

# *supports attendus bi*

## 1 Introduction

*supportsAttendusBi* calcule les supports attendus de patterns clos abstraits locaux identifiés par *minerLC* dans un graphe bi-partie.

## 2 Dossiers disponibles

Le dossier *supports\_attendus\_bi* contient le *makefile* ainsi que les sous-dossiers :

- *bin* contenant l'exécutable,
- *src* contenant les programmes sources et le makefile,
- *include* contenant les fichiers inclus,
- *doc* contenant la documentation,
- *data* contenant des données exemples.

## 3 Compiler *supportsAttendus*

*supportsAttendusBi* peut être exécuté sur tout système qui compile des programmes c++11. Ouvrir un terminal, aller dans le dossier *supportsAttendusBi* et taper *make* pour créer un fichier exécutable *supportsAttendusBi*.

## 4 Executer *supportsAttendusBi*

*supportsAttendusBi* attend obligatoirement et dans l'ordre 1) le nom du fichier (format nri) contenant les transactions et le graphe et 2) le nom du fichier contenant les patterns clos abstraits identifiés par *minerLC*. Par exemple :

`supportsAttendusBi mougel.nri mougel.res`

On peut facultativement utiliser les paramètres relatifs aux abstractions définies dans la doc de *minerLC*.

Par ailleurs, l'argument  $-n\ k$  signifie qu'on demande  $k$  tirages aléatoires (par défaut  $k = 1$ ). Par exemple

`supportsAttendusBi mougel.nri mougel.res -c 4 -t1 -n 5`

exécute *supportsAttendusBi* à partir du graphe contenu dans le fichier *mougel.nri* et des patterns clos abstraits issus du programme *minerLC* contenus dans le fichier *mougel.res*. Le programme exécutera  $k = 5$  itérations en appliquant une abstraction de composante connexes de taille au moins 4 et une abstraction de triangles.

## 5 Format des données d'entrée

Le fichier *test\_bip.nri* et *test\_bip.res* du dossier *data* contiennent des exemples des données d'entrées. Chaque ligne (hors commentaire) d'un fichier *.res* est de la forme

$$c, ext(c), \{authorities\{\dots\}, hubs\{\dots\}\}$$

où  $c$  est le motif clos abstrait,  $ext(c)$  son extension,  $authorities\{\dots\}$  le sous-ensemble de l'extension abstraite contenant les nœuds de rôle "authorities" et  $hubs\{\dots\}$  le sous-ensemble de l'extension abstraite contenant les nœuds de rôle "hub". Ex)

`{"blues", "folk"}, {"oB", "iE", "iF"}, {authorities{"iF"}, hubs{"oB"}}`

Si l'abstraction *hub\_authority* n'a pas été utilisée chaque ligne est de la forme :

$$c, ext(c), \{no\_role\{\dots\}\}$$

où *no\_role* contient l'extension abstraite.

## 6 Format des sorties

Chaque ligne du fichier de sortie de *supportsAttendusBi* est de la forme

$$c, sOut, sIn, sa, \mu, \sigma$$

où  $c$  est le motif clos abstrait,  $sOut$  son support observé sur les noeuds  $Out$ ,  $sIn$  son support observé sur les noeuds  $In$ ,  $sa$  son support abstrait observé,  $\mu$  son support abstrait attendu et  $\sigma$  l'écart-type de ce support sur les  $n$  tirages.

Par exemple :

`{"folk", "jazz", "rock"}, 3, 1, 3, 4, 0`