all_coords1, 2, min)

# hom aplica el desplaçament
#coords3Dn_diag_corr1_MOD_MOD <- sweep(coords3Dn_diag_corr1_MOD, 2, offset6_MOD, "-")

#p0_flux1 <- p0_flux1 - offset6_MOD
#p1_flux1 <- p1_flux1 - offset6_MOD
#p0_flux2 <- p0_flux2 - offset6_MOD
#p1_flux2 <- p1_flux2 - offset6_MOD

#p0_flux4 <- p0_flux4 - offset6_MOD
#p1_flux4 <- p1_flux4 - offset6_MOD

# [2]
#
#cadena3n_Raf_Mek_Erk <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1_MOD)
nodes_cadena_Raf_Mek_Erk <- c("Raf", "Mek", "Erk")
cadena3n_Raf_Mek_Erk <- induced_subgraph(g_bosc1, vids = nodes_cadena_Raf_Mek_Erk)

# Longitud camí
flux_objectiu <- theta_flux_vectorial_Z_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1_MOD)
len_flux <- longitud_cami_fisic(cadena3n_Raf_Mek_Erk, coords3Dn_diag_corr1_MOD)

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_MOD[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
#_____
```

### Taula 28

```
collididor3n_PKA_P38_PKC <- crea_collididor_artificial(g_bosc, chain_nodes = c("PKA", "P38", "PKC"))
collididor3n_PKA_Jnk_PKC <- crea_collididor_artificial(g_bosc, chain_nodes = c("PKA", "Jnk", "PKC"))
#collididor3n_PKA_Erk_Mek <- crea_collididor_artificial( g_bosc, chain_nodes = c("PKA", "Erk", "Mek"))

collididor3n_PIP3_Akt_PKA_Erk <- crea_collididor_k(g_bosc, centre="Akt", parents = c("PIP3", "PKA", "Erk"))

#_____
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_MOD)

# Flux fulla vertical (1 col·lididor fulla vertical)

# Fluxos fulles retroverticals (2 col·lididors fulla retro-verticals)
flux2_coll_retrover1 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_P38_PKC, coords3Dn_diag_corr1_MOD)
flux3_coll_retrover2 <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Jnk_PKC, coords3Dn_diag_corr1_MOD)

# Flux fulla laterovertical (1 col·lididor fulla latero-vertical)
#flux4_coll_laterover <- theta_flux_collididor_Z_fisic(collididor3n_PKA_Erk_Mek, coords3Dn_diag_corr1_MOD)
flux1_coll_laterover <- theta_flux_collididor_k(collididor3n_PIP3_Akt_PKA_Erk, coords3Dn_diag_corr1_MOD)

fluxos3MOD <- list(
  list(
    nom = "flux_objectiu",
    res = flux_objectiu,
    nodes = c("Raf", "Mek", "Erk")
  ),
  list(
    nom = "flux1_coll_laterover",
    res = flux1_coll_laterover,
    nodes = c("PIP3", "Akt", "PKA", "Erk")
  ),
  list(
    nom = "flux2_coll_retrover1",
```

```

    res = flux2_coll_retrover1,
    nodes = c("PKA", "P38", "PKC")
),
list(
  nom = "flux3_coll_retrover2",
  res = flux3_coll_retrover2,
  nodes = c("PKA", "Jnk", "PKC")
)#
#list(
#  nom = "flux4_coll_laterover",
#  res = flux4_coll_laterover,
#  nodes = c("PKA", "Erk", "Mek")
#)
)
## ---- Diferències angulars detallades ----
taula_general_fluxos_fulla3MOD <- taula_fluxos_general(fluxos3MOD)
resultat3MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla3MOD)

v_obj3MOD <- as.numeric(t(taula_general_fluxos_fulla3MOD[taula_general_fluxos_fulla3MOD$flux == "flux_objectiu", c("u_bar_X", "u_bar_Y", "u_bar_Z")])))

# normalitza vectors
ng3MOD <- resultat3MOD$v_global / sqrt(sum(resultat3MOD$v_global^2))
no3MOD <- v_obj3MOD / sqrt(sum(v_obj3MOD^2))

# angles per eix
ang_X3MOD <- acos(ng3MOD[1] * no3MOD[1]) * 180 / pi
ang_Y3MOD <- acos(ng3MOD[2] * no3MOD[2]) * 180 / pi
ang_Z3MOD <- acos(ng3MOD[3] * no3MOD[3]) * 180 / pi

# angle total 3D
ang_total3MOD <- acos(sum(ng3MOD * no3MOD)) * 180 / pi

resultat_angles3MOD <- list(
  angle_X = ang_X3MOD,
  angle_Y = ang_Y3MOD,
  angle_Z = ang_Z3MOD,
  angle_total_3D = ang_total3MOD,
  angles_fulla = as.numeric(resultat3MOD$angles_fulla),
  angle_max = resultat3MOD$angle_max,
  angle_mean_collididors = resultat3MOD$angle_mean
)
resultat3MOD

```

```

## $vectors_utilitzats
## [1] "flux1_coll_laterover" "flux2_coll_retrover1" "flux3_coll_retrover2"
##
## $v_sum
##   u_bar_X   u_bar_Y   u_bar_Z
## -226.07326 112.92981 15.89437
##
## $v_global
##   u_bar_X   u_bar_Y   u_bar_Z
## -0.8928320  0.4459941  0.0627717
##
## $angle_vertical
##   u_bar_Z
## 86.40108
##
## $angle_objectiu
## [1] 84.35439
##
## $diferencia_angles
##   u_bar_Z
## 2.046695
##
## $angles_fulla
## [1] 30.40426 21.32849 11.23928
##
## $angle_mean
## [1] 21.22882
##
## $angle_max
## [1] 30.40426

```

## interseccio\_fluxos amb coords3Dn\_diag\_corr1\_MOD

*Taula 29*

```

res_holistic3MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla3MOD)

v_holos3MOD <- res_holistic3MOD$v_sum
p03MOD <- mean(coords3Dn_diag_corr1_MOD)

v_local3MOD <- as.numeric(res_flux$u_bar)
q03MOD <- p0_flux

(interseccio_fluxos3MOD <- angle_interseccio_fluxos(p03MOD, v_holos3MOD, q03MOD, v_local3MOD))

## $angle_graus
## [1] 11.80659
##
## $distancia_minima
## [1] 0.05032201

```

```

## 
## $punt_holistic
##   u_bar_X    u_bar_Y    u_bar_Z
## -29.139825  14.556144  2.048713
##
## $punt_local
##   X          Y          Z
## -29.120467  14.590498  2.079975
##
## $vector_PQ
##   X          Y          Z
## 0.01935875  0.03435407  0.03126246

```

Plot casquet esfèric causal MODEL \*FLUX CAUSAL RETROLATERAL (4 col·lididors fulla)

*Figura 27 <Plot coords3Dn\_diag\_corr1\_MOD amb 3 vectors directors de flux AMB ARESTA Erk->Akt>*  
**open3d()**

```

## wgl
## 64

plot3d(coords3Dn_diag_corr1_MOD, col = "orangered", size = 1, type = "s")

edges1 <- as_edgelist(g_bosc1)
#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 = "_")
g_retrolateral3 <- add_edges(g_retrolateral1, c("Erk", "Akt"))

edges1_lat <- apply(as_edgelist(g_retrolateral3), 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_bosc1)$name == edges1[i, 1])
  v2 <- which(V(g_bosc1)$name == edges1[i, 2])
  segments3d(rbind(coords3Dn_diag_corr1_MOD[v1, ], coords3Dn_diag_corr1_MOD[v2, ]),
             col = edge_colors[i], lwd = 0.9)
}

```

```

#offset <- seq(-0.04, 0.04, Length.out=nrow(coords3Dn_diag_corr1_MOD))
text3d(coords3Dn_diag_corr1_MOD[, "X"],
       coords3Dn_diag_corr1_MOD[, "Y"],
       coords3Dn_diag_corr1_MOD[, "Z"],
       texts = rownames(coords3Dn_diag_corr1_MOD),
       adj = c(0, 0.2),
       cex = 0.75,
       depth_test="always")

#edges1 <- E(g_bosc1)[c(18,1)]
#segments3d(rbind(coords3Dn_diag_corr1_MOD[7, ], coords3Dn_diag_corr1_MOD[1, ]),col
= "blue", lwd = 1.5)
#segments3d(rbind(coords3Dn_diag_corr1_MOD[1, ], coords3Dn_diag_corr1_MOD[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,#[1], p1_flux1[2], p1_flux1[3] + 4,
       texts = "Flux retrovertical1 ", color = "green4",cex = 0.6,
       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_flux2-1,
       texts = "Flux retrovertical2 ", color = "green4",cex = 0.6,
       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+1,
  texts = "Flux laterovertical", color = "green4",cex = 0.6,
  depth_test="always")

# 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]-6, p1_flux[2], p1_flux[3]-6,
  texts = paste0("Flux local versus causal | (Angle; Distancia) = (",round(interseccio_fluxos3MOD$angle_graus,0), ";", round(interseccio_fluxos3MOD$distancia_minima,1), ")"),
  color = "purple4",
  cex = 0.6,
  depth_test="always")

```

```

#edges1 <- E(g_bosc1)[c(18,1)]
segments3d(rbind(coords3Dn_diag_corr1_MOD[8, ], coords3Dn_diag_corr1_MOD[9, ]),
            col = "blue", lwd = 3)
#segments3d(rbind(coords3Dn_diag_corr1_MOD[1, ], coords3Dn_diag_corr1_MOD[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")

#taula_general_fluxos_fulla000 <- taula_fluxos_general000(fluxos000)
#resultat1 <- mitjana_vectorial_global000(taula_general_fluxos_fulla000)
# el flux causal holistic

taula_general_fluxos_fulla3MOD <- taula_fluxos_general(fluxos3MOD)
#resultat3MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla3MOD)

res_holistic3MOD <- mitjana_vectorial_global(taula_general_fluxos_fulla3MOD)

## con causal
p03MOD <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z, punt d'origen
v_geom3MOD <- resultat3MOD$v_sum #suma de tots els fluxos fulla (vector geomètric del flux holistic)
h3MOD <- sqrt(sum(v_geom3MOD^2))/3# MÒDUL!!!! Llargària del con (la mateixa que la del seu eix, el flux causal holistic)
v_h_norm3MOD <- res_holistic3MOD$v_global # vector holistic normalitzat: direcció normalitzada del flux holistic

# EL CODI CORRECTE ÉS AQUEST p1 <- p0 + v_h_norm1 * h2 (h2 és el punt final del flux holistic (direcció + mòdul real)) PERO HOM MULTIPLICA PER max(taula_general_fluxos_fulla111$modul) A FI DE NO FER ELS EIXOS DEL PLOT MASSA LLARGS

p13MOD <- p03MOD + v_h_norm3MOD * h3MOD
#p1 <- p0 + v_h_norm3MOD * max(taula_general_fluxos_fulla3MOD$modul) # EL CORRECTE ÉS p1 NO PAS max(taula_general_fluxos_fulla000$modul)... intensitat del flux (per m

```

itjà del mòdul): n'és l'eix

```
# paràmetres del con
theta_con3MOD <- res_holistic3MOD$angle_mean * pi / 180 # en radians
# EL CORRECTE : rad1 <- h2 * tan(theta_con1) # radi base
#rad3MOD <- max(taula_general_fluxos_fulla3MOD$modul) * tan(theta_con3MOD) # radi base
rad3MOD <- h3MOD * tan(theta_con3MOD)

arrow3d(
  p0 = p03MOD,
  p1 = p13MOD,
  type = "rotation",
  col = "red",
  lwd = 3,
  width = 0.3,
  barrlen = 0.02
)

text3d(p13MOD+2,
texts = paste0("Flux causal holistic | (Angle; Modul) = (",round(res_holistic3MOD$angle_mean,0), ";", round(h3MOD,0), ")"),
  col = "red",
  cex = 0.75,
  depth_test="always")

## ANGLES DE LES 4 FULLES RESPECTE AL VECTOR HOLISTIC
# 2. Extreiem els vectors de les 4 fulles
fulles3MOD <- taula_general_fluxos_fulla3MOD %>%
  filter(flux != "flux_objectiu") %>%
  select(u_bar_X, u_bar_Y, u_bar_Z) %>%
  as.matrix()

fulles_norm3MOD <- fulles3MOD / sqrt(rowSums(fullles3MOD^2))

angles3MOD <- acos(fulles_norm3MOD %*% v_h_norm3MOD) * 180/pi
#theta_con <- max(angles)# amplada del con = màxima variació angular

#theta_con# aquest ÉS l'angle correcte del con causal
```

```

#CON CAUSAL
# hom gira el con perquè apunti com el flux objectiu
z_axis <- c(0, 0, 1) # vector unitari Z
v_obj3MOD <- v_h_norm3MOD # <<< el vector direcció normalitzat

# Producte vectorial per trobar eix de rotació
rot_axis3MOD <- pracma::cross(z_axis, v_obj3MOD)
rot_angle3MOD <- acos(sum(z_axis * v_obj3MOD))

# Matriu de rotació (Rodrigues)
if (sqrt(sum(rot_axis3MOD^2)) > 0) {
  k_vec3MOD <- rot_axis3MOD / sqrt(sum(rot_axis3MOD^2))

  k_mat3MOD <- matrix(c(0, -k_vec3MOD[3], k_vec3MOD[2],
                        k_vec3MOD[3], 0, -k_vec3MOD[1],
                        -k_vec3MOD[2], k_vec3MOD[1], 0),
                        nrow = 3,
                        byrow = TRUE)

  RR <- diag(3) * cos(rot_angle3MOD) +
    sin(rot_angle3MOD) * k_mat3MOD +
    (1 - cos(rot_angle3MOD)) * (k_vec3MOD %*% t(k_vec3MOD))
} else {
  RR <- diag(3)
}

# hom renderitza el con
n_cercle <- 20 # nombre de punts que aproximen el cercle
theta <- seq(0, 2 * pi, length.out = n_cercle)
x3MOD <- rad3MOD * cos(theta)
y3MOD <- rad3MOD * sin(theta)
# EL CORRECTE z <- rep(max(h2, n_cercle),
z3MOD <- rep(h3MOD, n_cercle)

# con alineat amb l'eix Z i vèrtex a l'origen
cone_base3MOD <- rbind(x3MOD, y3MOD, z3MOD) # 3 x n_cercle
#vertex <- matrix(c(0,0,0), nrow = 3, ncol = 1) # 3x1
# Apliquem la rotació correcta
cone_rot3MOD <- RR %*% cone_base3MOD

#origin <- c(B["X"], B["Y"], A["Z"])
origin3MOD <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z
cone_pts3MOD <- sweep(t(cone_rot3MOD), 2, origin3MOD, "+")

```

```

triangles <- c()
for (i in 1:(n_cercle - 1)) {
  triangles <- rbind(triangles, cbind(i, i + 1, n_cercle + 1))
}

cone_mesh <- tmesh3d(
  vertices = t(rbind(cone_pts3MOD, origin3MOD)),
  indices = t(triangles),
  homogeneous = FALSE
)

# 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_fullles <- coords3Dn_diag_corr1_MOD[c("P38", "Jnk", "Erk", "Akt"), c("X", "Y", "Z")]
#B <- colMeans(coords_fullles) # barycentre complet
#B <- colMeans(coords3Dn_diag_corr1_MOD) # mitjana de cada eix X,Y,Z

## --- 2) mitjana del nivell Z dels nodes arrel ---

#coords_arrel <- coords3Dn_diag_corr1_MOD[c("PIP3", "PKA"), ]
#A <- colMeans(coords_arrel)

## --- 3) punt híbrid ---

shade3d(cone_mesh, alpha = 0.18, col = "red")
#spheres3d(0,0,0, radius = h*0.03) # punt del vèrtex a fi d'orientar-se
text3d(origin3MOD, texts = "Con causal holistic (3MOD)", col = "red", cex = 0.6)

aspect3d(1,1,1)

## =====
## ESFERA TANGENT A LA BASE DEL CON (CON SAGRAT, NO ES TOCA)
## =====

```

```

par3d(ignoreExtent = TRUE)
par3d(scale = c(1,1,1))

# --- centre sagrat (centre de masses) ---
centre <- colMeans(coords3Dn_diag_corr1_MOD)

# --- paràmetres sagrats del con (JA DEFINITS ABANS) ---
L3MOD <- h3MOD
theta_con3MOD <- res_holistic3MOD$angle_mean * pi/180

# --- radi CORRECTE de l'esfera: base del con en contacte ---
r3MOD <- L3MOD / cos(theta_con3MOD)

# --- malla esfèrica ---
nu <- 20
nv <- 20

u <- seq(0, pi, length.out = nu)
v <- seq(0, 2*pi, length.out = nv)

Xs3MOD <- centre[1] + r3MOD * outer(sin(u), cos(v))
Ys3MOD <- centre[2] + r3MOD * outer(sin(u), sin(v))
Zs3MOD <- centre[3] + r3MOD * outer(cos(u), rep(1, nv))

# --- render de l'esfera ---
surface3d(
  Xs3MOD, Ys3MOD, Zs3MOD,
  color = "gray77",
  alpha = 0.20,
  lit = FALSE
)

# --- centre (orientació visual) ---
points3d(
  centre[1], centre[2], centre[3],
  col = "orangered4",
  size = 6
)

## =====
## CASQUET ESFÈRIC ADMISSIBLE (OMBRA DEL CON)
## =====

# direcció unitària del con
v_dir3MOD <- as.numeric(res_holistic3MOD$v_global)

# resolució del casquet

```

```

nu_cap <- 20
nv_cap <- 20

# angle polar limitat al con
u_cap3MOD <- seq(0, theta_con3MOD, length.out = nu_cap)
v_cap3MOD <- seq(0, 2*pi, length.out = nv_cap)

# casquet en coordenades esfèriques (eix Z)
Xs03MOD <- outer(sin(u_cap3MOD), cos(v_cap3MOD))
Ys03MOD <- outer(sin(u_cap3MOD), sin(v_cap3MOD))
Zs03MOD <- outer(cos(u_cap3MOD), rep(1, nv_cap))

# rotació Z -> v_h_norm (Rodrigues, reutilitzant criteri)
z_axis <- c(0,0,1)
rot_axis3MOD <- pracma::cross(z_axis, v_dir3MOD)
rot_angle3MOD <- acos(sum(z_axis * v_dir3MOD))

if (sqrt(sum(rot_axis^2)) > 0) {
  kk_vec <- rot_axis3MOD / sqrt(sum(rot_axis3MOD^2))
  kk_mat <- matrix(c(0,-kk_vec[3],kk_vec[2],
                     kk_vec[3],0,-kk_vec[1],
                     -kk_vec[2],kk_vec[1],0), 3,3, byrow=TRUE)
  RRcap <- diag(3)*cos(rot_angle3MOD) +
    sin(rot_angle3MOD)*kk_mat +
    (1-cos(rot_angle3MOD))*(kk_vec %*% t(kk_vec))
} else {
  RRcap <- diag(3)
}

# aplicar rotació, escala i trasllat
cap_rot <- RRcap %*% rbind(
  as.vector(Xs03MOD),
  as.vector(Ys03MOD),
  as.vector(Zs03MOD)
)
Xs_cap3MOD <- matrix(cap_rot[1,], nu_cap, nv_cap) * r3MOD + centre[1]
Ys_cap3MOD <- matrix(cap_rot[2,], nu_cap, nv_cap) * r3MOD + centre[2]
Zs_cap3MOD <- matrix(cap_rot[3,], nu_cap, nv_cap) * r3MOD + centre[3]

# render del casquet
surface3d(
  Xs_cap3MOD, Ys_cap3MOD, Zs_cap3MOD,
  color = "red",
  alpha = 0.35,
  lit = FALSE
)

```

```

centre <- colMeans(coords3Dn_diag_corr1_MOD) # 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))

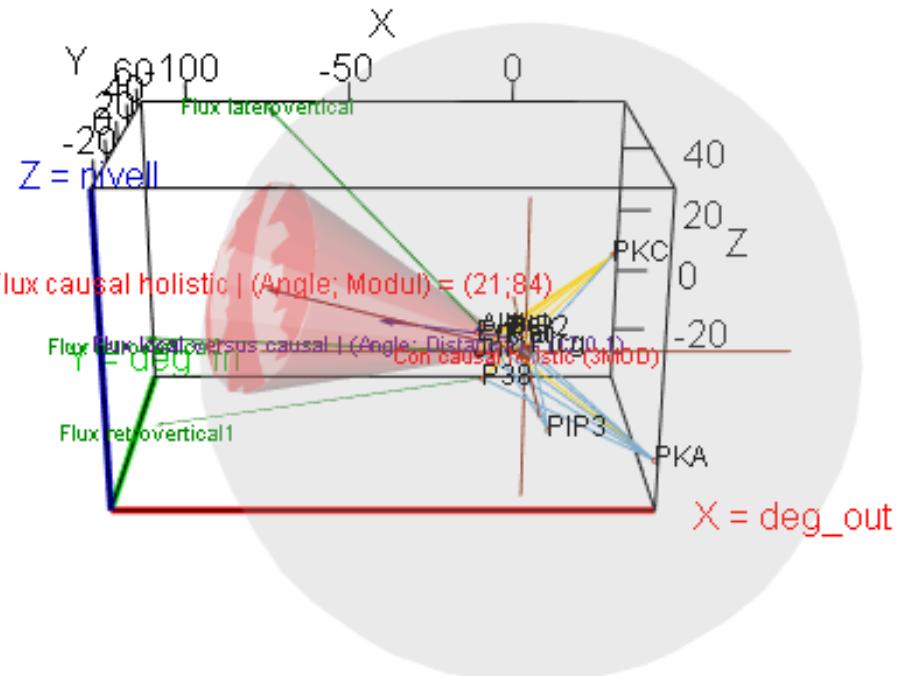
segments3d(rbind(c(xmin, ymin, zmin), c(xmin, ymax, zmin)), col="green3", lwd=3)
text3d(xmin, ymax, zmin, texts="Y = deg_in", color="green3", adj=c(0.5,-0.2))

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

```



## ANNEX2 CODI R COMENTAT PER A LA VALIDACIÓ ESTRUCTURAL DEL MODEL CAUSAL GEOMÈTRIC: TOR FUNCIONAL DEFORMABLE

Aquí hom té 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 pugui 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 ( $in>=1$  i  $out>=2$ ) o bé ( $in>=2$  i  $out>=1$ ) i que hi hagi com a mínim un col·lididor-fulla ( $in>=2$ ,  $out=0$ ), el DAG ja comença a esser 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 les 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 divergeix. Per exemple, un pathway metabòlic: l'enzim inicial (arrel) comença una cadena de reaccions, diversos enzims mitjancers (laterals) comparteixen substrats (aquí apareix el ciliar 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 horitzontal (hi ha lateralitat, no és una cadena ni tampoc un arbre) i segon vertical (amb un únic col·lididor-fulla ja comença a esser possible), hom obté el model tor empíric deformable, el qual permet analitzar el model des d'un punt de vista semàntic.

### *Model de tor empíric deformable*

Així doncs 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 arestes \_OUT>=2 i amb col·lididors-fulla (in>=2, 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 HORITZONTAL*

El model en tor es plega funcionalment primer (encara no pas causalment) pels laterals de cada nivell: cada nivell genera un anell funcional horitzontal (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, altrament 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.

## Model toroïdal acíclic amb retrofluxos

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

Figura 28

```
# 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 toroïdals
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
## 67

plot3d(coords3Dn_diag_corr1_tor, col = cols, size = 1.2, type = "s", box = FALSE)
axes3d(col = "gray60")

# arestes direcccionades 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="+")

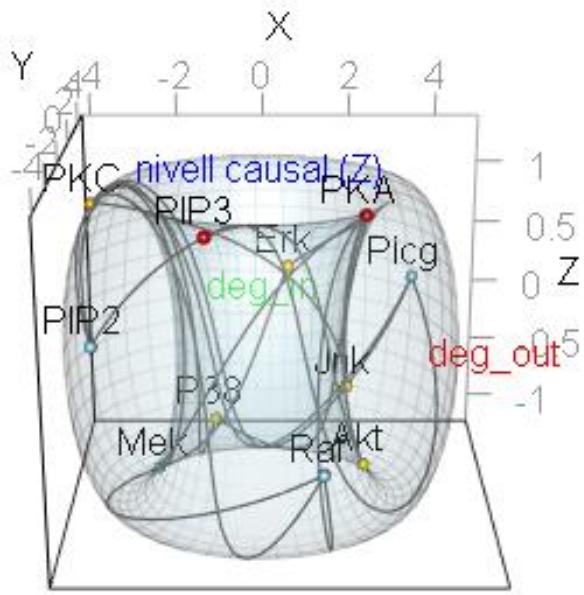
x_surf <- (R + r * cos(V)) * cos(U)
y_surf <- (R + r * cos(V)) * sin(U)
z_surf <- r * sin(v)

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

```



### ANNEX3 CODI R COMENTAT SOBRE LA VALIDACIÓ DEL DAG\_combinatori: PER MITJÀ D'APRENENTATGE ESTADÍSTIC O MACHINE LEARNING

Hom carrega els paquets calguts

```
library(knitr)
## Warning: package 'knitr' was built under R version 4.4.3
library(rmarkdown)
## Warning: package 'rmarkdown' was built under R version 4.4.2
library(dagitty)
library(e1071)
## Warning: package 'e1071' was built under R version 4.4.2
##
## Attaching package: 'e1071'
```

```

## The following object is masked from 'package:pracma':
##
##     sigmoid

library(coin)

## Warning: package 'coin' was built under R version 4.4.3

## Loading required package: survival

## Warning: package 'survival' was built under R version 4.4.2

##
## Attaching package: 'coin'

## The following object is masked from 'package:pracma':
##
##     size

## The following object is masked from 'package:scales':
##
##     pvalue

#install.packages("reticulate")

library(reticulate)

## Warning: package 'reticulate' was built under R version 4.4.3

#install.packages("tensorflow")

#reticulate::install_miniconda()
#reticulate::conda_create("r-tf", packages = "python=3.10")
#reticulate::conda_create("r-tf", packages = c("tensorflow", "keras"), pip=TRUE)
#use_condaenv("r-tf", required = TRUE)

library(tensorflow)

## Warning: package 'tensorflow' was built under R version 4.4.3

#install_tensorflow(version = "2.10.0", method = "conda", envname = "r-reticulate")

import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score, confusion_matrix, accuracy_score
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Input, Concatenate
from tensorflow.keras.optimizers import RMSprop, Adam

```

```
from tensorflow.keras.losses import BinaryCrossentropy
from tensorflow.keras.metrics import Accuracy
```

Hom defineix l'estructura del model convencionalment acceptada amb dagitty.

```
library(dagitty)
model <- dagitty(
  'dag {
    Akt [pos="-0.188,0.071"]
    Erk [pos="-0.128,0.065"]
    Jnk [pos="-0.184,-0.012"]
    Mek [pos="-0.106,-0.010"]
    P38 [pos="-0.170,-0.105"]
    PIP2 [pos="-0.245,-0.019"]
    PIP3 [pos="-0.210,-0.157"]
    PKA [pos="-0.128,-0.155"]
    PKC [pos="-0.086,-0.260"]
    Plcg [pos="-0.245,-0.077"]
    Raf [pos="-0.097,-0.087"]
    Mek -> Erk
    PIP2 -> PKC [pos="-0.485,-0.379"]
    PIP3 -> Akt
    PIP3 -> PIP2
    PIP3 -> Plcg
    PKA -> Akt
    PKA -> Erk
    PKA -> Jnk
    PKA -> Mek
    PKA -> P38
    PKA -> Raf
    PKC -> Jnk [pos="-0.188,-0.258"]
    PKC -> Mek [pos="-0.021,-0.245"]
    PKC -> P38 [pos="-0.166,-0.227"]
    PKC -> Raf
    Plcg -> PIP2
    Plcg -> PKC [pos="-0.248,-0.271"]
    Raf -> Mek
  }'
)
```

Hom representa gràficament l'estructura del model

Hom carregueu l'aplec de dades

```
data <- read.csv("C:/Users/grau/Desktop/MÀSTER BIOinfoestat/MACHlern/REPTE4/unitat
8/2020-dagitty-manual-master/2020-dagitty-manual-master/protocol3/protein_signal.cs
v")
```

Hom assaja totes les independències condicionals implícites del model amb el conjunt de dades

```

res <- localTests(x = model, data = data, type = "cis")
print(res)

##                                     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

```

Hom assaja les independències condicionals amb un màxim de 2 variables condicionants.

```

res <- localTests(x = model,
                    data = data,
                    max.conditioning.variables = 2)
print(res)

##                                         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 | PIP3, PKA    0.8979297031  0.000000e+00  0.964875186
## Akt _||_ Jnk | PKA, PKC    -0.0571097695  6.600962e-10 -0.075175525
## Akt _||_ Jnk | PIP3, PKA    -0.0624895340  1.393854e-11 -0.080541934
## Akt _||_ Mek | PKA, PKC    0.0648078273  2.383684e-12  0.046719287
## Akt _||_ Mek | PIP3, PKA    0.0617998837  2.328828e-11  0.043705410
## Akt _||_ P38 | PKA, PKC    0.2137041498  1.508917e-121 0.196341569
## 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 | 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 | PIP3, PKA    0.9764490934
## Akt _||_ Jnk | PKA, PKC    -0.0390067031
## Akt _||_ Jnk | PIP3, PKA   -0.0443963911
## Akt _||_ Mek | PKA, PKC    0.0828541553
## Akt _||_ Mek | PIP3, PKA    0.0798540533
## Akt _||_ P38 | PKA, PKC    0.2309967871
## 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 | 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

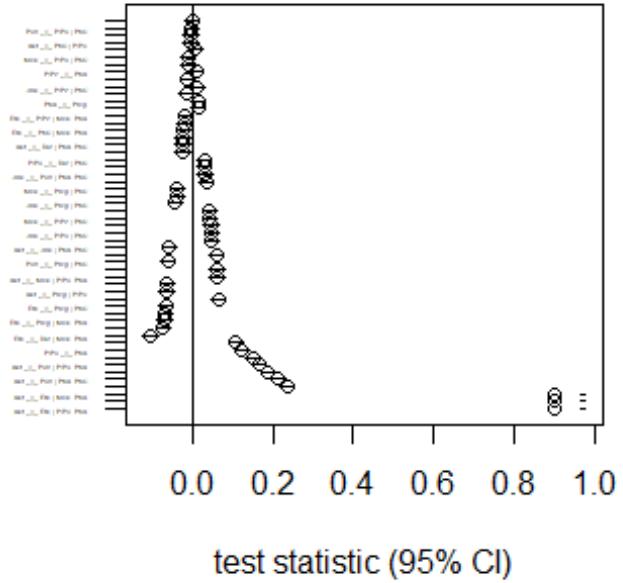
```

Hom representà gràficament els resultats de les proves més problemàtics

```

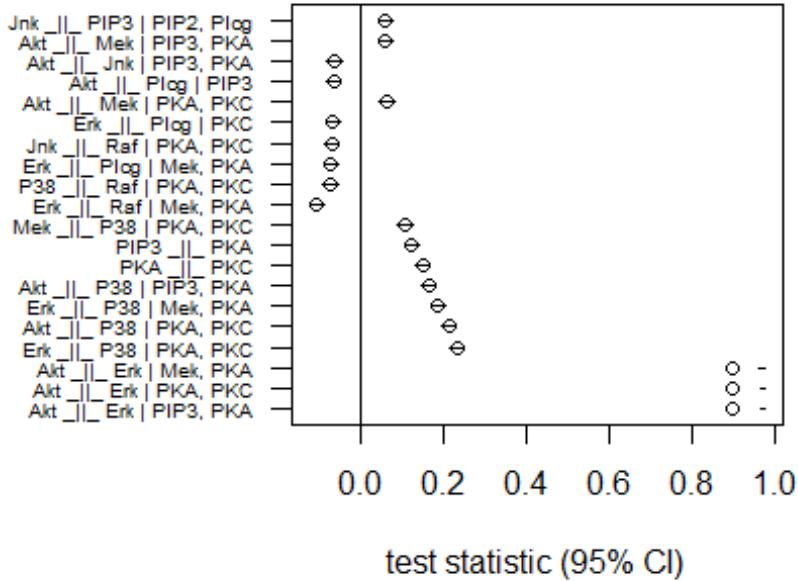
res <- res[order(abs(res$estimate)),]
plotLocalTestResults(res, axis.pars = list(cex.axis=0.2, las=1))

```



Hom representa gràficament els 20 resultats de les proves més problemàtics

```
res <- res[order(abs(res$estimate)),]  
plotLocalTestResults(tail(res, 20), axis.pars = list(cex.axis=0.6, las=1))
```



## VALIDACIÓ LOCAL

## CSV com a validació empírica complementària d'un DAG

La validació empírica del DAG és un dels moments crítics de la inferència causal, és per això, que el machine learning per fer molt més robust aquesta fase del pipeline.

L'algoritme més robust en termes euclidians del machine learning és el suport vector machine; hom pot aprofitar una eina que —malgrat tenir un altre usatge— pot esser de molta d'utilitat en termes de validació del DAG causal.

## l'antecedent: DAGSVM

DAGSVM que vol dir “Directed Acyclic Graph SVM” proposat per John Platt et al. a l’any 2000 com una millora d’un contra un (Platt, Cristianini i Shawe-Taylor, 2000,

Primer cal descriure la idea que hi ha darrere de DAGSVM: hom empra el mateix entrenament que l'u contra u, però accelerant els assaigs per mitjà d'un DAG. Per exemple, si hom té 4 classes A,B,c i D i sis classificadors entrenats cadascun en un parell de classes: (A,B);(A,C);(A,D);(B,C);(B,D) i (C,D).

=>Hom fa servir el primer classificador (A,D) i prediu la classe A (que és el mateix que predir que no és la classe D)

=>Hom fa servir el segon classificador (A,C) i també prediu la classe A (que és el mateix que predir que no és la classe C)

LA QUAL COSA vol dir que hom pot ignorar els classificadors (B,C),(B,D) o (C,D) perquè hom ja sap que la classe no és ni C ni D.

==> El darrer classificador útil és (A,B), el qual és assignat al punt de dades si prediu B (ÉS A DIR l'etiqueta de classificació final del punt de dades és B)

Però aquí hom no pot anar tan lluny, perquè implicaria categoritzar les variables contínues (per exemple, agrupar Akt com a activat segons un llindar biològic o sense supervisió tot agrupant per quantils...), construir tots els classificadors binaris per parelles de classes (aplicar K-means per a fer agrupacions) i estructurar-los en termes de DAG (aplicar la funció DAGSVM per aplicar-hi la lògica de classificació seqüencial). Aquest enfocament suposaria transformar el model causal original en un sistema de classificació jeràrquica tot incloent assumptions (llindars, separabilitat, ordre de decisió) que al seu torn també cal validar.

Per això aquí hom es limita a la seva funció com a eina empírica a fi de validar les dependències direccionals proposades pel DAG, sense imposar cap discretització ni estructura classificadora artificial.

Una proposta d'usatge com a eina de validació empírica de relacions causals suggerides o rebutjades fóra entrenar un model SVM. Hom assaja primer un cas particular, per després, automatitzar-lo a tots els tests locals.

Aquest cas sembla prou evident que hi ha dependència residual entre Akt i Erk; hom pot entrenar un SVM per predir Erk a partir d'Akt, PKA i PKC i un segon SVM per predir també Erk a partir de PKA i PKC però sense Akt. Si afegint Akt millora significativament la predicció d'Erk hi ha evidència empírica que el lligam entre Akt i Erk no és espuri.

```
res[1,]

##                               estimate      p.value      2.5%      97.5%
## Erk _||_ PIP3 | Mek, PKA 0.0007349409 0.9367273 -0.01740878 0.01887818

svm11 <- svm(Erk~PKA+PKC, data=data, type="eps-regression")
summary(svm11)

##
## Call:
## svm(formula = Erk ~ PKA + PKC, data = data, type = "eps-regression")
##
##
## Parameters:
##   SVM-Type:  eps-regression
##   SVM-Kernel: radial
##     cost: 1
##     gamma: 0.5
##   epsilon: 0.1
##
##
## Number of Support Vectors:  10699

pred11 <- predict(svm11)
```

```

svm22 <- svm(Erk~Akt+PKA+PKC, data=data, type="eps-regression")
summary(svm22)

##
## Call:
## svm(formula = Erk ~ Akt + PKA + PKC, data = data, type = "eps-regression")
##
##
## Parameters:
##   SVM-Type:  eps-regression
##   SVM-Kernel: radial
##   cost: 1
##   gamma: 0.3333333
##   epsilon: 0.1
##
##
## Number of Support Vectors:  8512

pred22 <- predict(svm22)

```

Atès que és un mètode non-paramètric, hom aplica un test també non-paramètric per tal de veure si la diferència és empíricament significativa

```

comparat <- function(obs, pred1, pred2){
  residu1 <- abs(obs-pred1)
  residu2 <- abs(obs-pred2)
  #print(residu2)
  diferencies <- residu1-residu2
  diferencies <- diferencies[!is.na(diferencies)]

  p_val <- wilcox.test(diferencies, mu=0, alternative="greater")$p.value
  return(list(
    RMSE_model1 <- sqrt(mean((obs-pred1)^2)),
    RMSE_model2 <- sqrt(mean((obs-pred2)^2)),
    p_valua = p_val
  ))
}

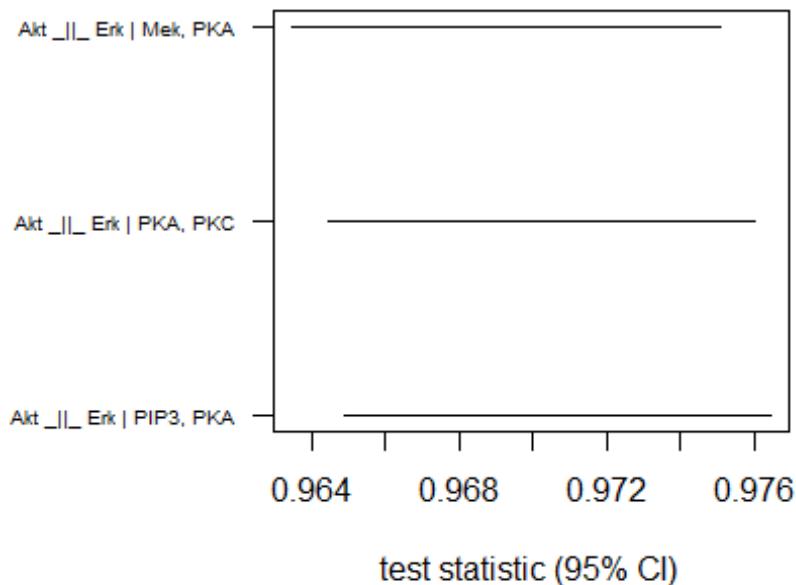
comparat(data$Erk, pred11, pred22)

## [[1]]
## [1] 0.5000091
##
## [[2]]
## [1] 0.2051589
##
## $p_valua
## [1] 0

```

Funció per tal d'automatitzar el procés a totes les independències (condicionades) estructurals llistades a l'objecte res; hom n'agafa sols tres, les més extremes, per a fer la comprovació (però hom pot ja verificar tots els tests locals per mitjà de SVM). I on hi hagi divergència de resultats, entre els tests i els outputs de SVM, s'obre un nou angle d'anàlisi perquè ja no és clar si el DAG té una dependència espúria o bé hi ha un error de validació.

```
plotLocalTestResults(tail(res, 3), axis.pars = list(cex.axis=0.6, las=1))
```



```
(res333 <- tail(res, 3))

##                                     estimate p.value      2.5%    97.5%
## Akt _||_ Erk | Mek, PKA 0.8970852     0 0.9634638 0.9750847
## Akt _||_ Erk | PKA, PKC 0.8976678     0 0.9644368 0.9760253
## Akt _||_ Erk | PIP3, PKA 0.8979297     0 0.9648752 0.9764491
```

Hom referma els resultats dels tests locals amb els resultats de SVM, ara hom sap que el DAG té problemes estructurals que cal corregir. Ara hom suposa que la validació empírica és més robusta perquè mètodes diferents convergeixen al mateix resultat.

```
analitza_LocalTest <- function(res, data){
  resultats <- list()

  for (i in seq_len(nrow(res))) {
    cond_caracter <- rownames(res)[i]
    parts <- strsplit(cond_caracter, " \| ")[[1]]
```

```

X <- trimws(strsplit(parts[1], "\\\\|\\\\|_")[[1]][1])
Y <- trimws(strsplit(parts[1], "\\\\|\\\\|_")[[1]][2])
Z <- if (length(parts)==2) unlist(strsplit(parts[2], ",\\s*")) else character(0)
}

if(length(Z)==0){
  formula1 <- as.formula(paste(Y,"~1"))
  formula2 <- as.formula(paste(Y,"~",X)))
} else {
  formula1 <- as.formula(paste(Y,"~", paste(Z,collapse="+"))))
  formula2 <- as.formula(paste(Y,"~", paste(c(X,Z),collapse="+")))}
#print(formula1)
#print(formula2)

svm1 <- svm(formula1, data=data, type="eps-regression")
svm2 <- svm(formula2, data=data, type="eps-regression")
#print(svm1)
#print(svm2)

pred1 <- predict(svm1)
pred2 <- predict(svm2)
#print(pred1)
#print(pred2)

obs<-data[[Y]]
#print(obs)

res_comp <- comparat(obs,pred1, pred2)
#print(res_comp)
resultats[[cond_caracter]] <- res_comp

}

return(resultats)
}

analitza_LocalTest(res333, data)

## $`Akt _||_ Erk | Mek, PKA`
## $`Akt _||_ Erk | Mek, PKA`[[1]]
## [1] 0.5004886
##
## $`Akt _||_ Erk | Mek, PKA`[[2]]
## [1] 0.207153

```

```

## 
## $`Akt _||_ Erk | Mek, PKA`$p_valua
## [1] 0
##
## 
## $`Akt _||_ Erk | PKA, PKC`
## $`Akt _||_ Erk | PKA, PKC`[[1]]
## [1] 0.5000091
##
## 
## $`Akt _||_ Erk | PKA, PKC`[[2]]
## [1] 0.2051589
##
## 
## $`Akt _||_ Erk | PKA, PKC`$p_valua
## [1] 0
##
## 
## 
## $`Akt _||_ Erk | PIP3, PKA`
## $`Akt _||_ Erk | PIP3, PKA`[[1]]
## [1] 0.5040106
##
## 
## $`Akt _||_ Erk | PIP3, PKA`[[2]]
## [1] 0.2075392
##
## 
## $`Akt _||_ Erk | PIP3, PKA`$p_valua
## [1] 0

resultats <- list()
cond_caracter <- rownames(res)[33]
  parts <- strsplit(cond_caracter, " \| ")[[1]]

  X <- trimws(strsplit(parts[1], "_\|_\|_")[[1]][1])
  Y <- trimws(strsplit(parts[1], "_\|_\|_")[[1]][2])
  Z <- if (length(parts)==2) unlist(strsplit(parts[2], ",\\s*")) else character(0)
)

  if(length(Z)==0){
    formula1 <- as.formula(paste(Y,"~1"))
    formula2 <- as.formula(paste(Y,"~",X)))} else {
    formula1 <- as.formula(paste(Y,"~", paste(Z,collapse="+"))))
    formula2 <- as.formula(paste(Y,"~", paste(c(X,Z),collapse="+))))}
  print(formula1)

## Raf ~ PKC
  print(formula2)

## Raf ~ Plcg + PKC

svm1 <- svm(formula1, data=data, type="eps-regression")
svm2 <- svm(formula2, data=data, type="eps-regression")

```

```

print(svm1)

##
## Call:
## svm(formula = formula1, data = data, type = "eps-regression")
##
##
## Parameters:
##   SVM-Type: eps-regression
##   SVM-Kernel: radial
##     cost: 1
##     gamma: 1
##   epsilon: 0.1
##
##
## Number of Support Vectors: 10411

print(svm2)

##
## Call:
## svm(formula = formula2, data = data, type = "eps-regression")
##
##
## Parameters:
##   SVM-Type: eps-regression
##   SVM-Kernel: radial
##     cost: 1
##     gamma: 0.5
##   epsilon: 0.1
##
##
## Number of Support Vectors: 10433

pred1 <- predict(svm1)
pred2 <- predict(svm2)
print(head(pred1))

##      1       2       3       4       5       6
## 3.609330 3.575898 3.586053 3.597651 3.556352 3.597651

print(head(pred2))

##      1       2       3       4       5       6
## 3.585283 3.576212 3.602365 3.617790 3.521569 3.620689

obs<-data[[Y]]
print(head(obs))

## [1] 3.179665 3.313155 3.531847 3.621384 3.285691 3.032219

```

```

res_comp <- comparat(obs,pred1, pred2)
print(res_comp)

## [[1]]
## [1] 0.3783664
##
## [[2]]
## [1] 0.3767922
##
## $p_valua
## [1] 2.926867e-06

resultats[[cond_caracter]] <- res_comp

```

## VALIDACIÓ GLOBAL: CNN com a validació empírica complementària d'un DAG

Fins ara hem validat cada test local del graf causal amb SVM, hom opera a trams; ara hom podria avaluar tot el DAG. El DAG suposa que té dos nodes sortints i dos d'entrants i la resta de nodes són intermediaris.

**ENTRANTS:** són els qui no tenen cap aresta entrant, són exògenes i són les dues neurones d'entrada

**MITJANCERS:** hom crea una neurona per a cada node, les capes paral·leles han d'esser connectades parcialment per tal de representar la modularitat del DAG, cosa que es tradueix en codificar les connexions manualment segons l'estructura del DAG (hom també pot assajar de fer una matriu de pesos de 0 i 1). El que és important és evitar crear camins nous que no siguin causals.

**SORTINTS:** S'és vist que hi ha dependències entre les dues variables sortints, però l'estructura del DAG és la que és i hom procedeix sense tenir en compte els resultats previs. Per tant, caldria fer una xarxa neuronal de sortida múltiple (cal no barrejar les neurones de sortida) amb regressió multivariant.

Hom compta amb 2 nodes d'entrada, 2 nodes de sortida i la resta de nodes medials; altrament dit 11 nodes i 18 arestes. Cal bastir una xarxa neuronal multivariant amb arquitectura definida pel DAG.

MODEL CNN D'ESTRUCTURA DAG hom agrupa els nodes entrants a cada node

```

#nodes exògens
pip3 = Input(shape=(1,), name='PIP3_input')
pka = Input(shape=(1,), name='PKA_input')

#nodes mitjancers que sols reben una aresta
# PIP3 ---> Plcg
plcg = Dense(1, activation='linear', name='Plcg')(pip3)
# Plcg ---> PIP2
pip2 = Dense(1, activation='linear', name='PIP2')(plcg)

#nodes mitjancers que reben dues o més arestes
#PIP2 + Plcg ---> PKC
pkc_input = Concatenate()([pip2,plcg])
pkc = Dense(1, activation='linear', name='PKC')(pkc_input)

```

```

#PKA + PKC ---> Raf
raf_input = Concatenate()([pka,pkc])
raf = Dense(1, activation='linear', name='Raf')(raf_input)

#PKA + PKC ---> P38
jnk_input = Concatenate()([pka,pkc])
jnk = Dense(1, activation='linear', name='Jnk')(jnk_input)

#PKA + PKC ---> Jnk
p38_input = Concatenate()([pka,pkc])
p38 = Dense(1, activation='linear', name='P38')(p38_input)

#PIP3 + PKA ---> Akt
akt_input = Concatenate()([pip3,pka])
akt = Dense(1, activation='linear', name='Akt')(akt_input)

#PKA + PKC + Raf ---> Mek
mek_input = Concatenate()([pka,pkc,raf])
mek = Dense(1, activation='linear', name='Mek')(mek_input)

#Mek + PKA ---> Erk
erk_input = Concatenate()([mek,pka])
erk = Dense(1, activation='linear', name='Erk')(erk_input)

model1 = Model(inputs=[pip3,pka], outputs=[erk,akt])
model1.summary()

```

## Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
PIP3_input (InputLayer)	(None, 1)	0	-
Plcg (Dense)	(None, 1)	2	PIP3_input[0][0]
PIP2 (Dense)	(None, 1)	2	Plcg[0][0]
concatenate (Concatenate)	(None, 2)	0	PIP2[0][0], Plcg[0][0]
PKA_input (InputLayer)	(None, 1)	0	-
PKC (Dense)	(None, 1)	3	concatenate[0][0]
concatenate_1 (Concatenate)	(None, 2)	0	PKA_input[0][0], PKC[0][0]

```

##          Raf (Dense)      (None, 1)      3  concatenate_1[0]...
##          concatenate_5      (None, 3)      0  PKA_input[0][0],
##          (Concatenate)          PKC[0][0],
##          Raf[0][0]
##          Mek (Dense)      (None, 1)      4  concatenate_5[0]...
##          concatenate_6      (None, 2)      0  Mek[0][0],
##          (Concatenate)          PKA_input[0][0]
##          concatenate_4      (None, 2)      0  PIP3_input[0][0],
##          (Concatenate)          PKA_input[0][0]
##          Erk (Dense)      (None, 1)      3  concatenate_6[0]...
##          Akt (Dense)      (None, 1)      3  concatenate_4[0]...
##          Total params: 20 (80.00 B)
##          Trainable params: 20 (80.00 B)
##          Non-trainable params: 0 (0.00 B)

```

MODEL DEL TOT CONNECTAT CAP A LES SORTIDES (HOM IGNORA LA XARXA INTERNA): model saturat sols respecte dels nodes sortida (tots els nodes connectats amb els nodes sortida)

```

from tensorflow.keras import backend as k
k.clear_session()
#nodes exògens
pip3 = Input(shape=(1,), name='PIP3_input')
pka = Input(shape=(1,), name='PKA_input')

#capa d'agregació inicial
x = Concatenate()([pip3, pka])
#nodes mitjancers
plcg = Dense(4, activation='relu', name='Plcg')(x)
pip2 = Dense(4, activation='relu', name='PIP2')(plcg)
pkc = Dense(4, activation='relu', name='PKC')(pip2)
raf = Dense(4, activation='relu', name='Raf')(pkc)
mek = Dense(4, activation='relu', name='Mek')(raf)
p38 = Dense(4, activation='relu', name='P38')(mek)
jnk = Dense(4, activation='relu', name='Jnk')(p38)
#concatenació de totes les capes ocultes

concat_tot = Concatenate()([plcg,pip2,pkc,raf,mek,p38,jnk])

#sortides fully connected
akt = Dense(1, activation='linear', name='Akt')(concat_tot)
erk = Dense(1, activation='linear', name='Erk')(concat_tot)

```

```

##model final
model2 = Model(inputs=[pip3,pka], outputs=[erk,akt])
model2.summary()

## Model: "functional"
##

| Layer (type)                        | Output Shape | Param # | Connected to                                                                                    |
|-------------------------------------|--------------|---------|-------------------------------------------------------------------------------------------------|
| PIP3_input<br>(InputLayer)          | (None, 1)    | 0       | -                                                                                               |
| PKA_input<br>(InputLayer)           | (None, 1)    | 0       | -                                                                                               |
| concatenate<br>(Concatenate)        | (None, 2)    | 0       | PIP3_input[0][0],<br>PKA_input[0][0]                                                            |
| Plcg (Dense)                        | (None, 4)    | 12      | concatenate[0][0]                                                                               |
| PIP2 (Dense)                        | (None, 4)    | 20      | Plcg[0][0]                                                                                      |
| PKC (Dense)                         | (None, 4)    | 20      | PIP2[0][0]                                                                                      |
| Raf (Dense)                         | (None, 4)    | 20      | PKC[0][0]                                                                                       |
| Mek (Dense)                         | (None, 4)    | 20      | Raf[0][0]                                                                                       |
| P38 (Dense)                         | (None, 4)    | 20      | Mek[0][0]                                                                                       |
| Jnk (Dense)                         | (None, 4)    | 20      | P38[0][0]                                                                                       |
| concatenate_1<br>(Concatenate)      | (None, 28)   | 0       | Plcg[0][0],<br>PIP2[0][0],<br>PKC[0][0],<br>Raf[0][0],<br>Mek[0][0],<br>P38[0][0],<br>Jnk[0][0] |
| Erk (Dense)                         | (None, 1)    | 29      | concatenate_1[0]...                                                                             |
| Akt (Dense)                         | (None, 1)    | 29      | concatenate_1[0]...                                                                             |
| ## Total params: 190 (760.00 B)     |              |         |                                                                                                 |
| ## Trainable params: 190 (760.00 B) |              |         |                                                                                                 |
| ## Non-trainable params: 0 (0.00 B) |              |         |                                                                                                 |


```

La inclusió explícita de l'aresta Erk → Akt en una xarxa neuronal constrenyida pel DAG permet avaluar si aquesta relació millora la coherència empírica global sense introduir camins causals addicionals, mantenint intacta l'estructura del model.

```
#nodes exògens
pip3 = Input(shape=(1,), name='PIP3_input')
pka = Input(shape=(1,), name='PKA_input')

#nodes mitjancers amb una sola aresta
plcg = Dense(1, activation='linear', name='Plcg')(pip3)
pip2 = Dense(1, activation='linear', name='PIP2')(plcg)

#nodes amb múltiples arestes
pkc_input = Concatenate()([pip2, plcg])
pkc = Dense(1, activation='linear', name='PKC')(pkc_input)

raf_input = Concatenate()([pka, pkc])
raf = Dense(1, activation='linear', name='Raf')(raf_input)

jnk_input = Concatenate()([pka, pkc])
jnk = Dense(1, activation='linear', name='Jnk')(jnk_input)

p38_input = Concatenate()([pka, pkc])
p38 = Dense(1, activation='linear', name='P38')(p38_input)

mek_input = Concatenate()([pka, pkc, raf])
mek = Dense(1, activation='linear', name='Mek')(mek_input)

#Mek + PKA --> Erk
erk_input = Concatenate()([mek, pka])
erk = Dense(1, activation='linear', name='Erk')(erk_input)

#CANVI ÚNIC: PIP3+ PKA + Erk --> Akt
akt_input = Concatenate()([pip3, pka, erk])
akt = Dense(1, activation='linear', name='Akt')(akt_input)

model3 = Model(inputs=[pip3, pka], outputs=[erk, akt])
model3.summary()

## Model: "functional_1"
##
```

Layer (type)	Output Shape	Param #	Connected to
PIP3_input (InputLayer)	(None, 1)	0	-
Plcg (Dense)	(None, 1)	2	PIP3_input[0][0]

```

## PIP2 (Dense)      (None, 1)      2  Plcg[0][0]
## concatenate_2     (None, 2)      0  PIP2[0][0],
## (Concatenate)          Plcg[0][0]
##
## PKA_input        (InputLayer)   (None, 1)      0  -
##
## PKC (Dense)       (None, 1)      3  concatenate_2[0]...
##
## concatenate_3     (None, 2)      0  PKA_input[0][0],
## (Concatenate)          PKC[0][0]
##
## Raf (Dense)       (None, 1)      3  concatenate_3[0]...
##
## concatenate_6     (None, 3)      0  PKA_input[0][0],
## (Concatenate)          PKC[0][0],
##          Raf[0][0]
##
## Mek (Dense)       (None, 1)      4  concatenate_6[0]...
##
## concatenate_7     (None, 2)      0  Mek[0][0],
## (Concatenate)          PKA_input[0][0]
##
## Erk (Dense)       (None, 1)      3  concatenate_7[0]...
##
## concatenate_8     (None, 3)      0  PIP3_input[0][0],
## (Concatenate)          PKA_input[0][0],
##          Erk[0][0]
##
## Akt (Dense)       (None, 1)      4  concatenate_8[0]...
##
## Total params: 21 (84.00 B)
## Trainable params: 21 (84.00 B)
## Non-trainable params: 0 (0.00 B)

import pandas as pd
import numpy as np

data1 = pd.read_csv("C:/Users/graup/Desktop/MÀSTER BIOinfoestat/MACHlern/REPTE4/unitat8/2020-dagitty-manual-master/2020-dagitty-manual-master/protocol3/protein_signal.csv")

```

La incorporació de l'aresta Erk -> Akt no degrada el model global i produeix una millora específica i localitzada en la prediccio d'Akt, sense afectar la resta de nodes. Això indica que l'aresta és compatible amb l'estruccura causal global i no introduceix incoherències funcionals.

```

# inputs
X_pip3 = data1["PIP3"].values.reshape(-1, 1)
X_pka = data1["PKA"].values.reshape(-1, 1)

```

```

# outputs
Y_erk = data1["Erk"].values.reshape(-1, 1)
Y_akt = data1["Akt"].values.reshape(-1, 1)

for m in [model1, model3]:
    m.compile(
        optimizer=Adam(learning_rate=0.01),
        loss=['mse', 'mse'],
        metrics=[[ 'mae'], [ 'mae']])
)

model1.fit([X_pip3, X_pka], [Y_erk, Y_akt],
epochs=200, validation_split=0.2, verbose=0)

## <keras.src.callbacks.history.History object at 0x000001D39B629550>

model3.fit([X_pip3, X_pka], [Y_erk, Y_akt],
epochs=200, validation_split=0.2, verbose=0)

## <keras.src.callbacks.history.History object at 0x000001D39B744CE0>

res1 = model1.evaluate([X_pip3, X_pka], [Y_erk, Y_akt], verbose=0)
res3 = model3.evaluate([X_pip3, X_pka], [Y_erk, Y_akt], verbose=0)

#model.save("cnn_model1.keras")
#model.save("cnn_model3.keras")

print("MODEL 1 (sense Erk -> Akt)")

## MODEL 1 (sense Erk -> Akt)

print(f"MSE Erk: {res1[1]:.4f} | MSE Akt: {res1[2]:.4f}")

## MSE Erk: 0.2661 | MSE Akt: 0.2162

print("MODEL 3 (amb Erk -> Akt)")

## MODEL 3 (amb Erk -> Akt)

print(f"MSE Erk: {res3[1]:.4f} | MSE Akt: {res3[2]:.4f}")

## MSE Erk: 0.2653 | MSE Akt: 0.2258

# res = [total_loss, erk_mse, akt_mse, erk_mae, akt_mae]

```