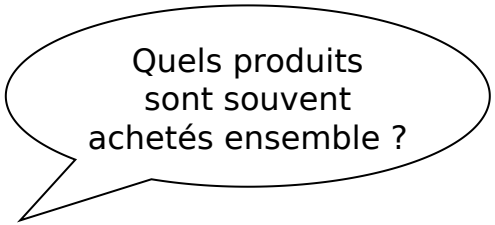
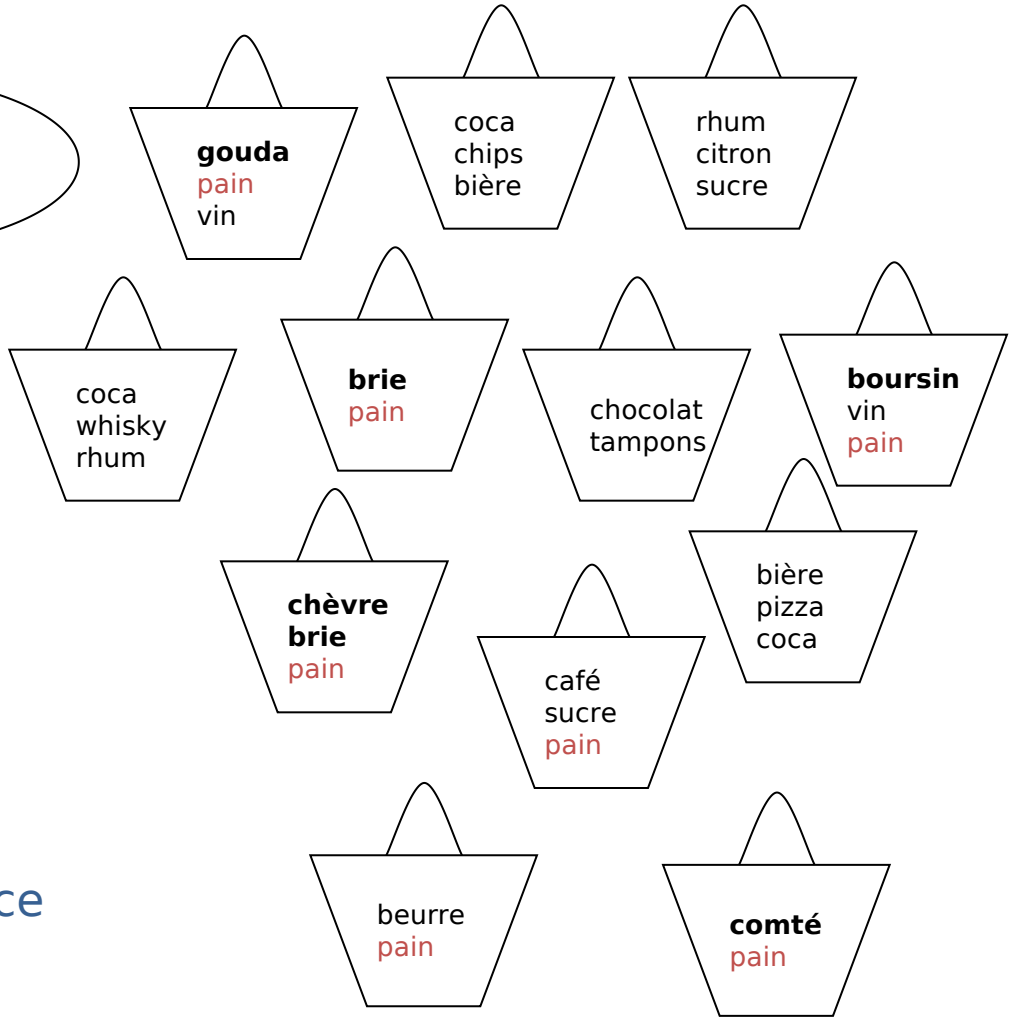
Règles d'associations





A ➡ B [mesures]

- Mesures : support et confiance
- Algorithme Apriori
- Algorithme FP-Trees
- Autres mesures

Recherche des associations

- Règles d'association :
  - motifs de la forme : Corps ➡ Tête
  - Exemple : achète(x, "fromage") ➡ achète(x, "pain")

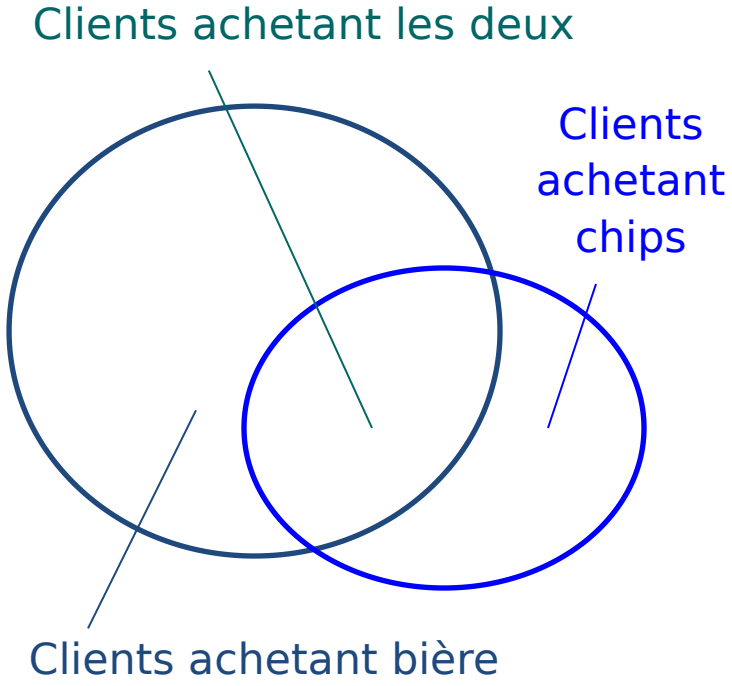
- Étant donnés :
  - une base de transactions D, $I = \{i_1, i_2, \ldots, i_n\}$
  - chaque transaction est décrite par un identifiant TID et une liste d'items $T_{TID} = \{i_1, i_2, \ldots, i_m\} \subseteq I$

  règle $A \Rightarrow B$ / $A \subset I$ et $B \subset I$ et $A \cap B = \emptyset$

  Trouver : toutes les règles qui expriment une corrélation entre la présence d'un ensemble d'items avec la présence d'un (autre) ensemble d'items

  Ex : 98% des personnes qui achètent des chips achètent de la bière

Mesures : support et confiance



Trouver les règles *X Y* ➡ *Z*
avec un support ≥ s et une confiance ≥ c

- support *s*, probabilité qu'une transaction contienne {X, Y, Z}
- confiance *c*, probabilité conditionnelle qu'une transaction qui contient {X, Y} contienne aussi *Z*

Confiance=support(X,Y,Z)/support(X,Y)

ID Transaction	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Soit support minimum 50%, et confiance minimum 50%,

A ➡ *C* (50%, 66.6%)

C ➡ *A* (50%, 100%)

1. Identifier les ensembles fréquents (dont le support ≥ *s*)

Transaction ID	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confiance 50%

Itemsets fréquents	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

2. Extraire les règles dont la confiance ≥ *c*

Pour $A \Rightarrow C$:

support = support($\{A, C\}$) = 50%

confiance = support($\{A, C\}$)/support($\{A\}$) = 66.6%

Puis pour $C \Rightarrow A$, ...

Remarque : en pratique, on considère souvent les *closed itemsets* ou ***maximal*** *itemsets*

Extraction des assiociations : algorithme Apriori

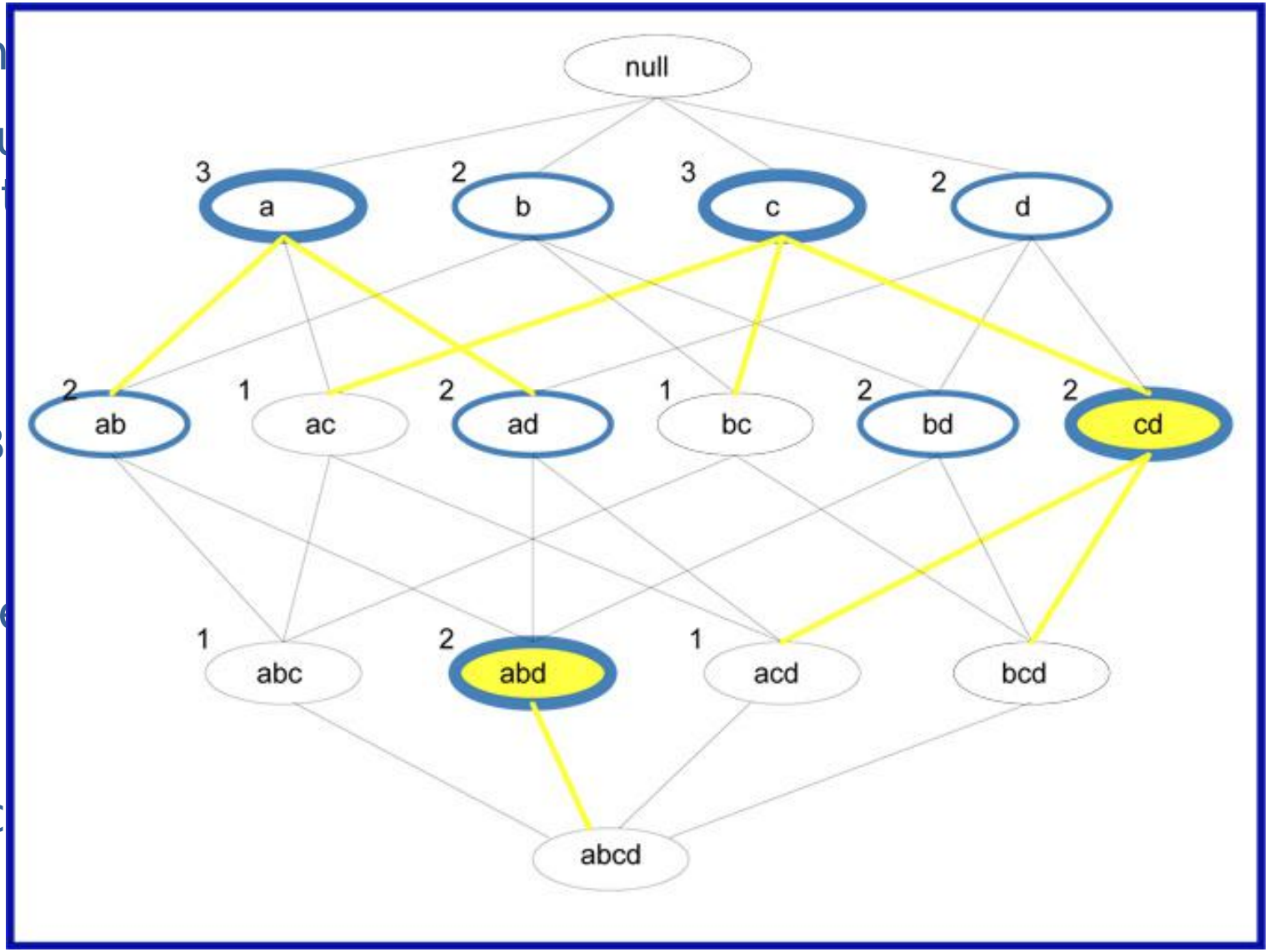
- Principe :

  Si un en

  alors tou
  fréquent

  - Si {A}
  - si {A,B

  - Itérative
    cardina
  - Utiliser
    d'assoc



Algorithme Apriori

- Étape de jointure: C_k est généré en joignant L_{k-1}avec lui même
- Étape d'élimination: Chaque (k-1)-itemset qui n'est pas fréquent ne peut être un sous ensemble d'un k-itemset fréquent

C_k: Itemset candidat de taille k,

L_k : itemset fréquent de taille k

L_1 = {items fréquents}

pour (k = 1; L_k !=∅; k++)

C_{k+1} = candidats générés à partir de L_k *% jointure*

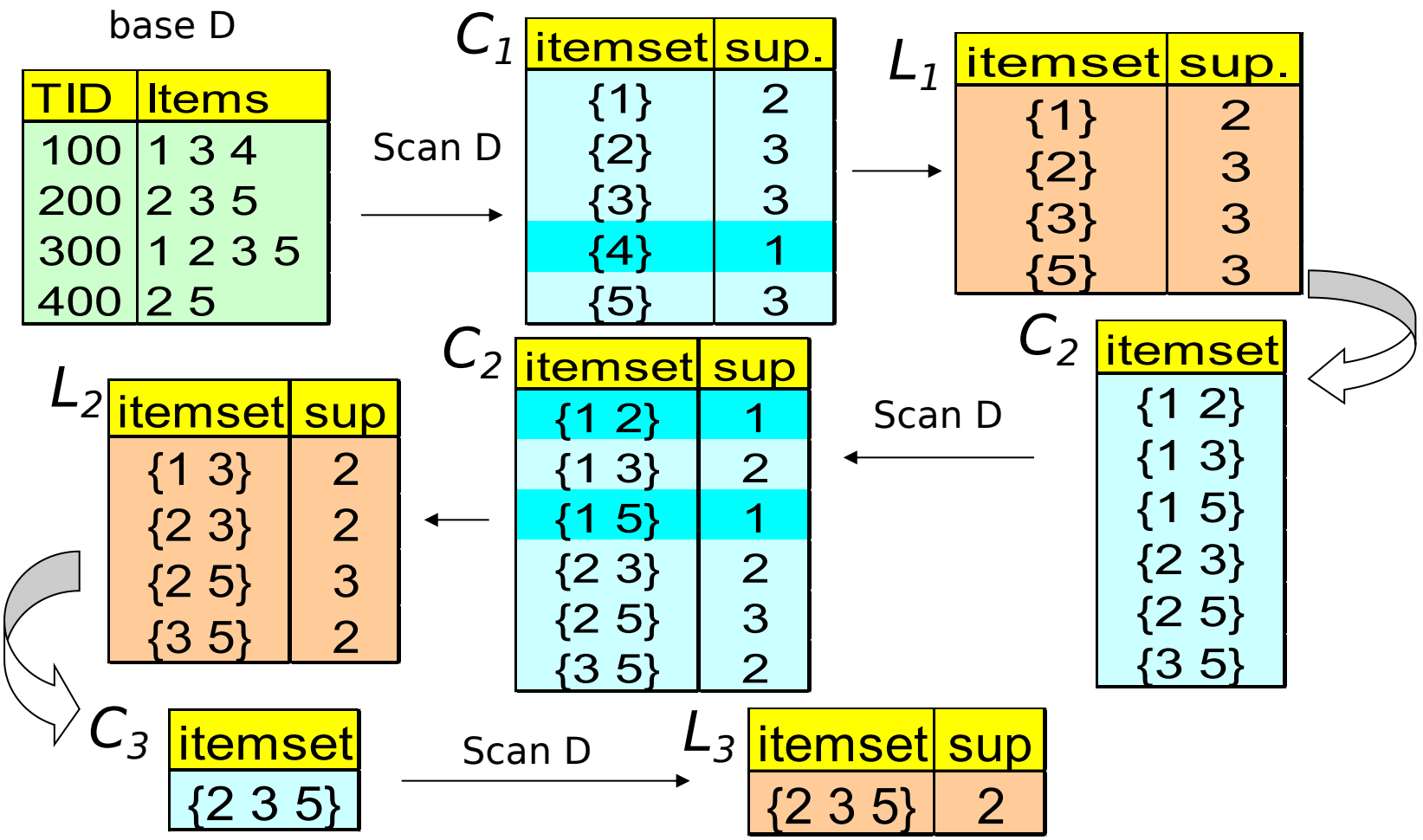
pour chaque transaction t dans la base

incrémenter le COUNT ***des candidats de C_{k+1} qui sont dans t***

L_{k+1} = candidats dans C_{k+1} dont COUNT > support_min

renvoyer $\cup_k L_k$

avec min_support=2



Extraction des règles

Confiance (A ➡ B)=support(AB)/support(A)

Algorithme :

pour chaque itemset fréquent f
  pour chaque sous ensemble s ⊂ f, avec s≠∅
    si confiance(s → f \ s) > min_conf
    alors afficher(s → f \ s)
  fin pour
fin pour

min_sup=50%
min_conf=75%

Items	count
1	2
2	3
3	3
5	3

Items	count
1, 3	2
2, 3	2
2, 5	3
3, 5	2
2, 3, 5	2

Règles	Conf.
1➡3	100%
3➡1	66%
2➡3	66%
3➡2	66%
2➡5	100%
5➡2	100%
3➡5	66%
5➡3	66%
2,3➡5	100%
2,5➡3	66%
3,5➡2	100%
2➡3,5	66%
3➡2,5	100%
5➡2,3	66%

Défauts d'Apriori

- Le principe de l'algorithme:
  - Utiliser les (k – 1)-itemsets fréquents pour générer les k-itemsets candidats
  - Scanner la base pour tester le support des candidats

- Point faible : génération des candidats
  - Beaucoup :
    - 10K 1-itemsets fréquents générant ~50M paires d'items candidates
    - Pour trouver les 100-itemsets, on doit générer $2^{100} \approx 10^{30}$ candidats.
  - Plusieurs scans de la base :
    - On doit faire (n +1) scans, pour trouver les n-itemsets fréquents

Exploration sans génération de candidats

- Compresser la base, Frequent-Pattern tree (FP-tree)
  - Une représentation condensée
  - Évite les scans coûteux de la base

- Développer une méthode efficace pour l'exploration basée sur une approche
  - diviser-pour-régner: décompose le problèmes en sous-problèmes
  - Pas de génération de candidats :

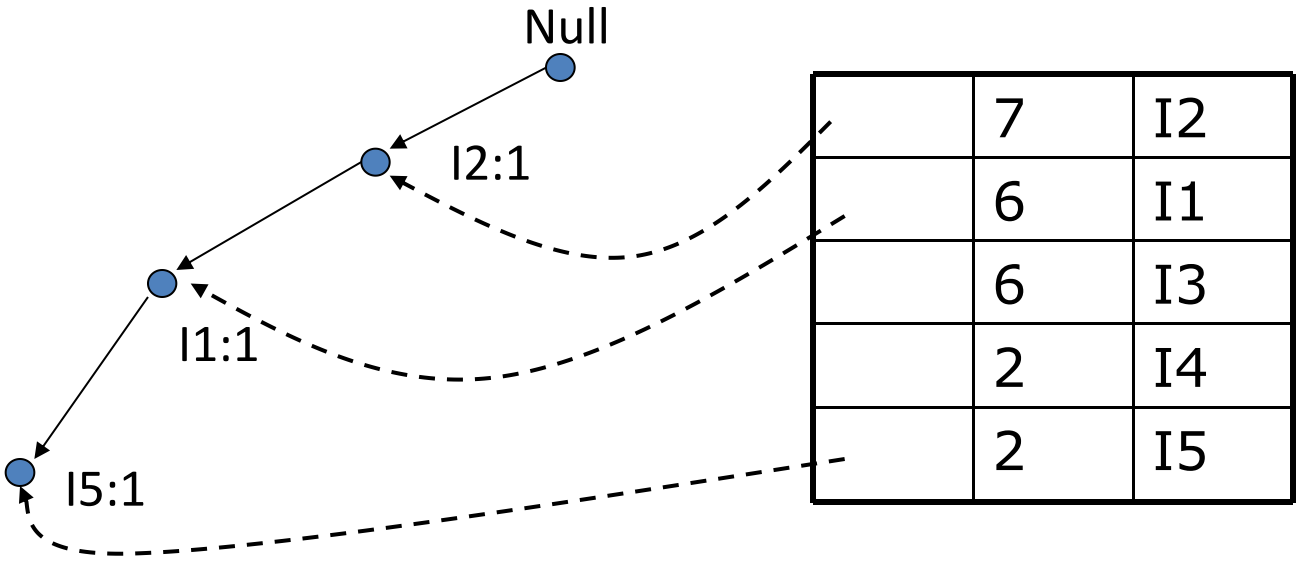
    test de la "sous-base" seulement

FP-Trees : exemple

TID	T100	T200	T300	T400	T500	T600	T700	T800	T900
Liste items	I1, I2, I5	I2, I4, I6	I1, I3	I1, I2, I4	I2, I3, I8	I2, I3	I1, I3, I7	I1,I2,I3,I5	I1, I2, I3

Supposons que min-support=2. On construit la liste « **triée** » :
L = [I2:7, I1:6, I3:6, I4:2, I5:2]

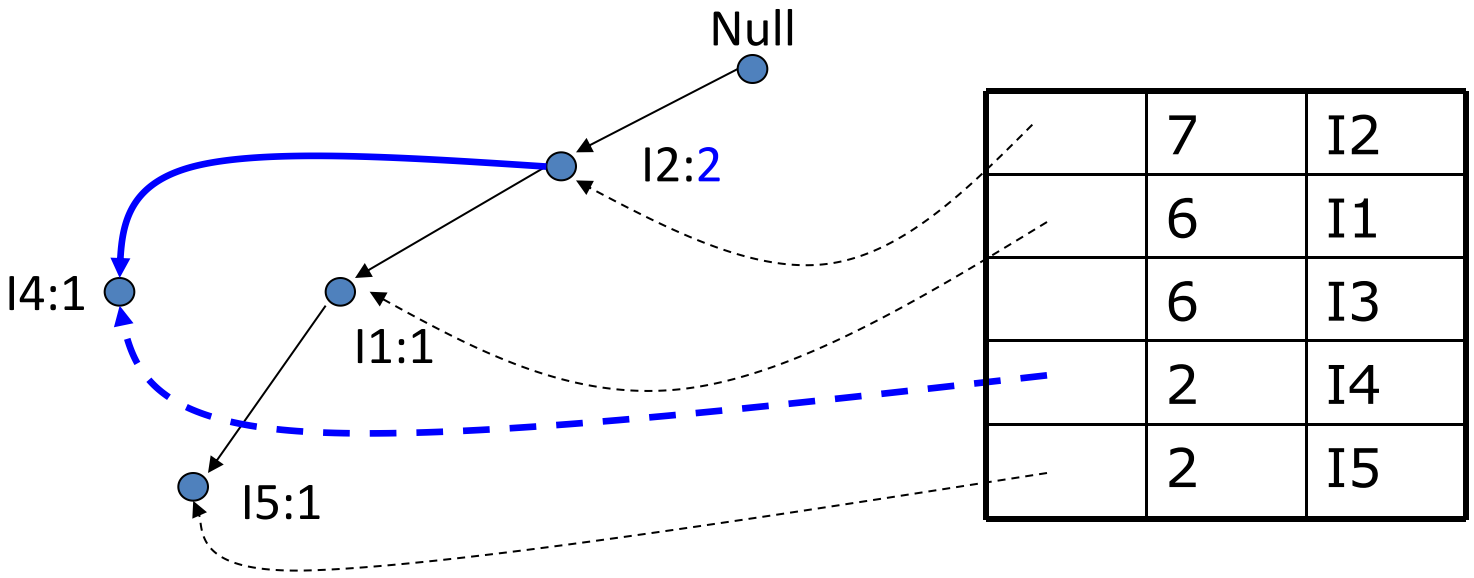
On parcourt une 2ème fois la base. On lit les transactions selon l'ordre des items dans L : pour T100 on a I2,I1,I5. La lecture de T100 donne



FP-Trees : exemple

TID	T100	T200	T300	T400	T500	T600	T700	T800	T900
Liste items	I1, I2, I5	I2, I4, I6	I1, I3	I1, I2, I4	I2, I3, I8	I2, I3	I1, I3, I7	I1,I2,I3,I5	I1, I2, I3

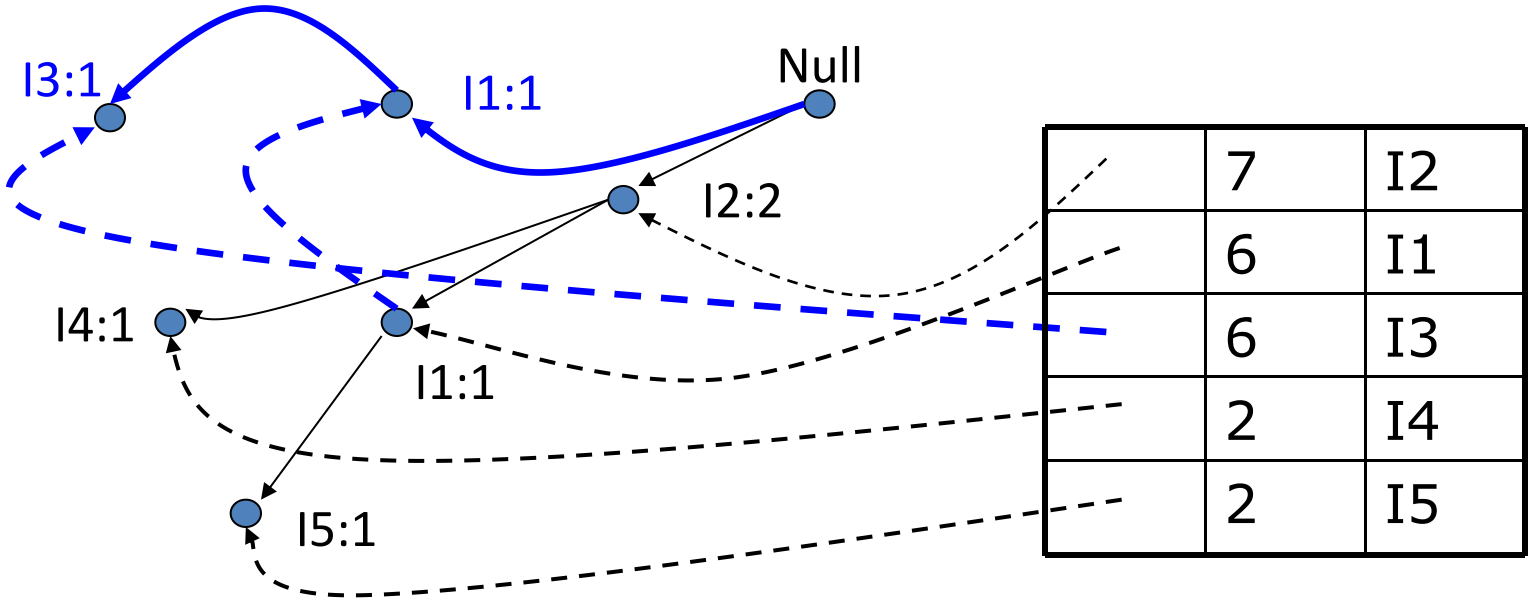
La lecture de T200 va *a priori* générer une branche qui relie la racine à I2 et I2 à I4. Or cette branche partage un préfixe (i.e I2) avec une branche qui existe déjà. L'arbre obtenu après lecture de T200 sera



FP-Trees : exemple

TID	T100	T200	T300	T400	T500	T600	T700	T800	T900
Liste items	I1, I2, I5	I2, I4, I6	I1, I3	I1, I2, I4	I2, I3, I8	I2, I3	I1, I3, I7	I1,I2,I3,I5	I1, I2, I3

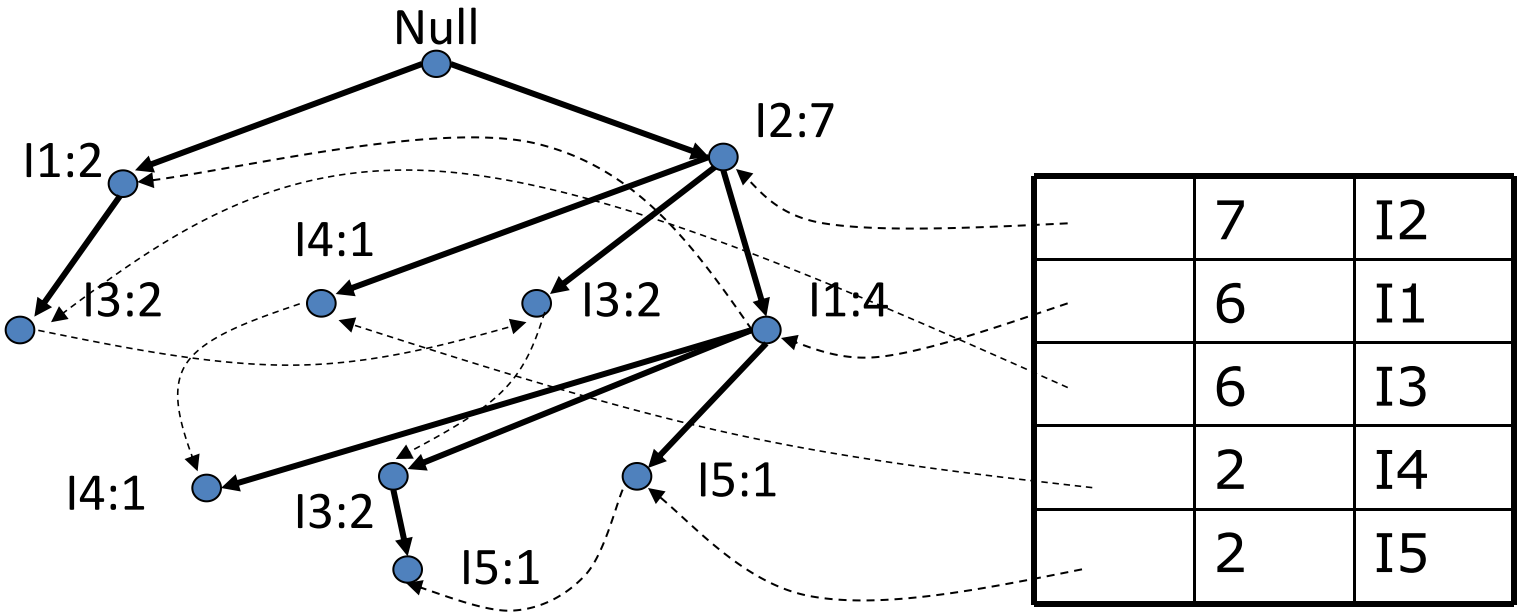
En lisant T300, l'ordre selon L est I1,I3. Ceci nous amène à ajouter une branche Null ➡ I1 ➡ I3. Noter qu'elle n'a pas de préfixe commun avec ce qui existe déjà. On obtient



FP-Trees : exemple

TID	T100	T200	T300	T400	T500	T600	T700	T800	T900
Liste items	I1, I2, I5	I2, I4, I6	I1, I3	I1, I2, I4	I2, I3, I8	I2, I3	I1, I3, I7	I1,I2,I3,I5	I1, I2, I3

Finalement, le FP-tree obtenu est



- Considérons I5. Il apparaît dans 2 branches

  I2 → I1 → I5:1 et I2 → I1 → I3 → I5:1

- Ainsi, pour le suffixe I5, on a 2 chemins préfixes: <I2,I1:1> et <I2,I1,I3:1>. Ils forment sa «table conditionnelle»

TiD	Itemset
1	I2, I1
2	I2, I3, I1

- Le «FP-tree conditionnel» de I5 contient

  une seule branche I2 → I1.

  I3 n'en fait pas partie car son support est 1 qui est <2

  (Rappel: min_support=2)

- Ce chemin unique va générer toutes les combinaisons de I5

  avec I1 et I2, i.e {I1,I5}:2, {I2,I5}:2, {I1,I2,I5}:2

Phase de l'exploration (suite)

- Considérons I4. Sa table conditionnelle est formée de <I2,I1:1> et <I2:1>

- Le FP-Tree conditionnel ne contient donc qu'un seul nœud I2

- Nous obtenons donc un itemset fréquent qui est {I2,I4}:2

Phase de l'exploration

Item	Base conditionnelle	FP-tree conditionnel	Itemsets générés
I5	<I2,I1>:1,<I2,I1,I3>:1	I2:2→I1:2	{I2, I5}:2 {I1, I5}:2 {I2,I1,I5}:2
I4	<I2,I1>:1,<I2>:1	I2:2	{I2, I4}:2
I3	<I2,I1>:2 ,<I2>:2, <I1>:2	I2:4→I1:2 I1:2	{I2, I3}:4 {I1,I3}:4 {I2, I1, I3}:2
I1	<I2>:4	I2:4	{I2,I1}:4

Ce n’est pas la peine de regarder I2 car ça va donner les combinaisons avec les autres items qui ont déjà été considérés

Application

Genome Biology 2007, 8:R3 (doi:10.1186/gb-2007-8-1-r3)

GENECODIS: a web-based tool for finding significant concurrent annotations in gene lists

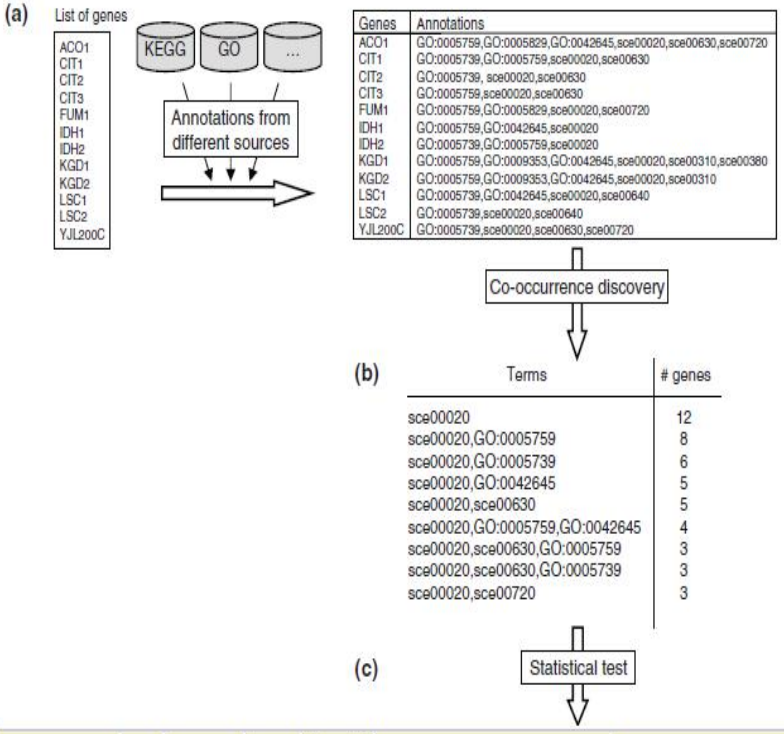
Pedro Carmona-Saez*, Monica Chagoyen*†, Francisco Tirado†, Jose M Carazo* and Alberto Pascual-Montano†

Abstract

We present GENECODIS, a web-based tool that integrates different sources of information to search for annotations that frequently co-occur in a set of genes and rank them by statistical significance. The analysis of concurrent annotations provides significant information for the biologic interpretation of high-throughput experiments and may outperform the results of standard methods for the functional analysis of gene lists. GENECODIS is publicly available at http://genecodis.dacya.ucm.es/.

Finding sets of terms that frequently appear together in a list of genes

To extract combinations of gene annotations, GENECODIS uses a modification to the methodology reported by Carmona-Saez and coworkers [6], which implements the *apriori* algorithm to extract associations among gene annotations and expression patterns.



Annotation/s	# List	# Reference	*p*-value	Corrected *p*-value	Genes	Description/s
00020	12(12)	30(6194)	1.90e-28	1.90e-27	CIT2, ACO1, KGD2, LSC2, YJL200C, IDH2, LSC1, KGD1, IDH1, CIT1, FUM1, CIT3	(KEGG)Citrate cycle (TCA cycle)
00020, GO:0005759	8(12)	9(6194)	1.52e-21	1.52e-20	ACO1, KGD2, IDH2, KGD1, IDH1, CIT1, FUM1, CIT3	(KEGG)Citrate cycle (TCA cycle) | (CC)mitochondrial matrix
00020, GO:0005739	6(12)	9(6194)	5.43e-15	5.43e-14	CIT2, LSC2, YJL200C, IDH2, LSC1, CIT1	(KEGG)Citrate cycle (TCA cycle) | (CC)mitochondrion
00020, GO:0042645	5(12)	7(6194)	5.83e-13	5.83e-12	ACO1, KGD2, LSC1, KGD1, IDH1	(KEGG)Citrate cycle (TCA cycle) | (CC)mitochondrial nucleoid
00020, 00630	5(12)	8(6194)	2.62e-12	2.62e-11	CIT2, ACO1, YJL200C, CIT1, CIT3	(KEGG)Citrate cycle (TCA cycle) | (KEGG)Glyoxylate and dicarboxylate metabolism

Critiques des notions de support et de confiance

- parmi 5000 séquences protéiques
  - 3000 arborent le domaine J
  - 3750 arborent le domaine Zn
  - 2000 arborent J et Zn
- J → Zn [40%, 66.7%]

  n'est pas informative car il y a 75% des protéines qui arborent le domaine Zn ce qui est plus que 66.7%.
- J → no Zn [20%, 33.3%] est plus pertinente même avec un support et une confiance inférieurs

	J	no J	total
Zn	2000	1750	3750
no Zn	1000	250	1250
total	3000	2000	5000

Critiques des notions de support et de confiance

- Exemple 2:
  - X et Y positivement corrélés,
  - X et Z négativement corrélés
  - Les support et confiance de

    X → Z dominent

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

- Nous avons besoin d'une mesure de corrélation

$$corr_{A,B} = \frac{P(AB)}{P(A)P(B)}$$

- est aussi appelé le lift de A → B

Autres mesures

- Intérêt (corrélation) $\frac{P(A \wedge B)}{P(A)P(B)}$
  - Prendre en compte P(A) et P(B)
  - P(A ^ B)=P(B)*P(A),

    si A et B sont des événements indépendants
  - A et B négativement corrélés, si corr(A,B)<1.

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Itemset	Support	Intérêt
X,Y	25%	2
X,Z	37,50%	0,9
Y,Z	12,50%	0,57

Pour aller plus loin

- associations multi-niveaux
  - hiérarchie de concepts : brie - fromage - produit laitier - ...
- assocations multi-dimensionnelles
  - achète(x), âge(x), occupation(x)
- associations temporelles
- associations avec contraintes
  - Contraintes sur les données :
    - Trouver les paires de produits vendus à Toulouse en Décembre 98
  - Contraintes sur l'intérêt
    - support, confiance, corrélation
  - Contraintes sur les dimensions :
    - En rapport à région, prix, marque, catégorie client
  - Contraintes sur les règles :
    - Nombres de prédicats dans le corps

Règles d'association multi-niveaux

Les items forment des hiérarchies.

Les items au niveau inférieur ont des supports inférieurs

Les bases de transactions peuvent prendre en compte les niveaux