

# Mode d'emploi CoGui – RCA

## Installation de CoGui

CoGui est un logiciel multiplateforme écrite en Java. Vous trouverez les instructions pour installer ici :

<http://www.lirmm.fr/cogui/userdoc/Installation.html>

## Installation de GraphViz

Ce type de fichier est traité par le logiciel GraphViz. L'assistant n'a pas besoin de GrapViz pour fonctionner mais son installation permet d'éditer et visualiser ces fichiers .DOT dans CoGui . Il y a deux étapes pour réaliser cette installation.

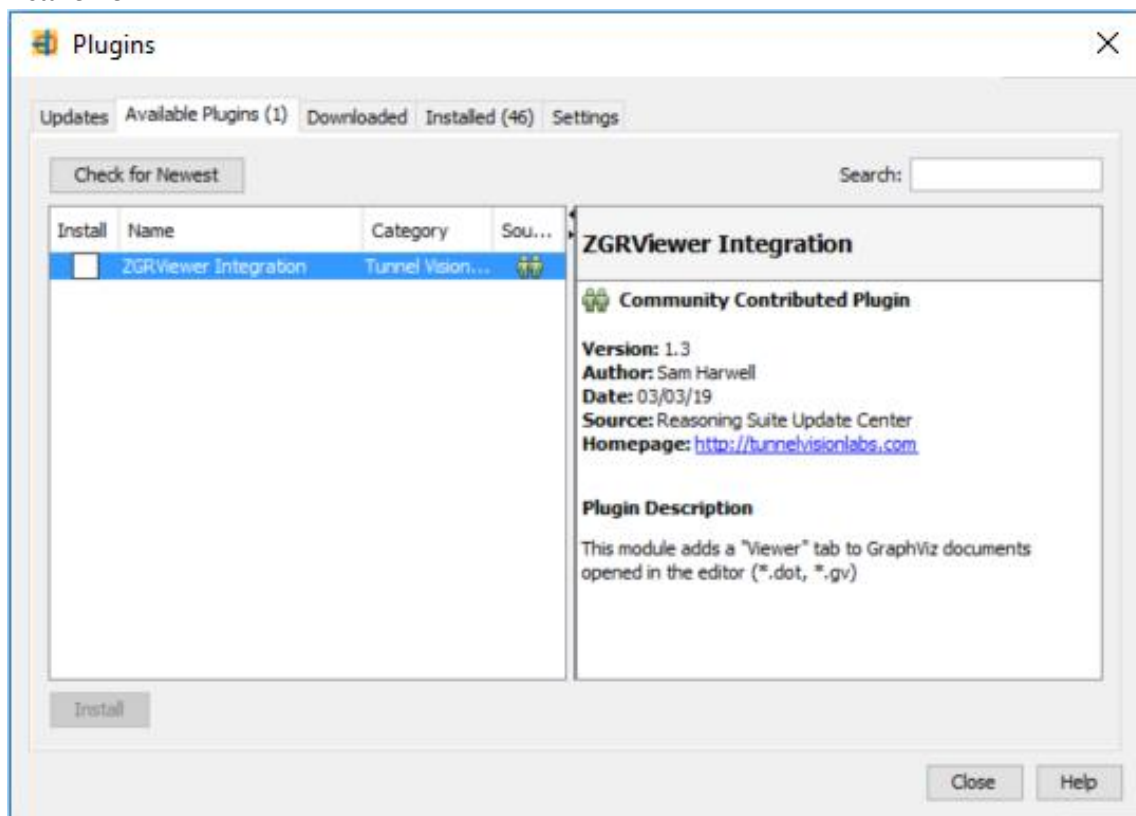
### 1) Installation du logiciel GrapViz

Vous trouverez le programme d'installation ici : <https://www.graphviz.org/download/>

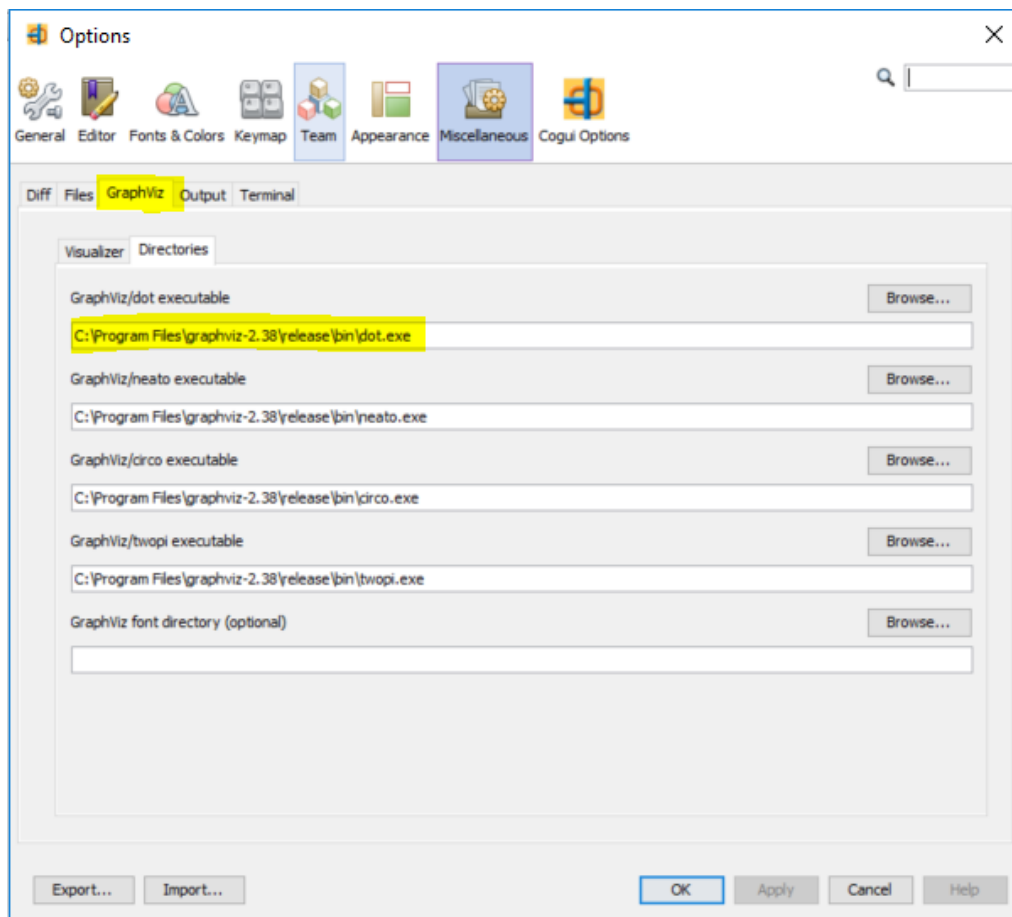
Notez l'emplacement sur lequel est installé le logiciel, vous en aurez besoin ultérieurement.

### 2) Installation du Plugin ZGRViewer

Ce plugin permet d'éditer les fichiers (.DOT) produits par l'assistant. Dans CoGui, ouvrez la fenêtre de gestion des plugins (Menu Tools/Plugins), vous trouverez le plugin « ZGRViewer Integration » dans le volet « Available Plugins ». Installez-le.

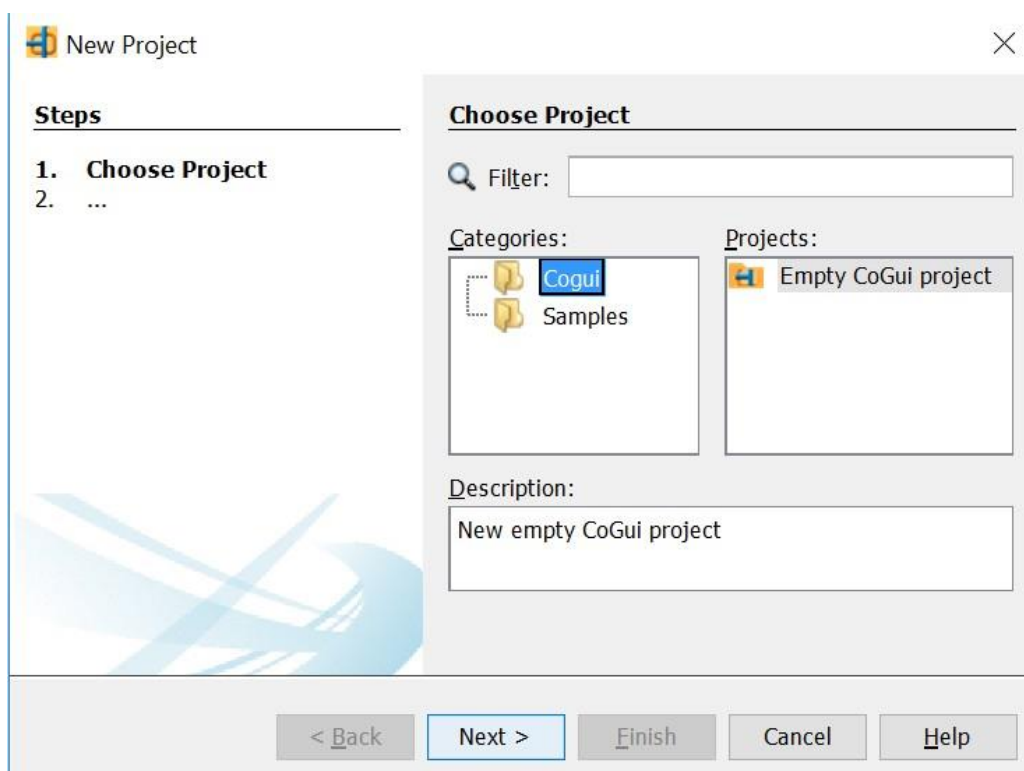


Après avoir relancé CoGui, il reste à configurer le plugin pour qu'il utilise GraphViz précédemment installé. Accédez aux options (menu Tools/Options/Miscellaneous/GraphViz/Directories) et précisez l'emplacement des fichiers executables, en particulier celui surligné en jaune, c'est le programme DOT (.exe sous windows) qui est utilisé par l'assistant.



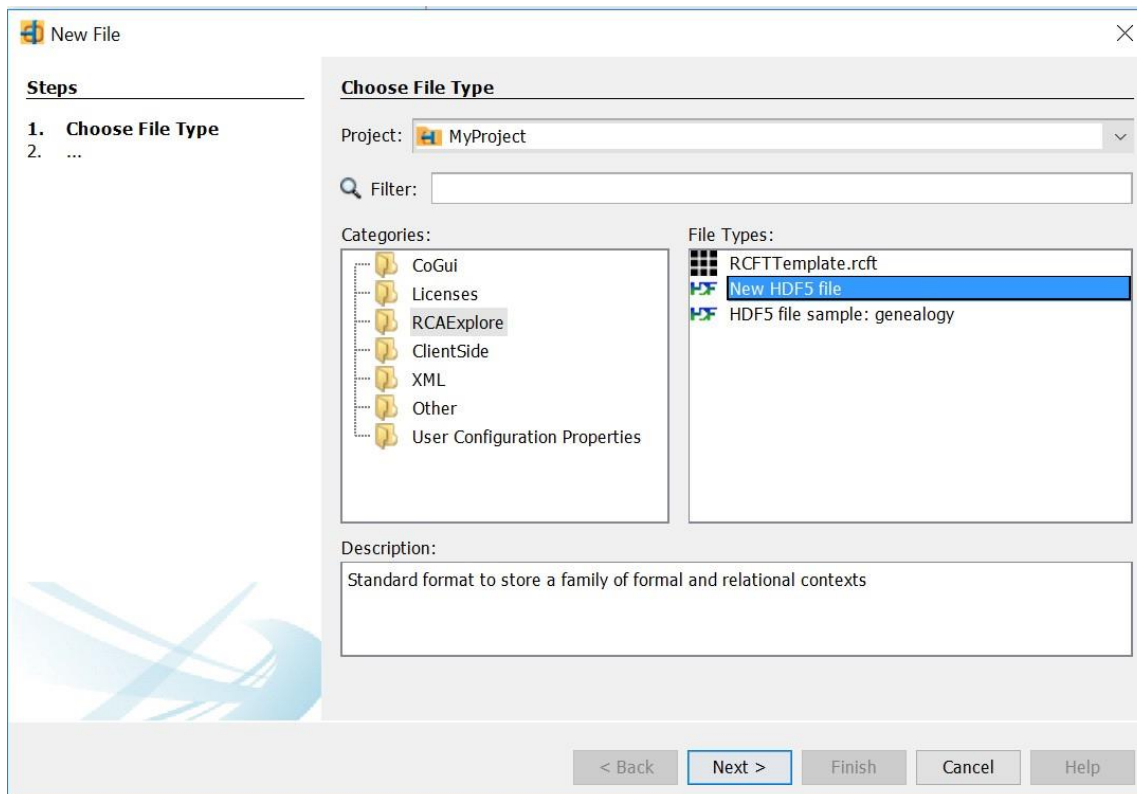
## Création d'un projet CoGui

Créer un projet vide avec la commande File/New project...

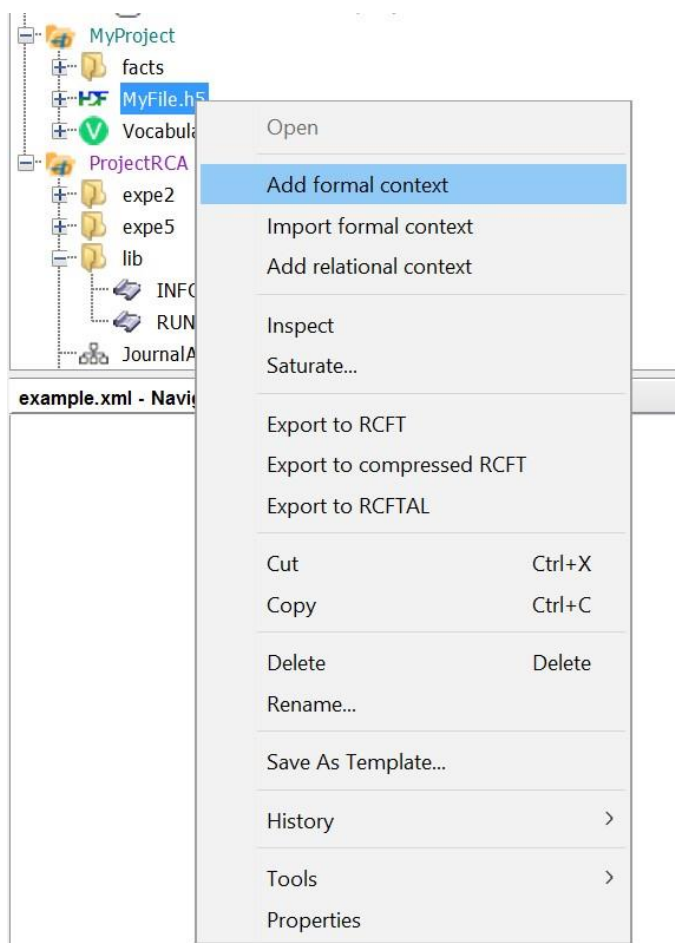


## Création d'une famille RCA

Utilisez l'assistant New file pour créer un fichier de type HDF5. Un exemple est également disponible.

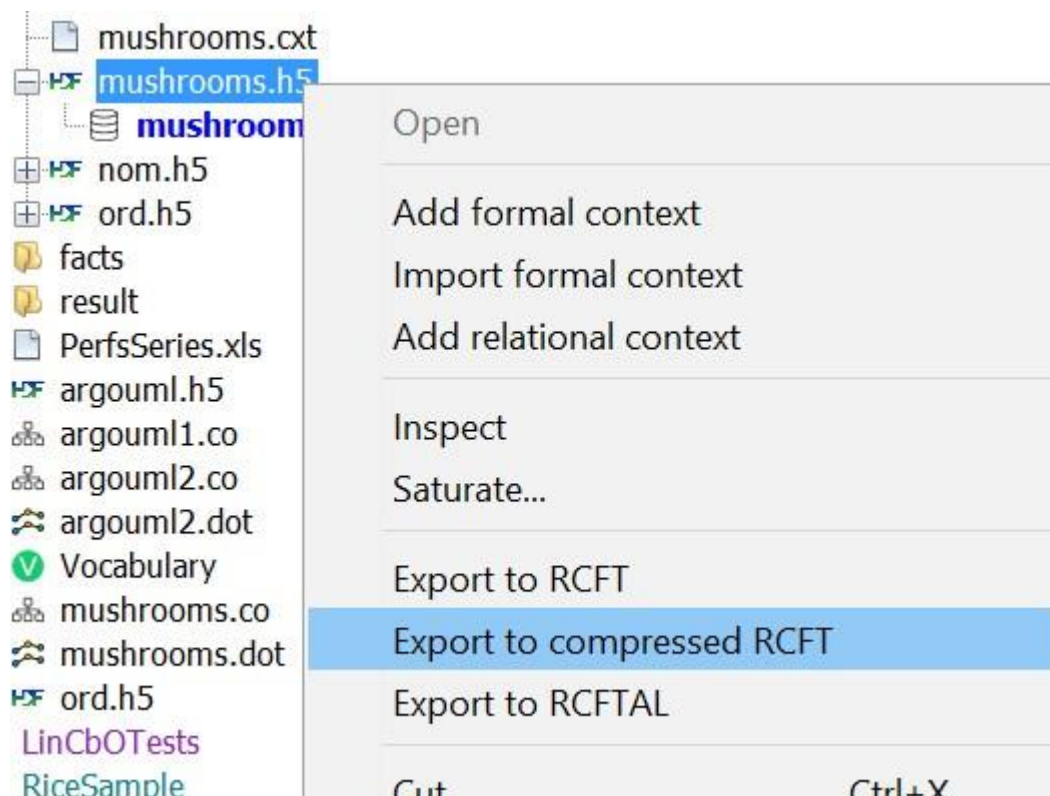


A partir du nouveau fichier HDF5 on peut créer, importer et stocker contextes formels et relations contextuelles.



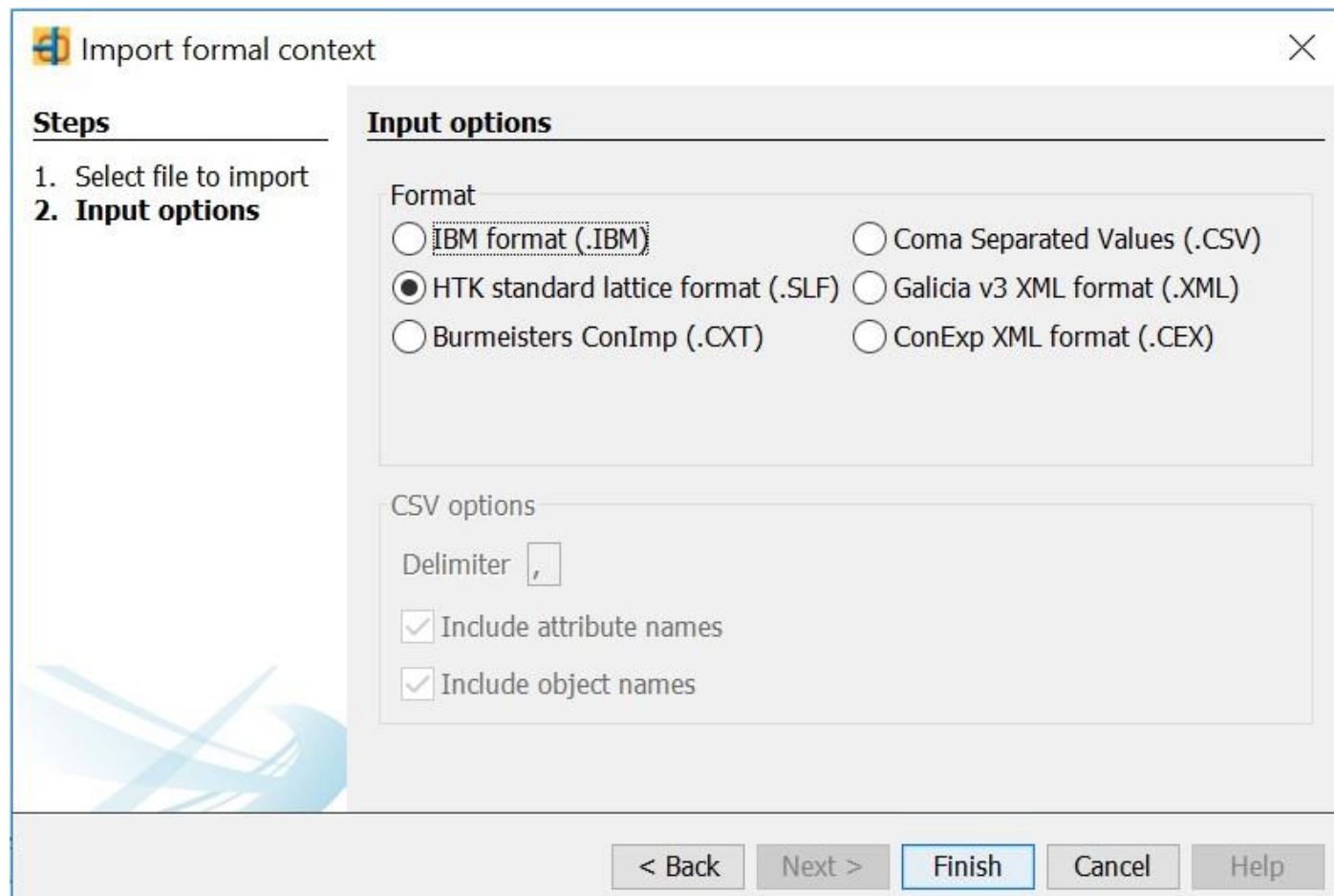
Suivez les instructions à partir des différentes commandes du menu contextuel.

On peut convertir les fichiers HDF5 qui contiennent une famille entière dans 3 autres formats RCFT, RCFT compressé et RCFTAL, Chacun de ces formats est convertible dans un autre sans perte d'information.



Les commandes Add... créent des contextes vides.

La commande « Import formal context » supporte 6 formats d'entrée.



Les 6 fichiers suivants représentent la même matrice :

[example.csv](#)

[example.ibm](#)

[example.xml](#)

[example.slf](#)

[example.cex](#)

[example.cxt](#)

0	1	1	0	1	1
1	0	0	1	1	1
1	0	0	1	0	0
1	0	0	1	0	1
0	0	1	1	1	1
1	0	0	1	1	1

On peut exporter chaque contexte formel dans l'un des formats précités :

**Export formal context**

**Steps**

- Output options**

**Output options**

File name

Parent folder

**Format**

☐ HDF5 RCAExplore format (.H5) ☐ RCAExplore format (.RCFT)

☐ Json adjacency list (.RFCTAL) ☐ Compressed RCFT (.RCFTGZ)

☐ Coma Separated Values (.CSV) ☐ HTK standard lattice format (.SLF)

☐ Galicia v3 XML format (.XML) ☒ Burmeisters ConImp (.CXT)

☐ ConExp XML format (.CEX)

**CSV options**

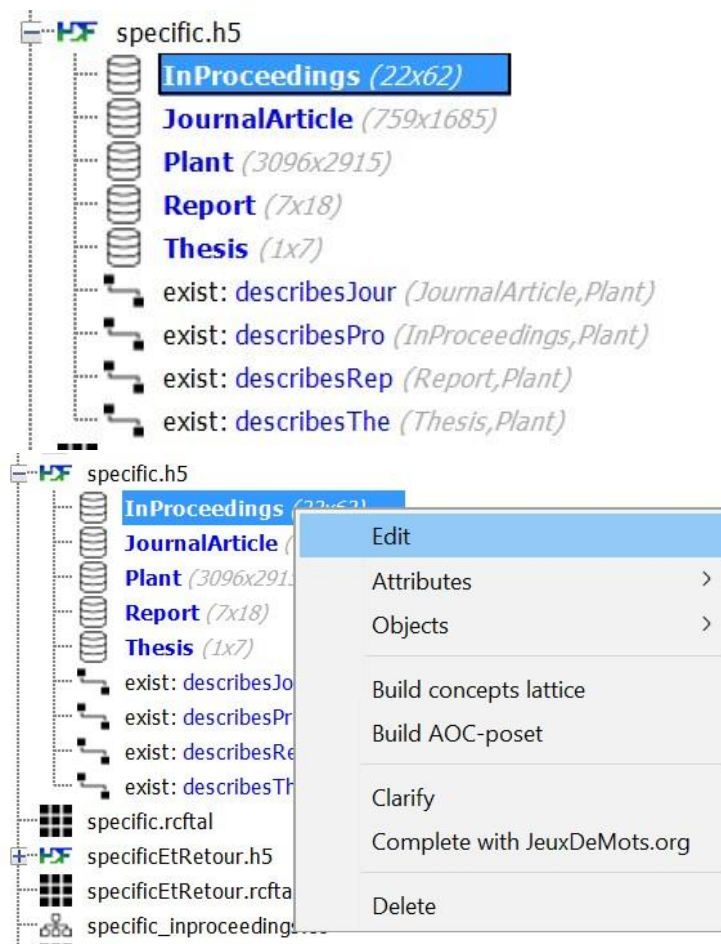
Delimiter

☒ Include attribute names

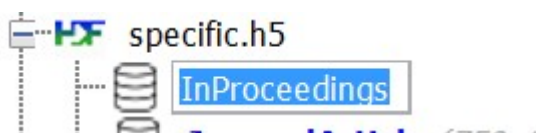
☒ Include object names

## Edition des contextes

Les différents contextes que contient le fichier HDF5 sont accessibles depuis l'arbre du projet. Il donne accès aux commandes opérables sur les contextes formels et les relations :



Le renommage des éléments peut se faire directement sur l'arbre de projet :



La commande Edit ouvre une fenêtre d'édition à partir de laquelle on peut modifier le contenu de la matrice mais également créer/ renommer/ supprimer attributs et objets :



specific:InProceedings														
Objects	Attributes	Author_Chikukura,J	Title_Evaluation...	BookTitle_African...	pages_(0-16)	pages_(0-40)	pages_(0-500)	pages_(0-6)	pages_(6-500)	pages_(10-40)	pages_(6-40)	pages_(10-500)	pages_(20-500)	pages_(6-16)
5U7MLWCE		<	<	<										
7HWF7VCT														
99VQRIYL														
9JKSB2QB														
ASQJAIPH					<	<	<		<	<	<	<		<
BF9ZV7E8														
DI3A5CPF				<	<	<	<	<						
IEENPM9M														
J4CRFUY6														<
MPKT555I				<	<	<	<	<						<
N8P93Z7J					<	<	<	<						<
TYTNPUIB														
X2GXBBNT					<	<	<	<						
XUR527CA														

specific:InProceedings						
Objects	Attributes	Author_Chikukura,J	Title_Evaluation...	BookTitle_African...	pages_(0-16)	pages_(0-40)
5U7MLWCE		<	<	<		
7HWF7VCT						
99VQRIYL						
9JKSB2QB						
ASQJAIPH						
BF9ZV7E8						
DI3A5CPF				<	<	<
IEENPM9M						
J4CRFUY6						

Add attribute

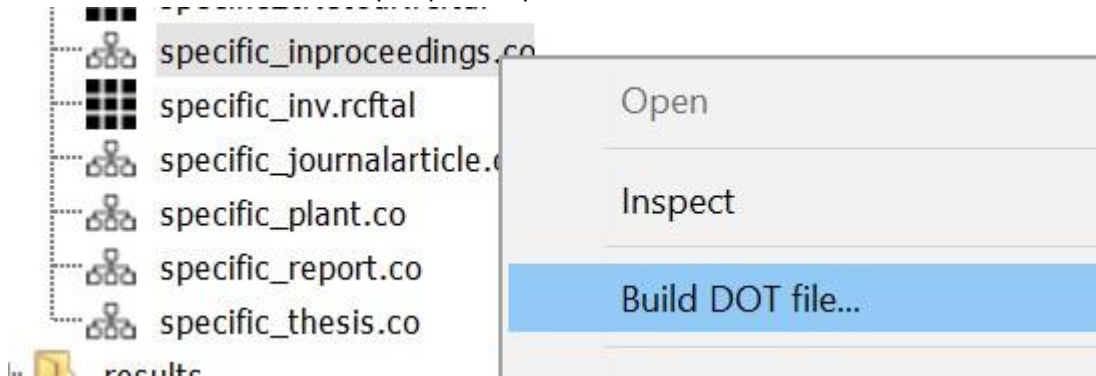
Add object

Double clic sur les noms d'objets et d'attributs pour les éditer

ATTENTION : il n'y a pas de contrôle d'intégrité référentielle entre les collections d'objets des contextes formels et les relations contextuelles. Toute modification d'un contexte formel après création de la relation doit être reporté dans celle-ci.

## Production des ordres de concepts

Les assistants « Build concept lattice » et « Build AOC-poset » attachés aux contextes formels vous permettront de construire des ordres de concepts qui sont produits et stockés dans des fichiers .CO :



C'est à partir des fichiers .CO qu'on peut accéder à l'assistant pour construire une représentation de l'ordre au format .DOT de graphiz

Vous pourrez ouvrir le fichier .DOT :

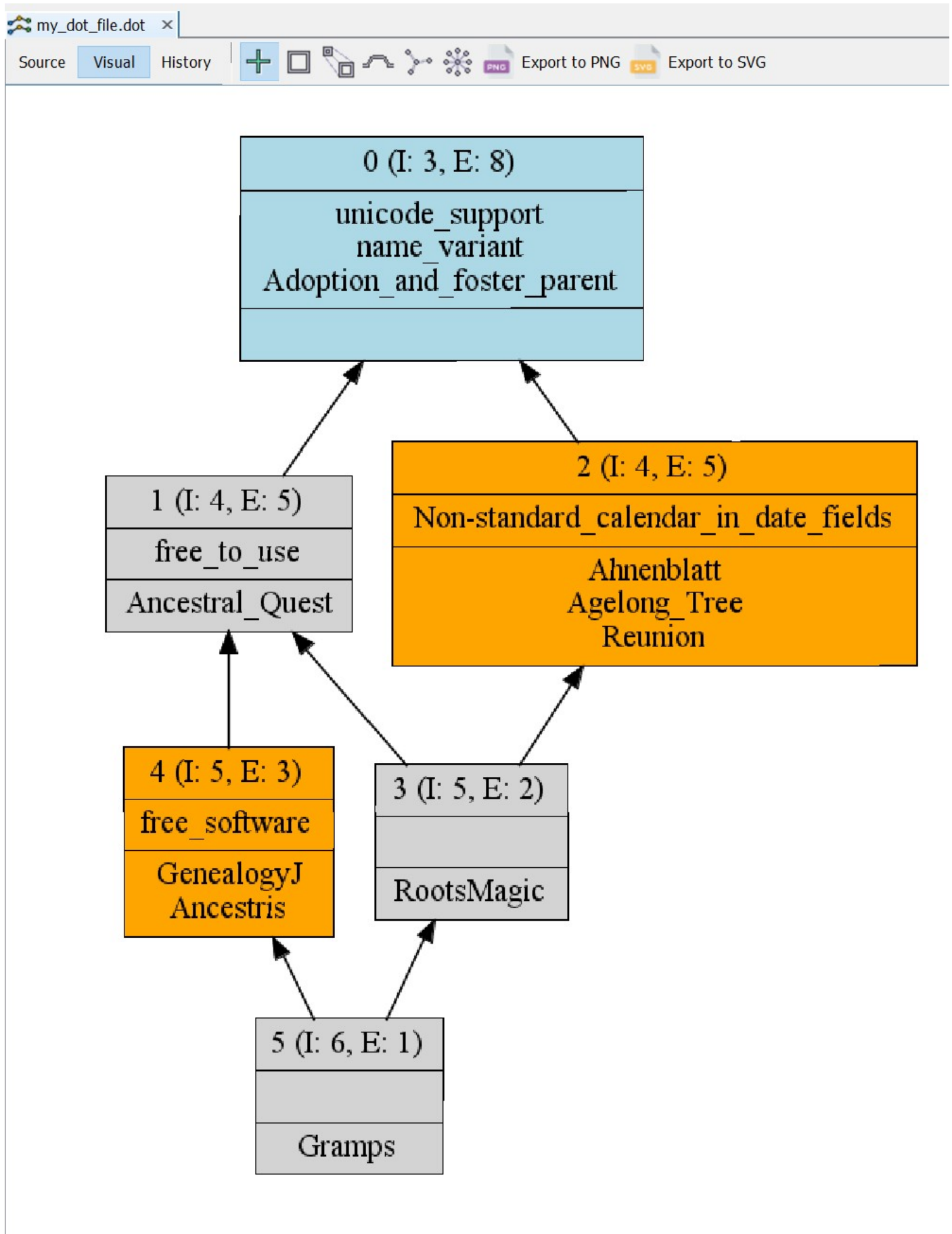
A screenshot of a text editor window titled 'my\_dot\_file.dot'. The editor has three tabs: 'Source' (selected), 'Visual', and 'History'. The 'Source' tab shows a .DOT file for a directed graph. The code is as follows:

```
1 digraph G {
2     rankdir=BT;
3     0 [shape=record,style=filled,fillcolor=lightblue,label="{0 (I: 3, E: 8)|u
4     1 [shape=record,style=filled,label="{1 (I: 4, E: 5)|free_to_use\n|Ancestr
5     2 [shape=record,style=filled,fillcolor=orange,label="{2 (I: 4, E: 5)|Non-
6     3 [shape=record,style=filled,label="{3 (I: 5, E: 2)||RootsMagic\n}"];
7     4 [shape=record,style=filled,fillcolor=orange,label="{4 (I: 5, E: 3)|free_
8     5 [shape=record,style=filled,label="{5 (I: 6, E: 1)||Gramps\n}"];
9         1 -> 0
10        2 -> 0
11        3 -> 1
12        3 -> 2
13        4 -> 1
14        5 -> 3
15        5 -> 4
16    }
```

Si le plugin ZGRViewer a été installé, l'éditeur dispose d'un volet supplémentaire « Visual » surligné en jaune.

Actionnez le volet pour accéder à la représentation graphique :

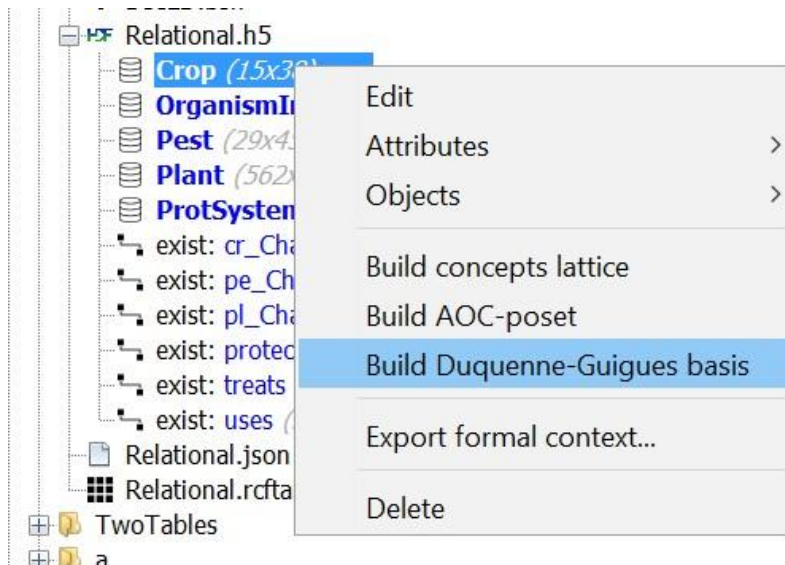




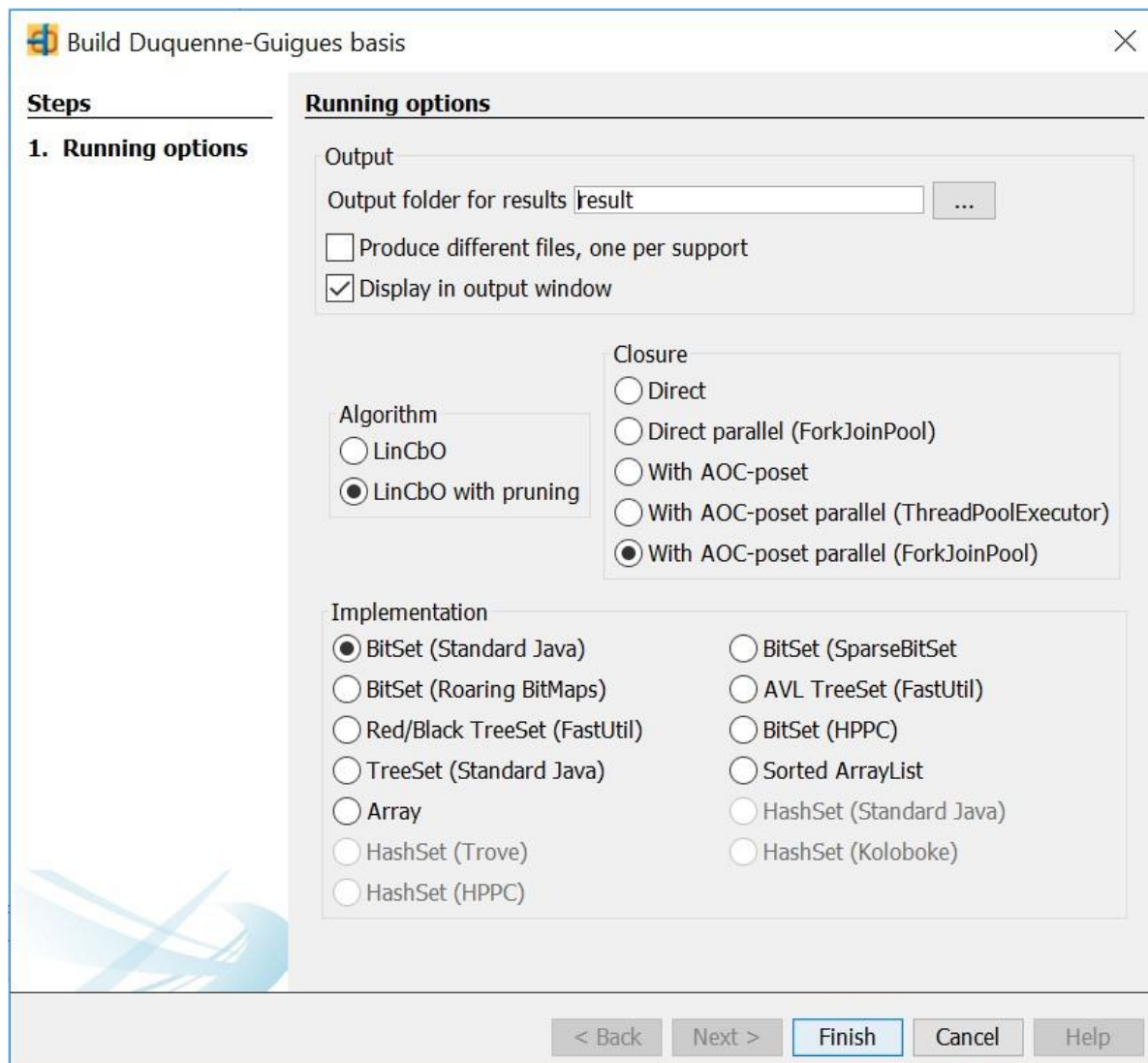
La barre d'outils propose plusieurs modes de navigation et de zoom. Deux commandes d'exportation sont disponibles pour des images classiques (.PNG) ou bien des dessins vectorisés (.SVG) ce dernier format est idéal pour afficher sur une page web ou bien pour apporter des modifications avec un logiciel tel que Inkscape (<https://inkscape.org/fr/> )

## Calcul de la base canonique (base de règles Duquenne-Guigues)

Pour calculer la base canonique d'un contexte formel, sélectionner l'action « Build Duquenne-Guigues basis » :



Tous les résultats sont placés dans un dossier unique. Un fichier contient les règles de la base sous la forme <support> prémice->conclusion. Optionnellement un fichier différent peut être produit par support. Deux algorithmes sont disponibles ainsi que plusieurs options d'implémentation, y compris pour l'opération de clôture qui est déterminante pour les performances.

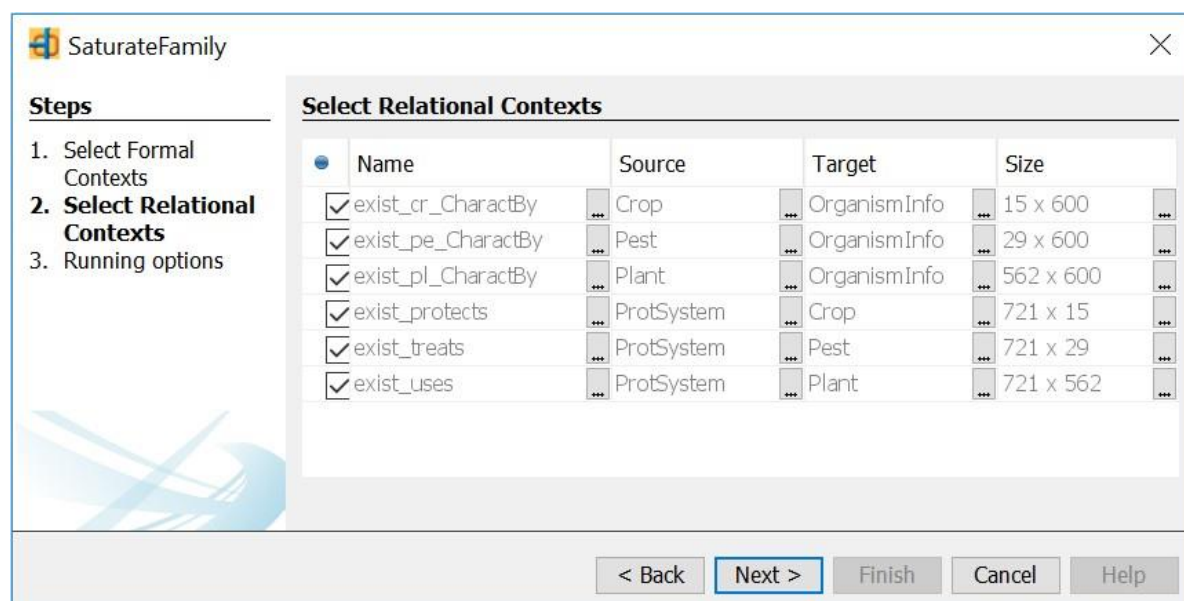
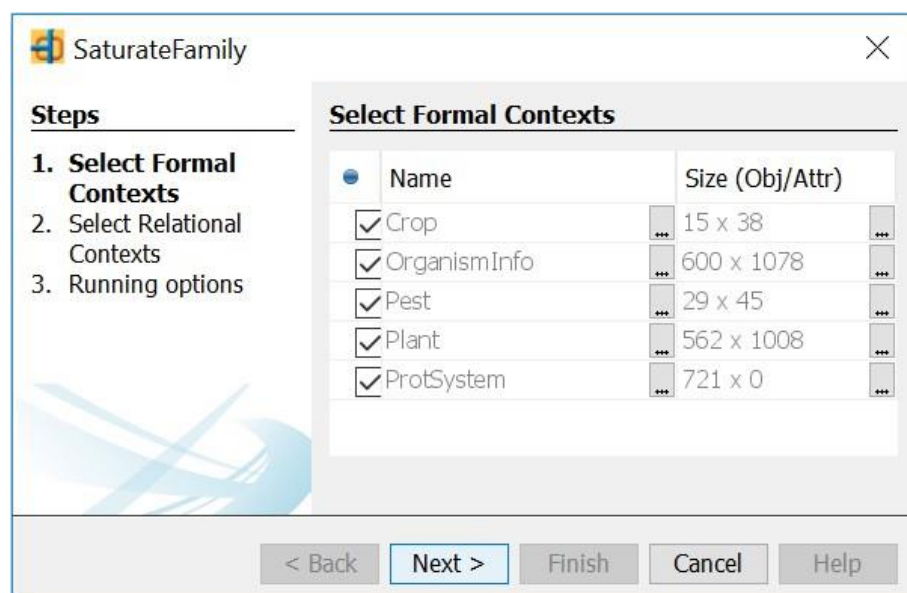
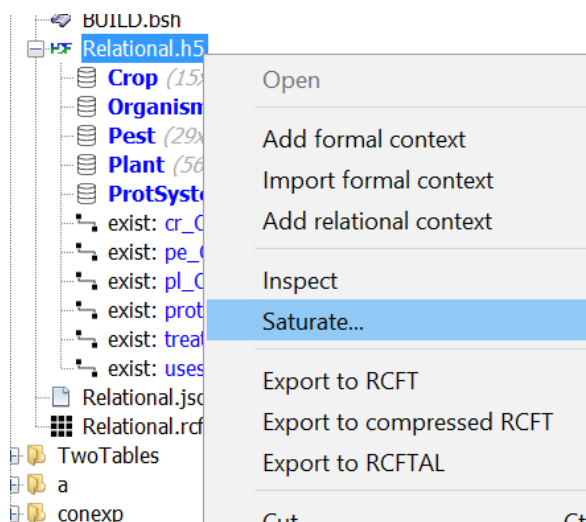


Seules les implémentations de représentation des ensembles qui sont triées peuvent être utilisés.

## Saturation

Un assistant permet de créer une structure associée à une famille RCA

Sélectionner l'action « Saturate » associé au fichier HDF5. On peut sélectionner tout ou partie de la famille.



On peut sélectionner diverses options de sortie, les résultats sont placés dans un dossier unique. La structure des concepts peut être fondée sur l'AOC poset ou bien un treillis.

SaturateFamily

×

Steps

1. Select Formal Contexts

2. Select Relational Contexts

3. **Running options**

Running options

Output folder for results  ...

☒ Build .DOT files

☒ Store extended family

☐ Build relational attribute names with intent

Stop after this step limit:

Algorithm

☐ AOC-poset with Ares

☐ AOC-poset with Ceres

☒ AOC-poset with Hermes

☐ AOC-poset with Pluton

☐ Complete lattice

☐ Iceberg

Implementation

☐ BitSet (Standard Java)

☐ BitSet (SparseBitSet)

☐ BitSet (Roaring BitMaps)

☐ AVL TreeSet (FastUtil)

☐ Red/Black TreeSet (FastUtil)

☐ BitSet (HPPC)

☐ TreeSet (Standard Java)

☐ Sorted ArrayList

☐ Array

☒ HashSet (Standard Java)

☐ HashSet (Trove)

☐ HashSet (Koloboke)

☐ HashSet (HPPC)

< Back

Next >

**Finish**

Cancel

Help