VisCad User Guide

1 Introducing VisCad

2 Visualization

3 Code Clone Metrics

4 Filtering

Introducing VisCad

Introduction

Detection and analysis of similar code fragments (“code clones”) has become an integral part of software maintenance. In response, over the last decade a great many clone detection techniques and tools have been proposed. However, identifying useful cloning information from the large volume of textual data produced by these detectors is challenging. VisCad is a tool with which a user can visualize and analyze large volumes of raw cloning data in an interactive fashion. Users can analyze and identify distinctive code clones through a set of visualization techniques, metrics and data filtering operations. The loosely coupled architecture of VisCad allows users to work with the clones of any clone detection tools that report source co-ordinates of the found clones. This yields the opportunity to work with the clone detectors of choice, which is important for clone analysis since clone detectors have their own strengths and weaknesses.

System Requirement

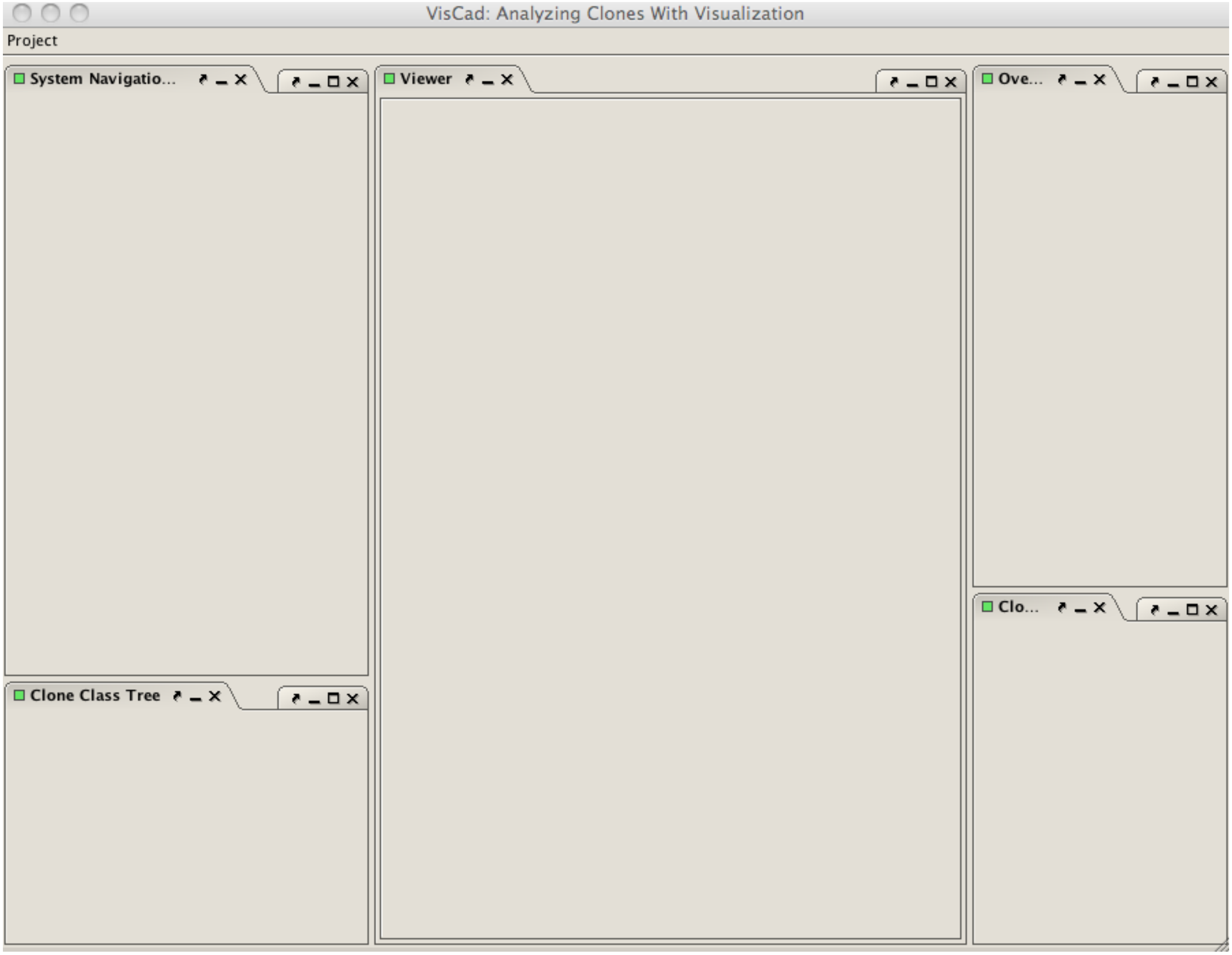
1. VisCad requires Java Runtime Environment (JRE) 6 or later. You can download the recent JRE from here .
2. We have successfully tested VisCad on Windows XP, Windows 7, Max OS X(version 10.6) and on some Linux distribution (such as Ubuntu).
4. The more RAM your computer has, the better performance you gain from VisCad. However, 2.0 GB or more is recommended.
5. Display dimension of 1024 x 768 or greater is recommended.

Obtaining and Running VisCad

Follow the steps listed below:

1. Download the VisCad_beta.zip file from here . You can obtain the most recent version of VisCad, documentation, source code from this location.
2. Extract the contents of the archive.
3. Double click on the VisCad.jar file to run the program.

VisCad user interface

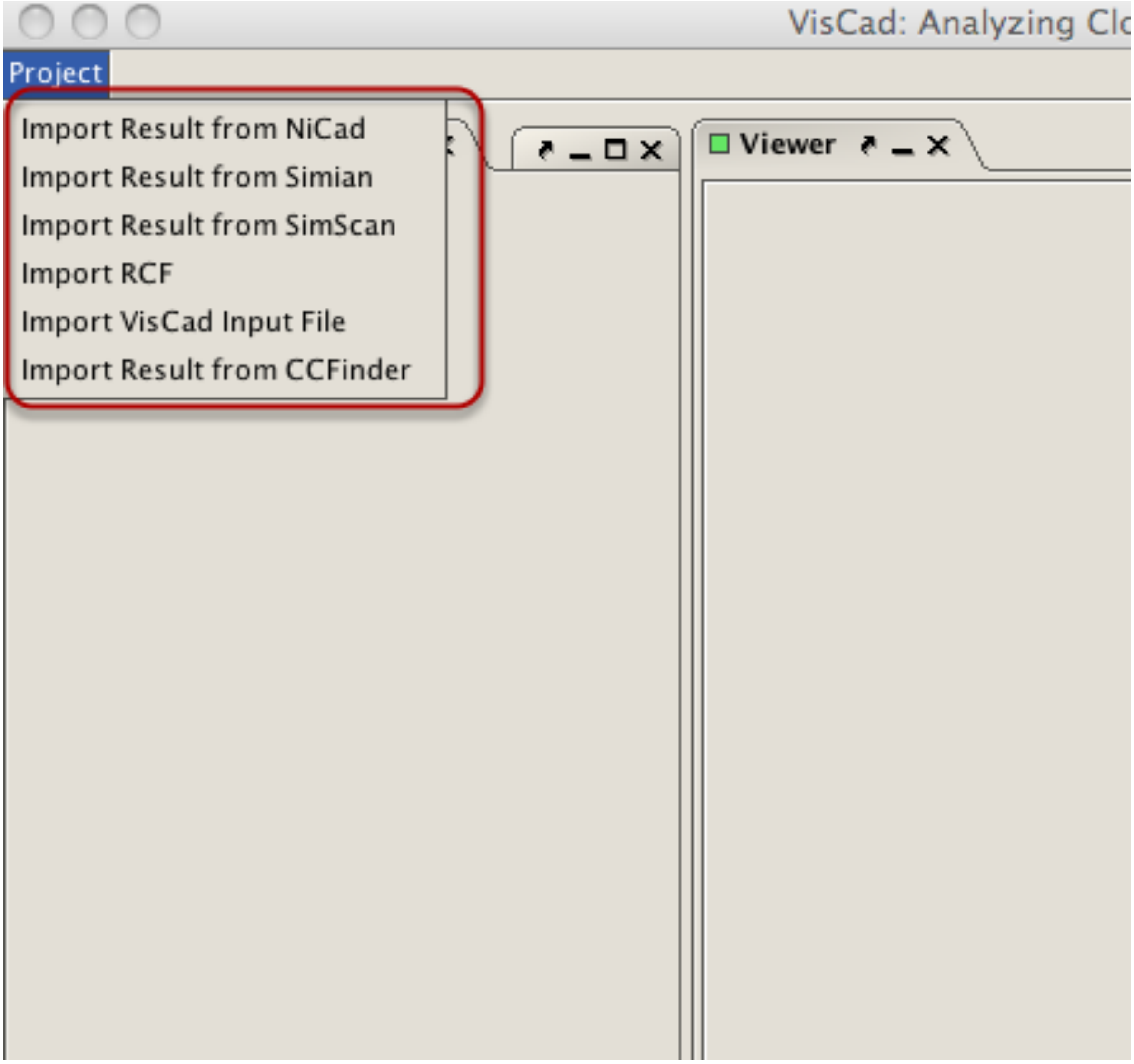


Running the viscad.jar file opens the above window.

Importing clone detection result

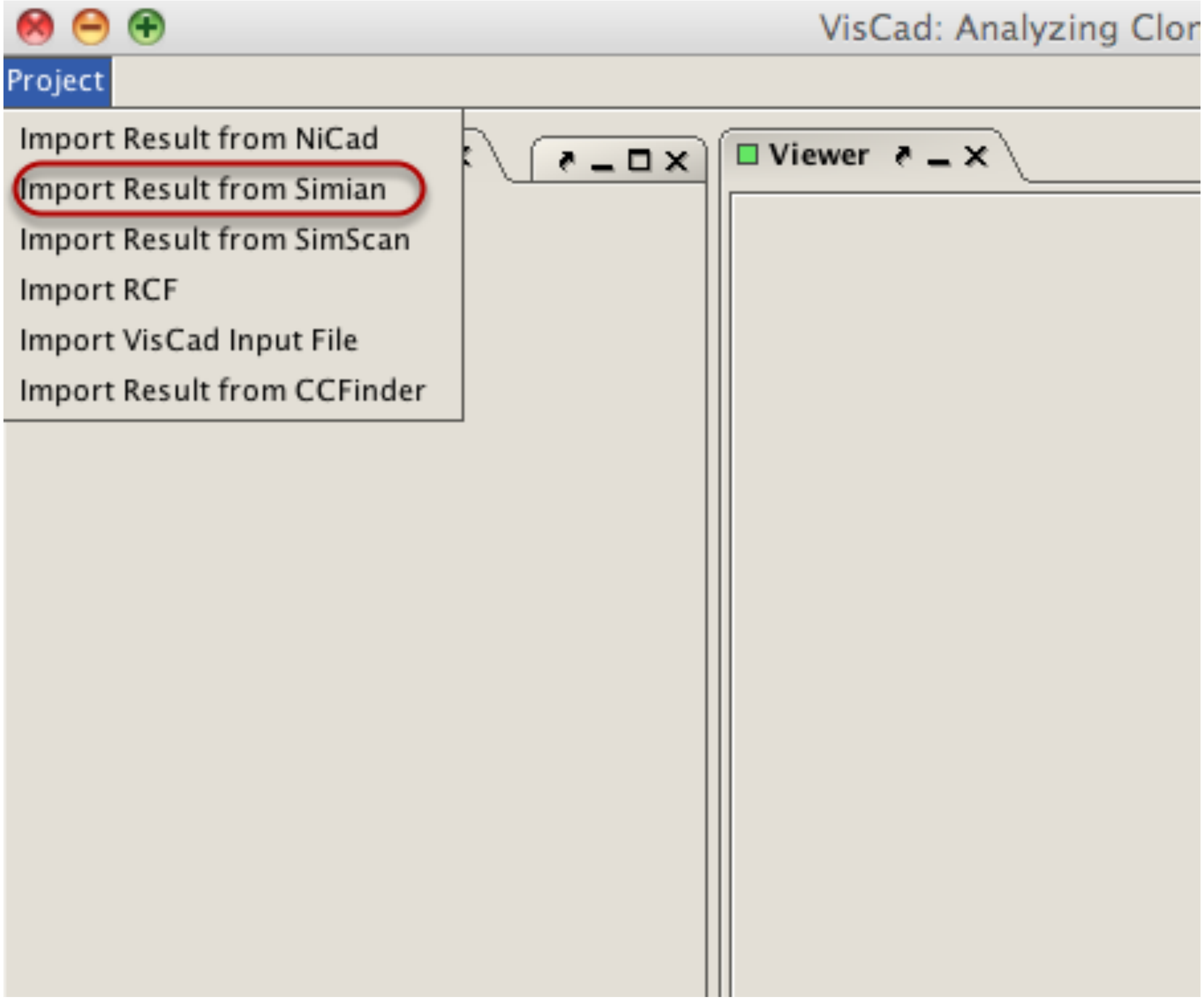
VisCad requires the subject system and the clone detection result you obtained by running clone detection on the subject system using the supported clone detectors.
Current you can directly import clone detection result of CCFinder, Simian, SimScan, NiCad. If you have clone detection result in RCF format, you can also import and anlyze the data in VisCad. For other clone detection tools, you need to convert the result into VisCad input file format. VisCadBeta.zip file contains an example of VisCad input file.

Make the selection



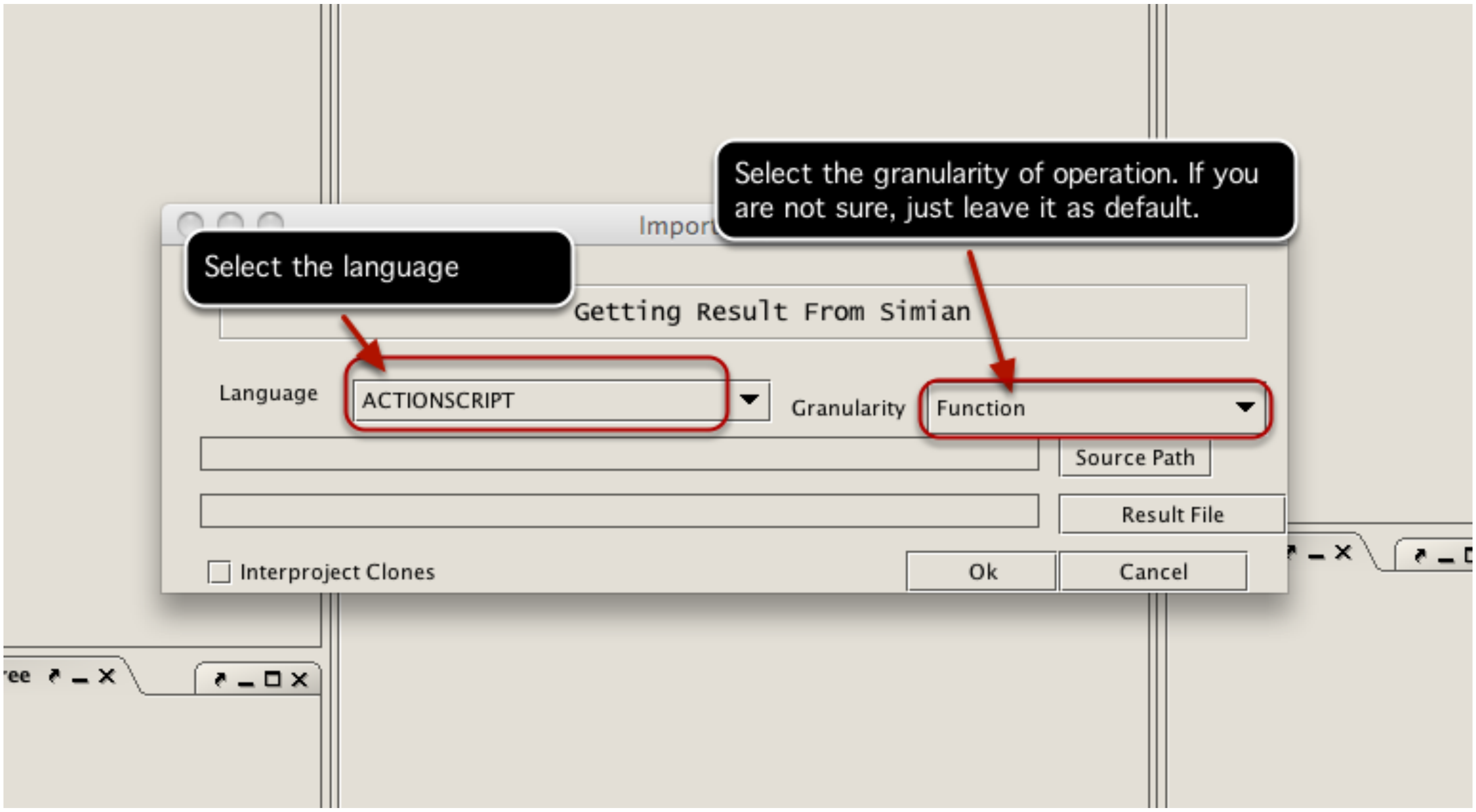
To import clone detection result or RCF data file, click on the *Project* menu. This opens a list of importing operations supported by VisCad.

Import clone detection result of Simian



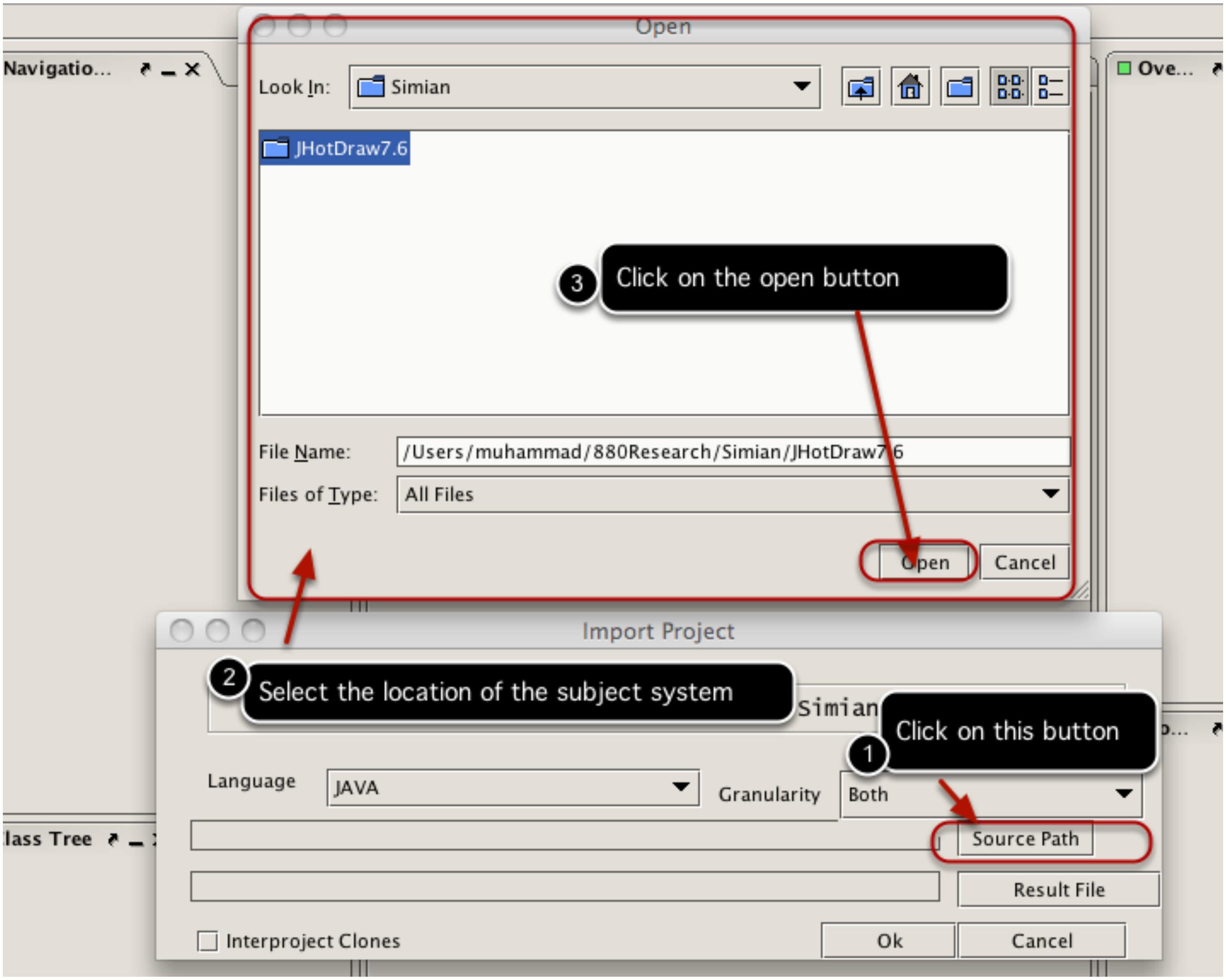
Suppose your computer contians the source code of JHotDraw(version 7.6) and located in the JHotDraw7.6 directory. You detect the clones using Simian on that directory and stores the result in jhotdrawResultSimian.txt file. To import the result, you need to click on the Project menu and select the *Import Result from Simian* menu item.

Importing clone detection result of Simian(Continued)



This opens the above dialog box.

Select subject system



3 Click on the open button

2 Select the location of the subject system

1 Click on this button

Open

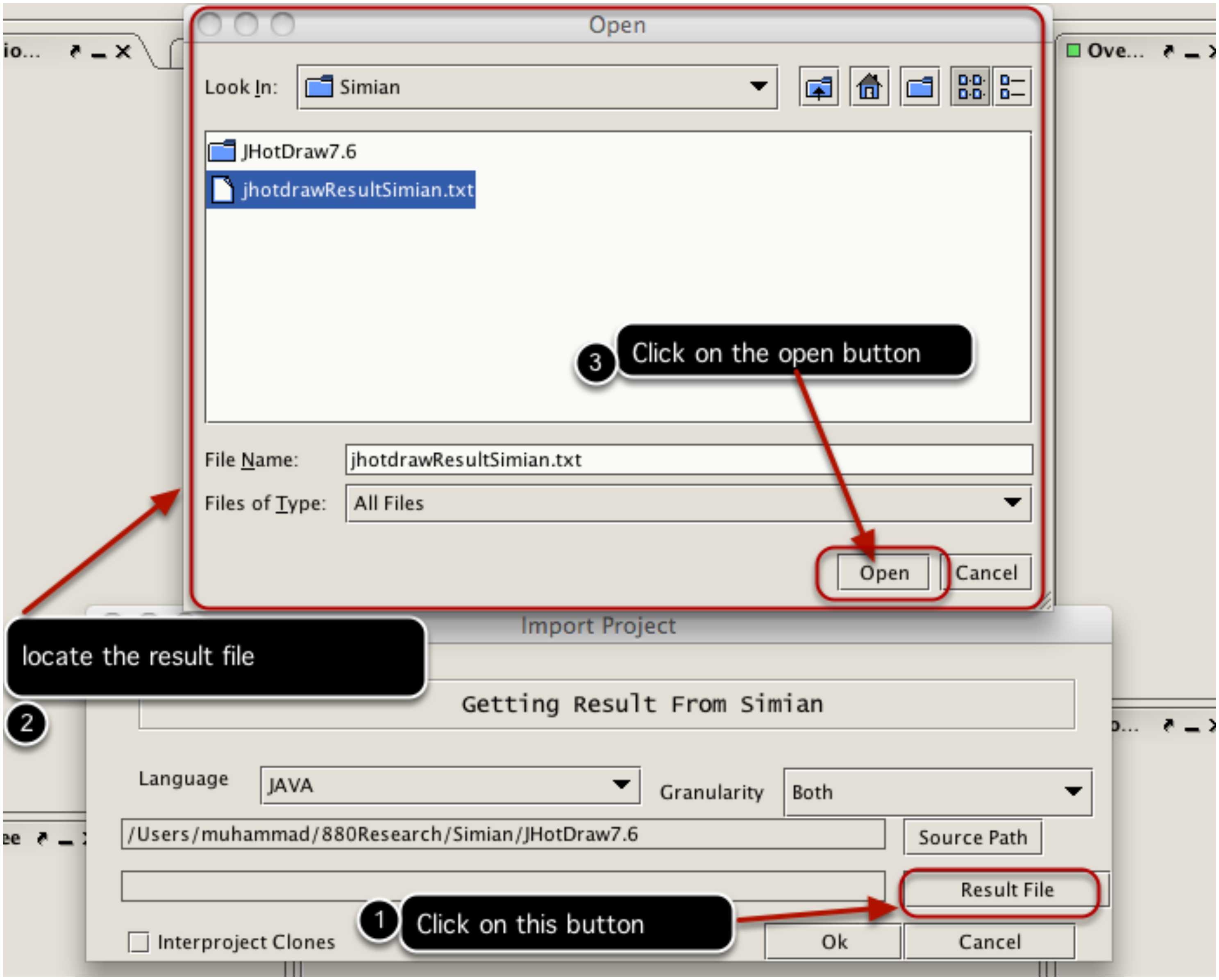
Look In: Simian

JHotDraw7.6

File Name: /Users/muhammad/880Research/Simian/JHotDraw7.6

Files of Type: All Files

Select the clone detection result file



1 Click on this button

2 locate the result file

3 Click on the open button

Look In: Simian

File Name: jhotdrawResultSimian.txt

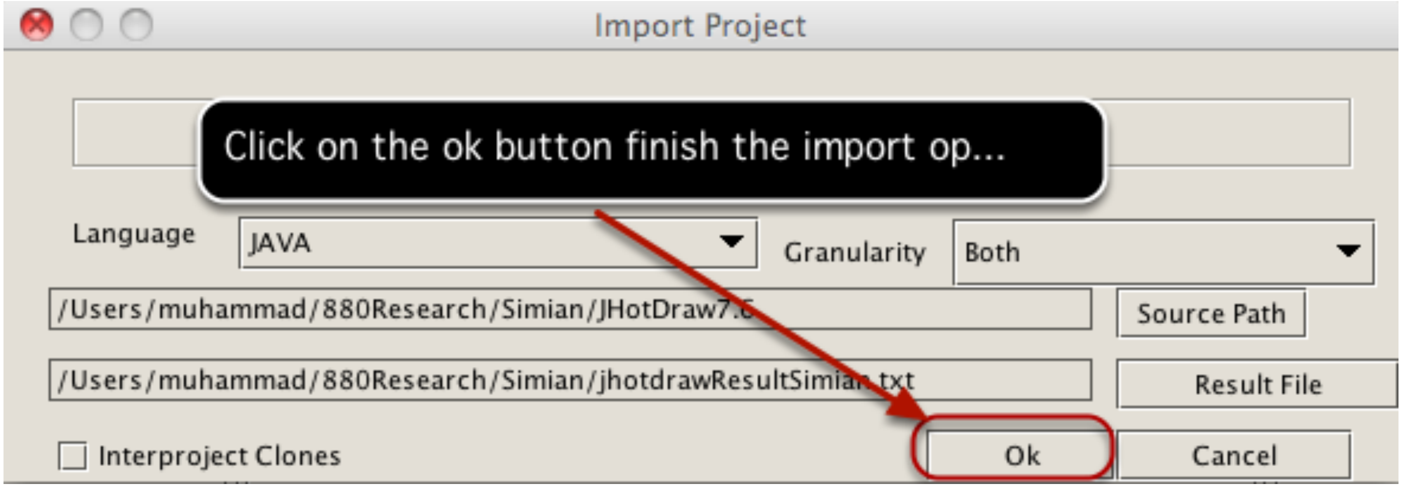
Files of Type: All Files

Language: JAVA

Granularity: Both

/Users/muhammad/880Research/Simian/JHotDraw7.6

Complete the import opeation



Click on the ok button finish the import op...

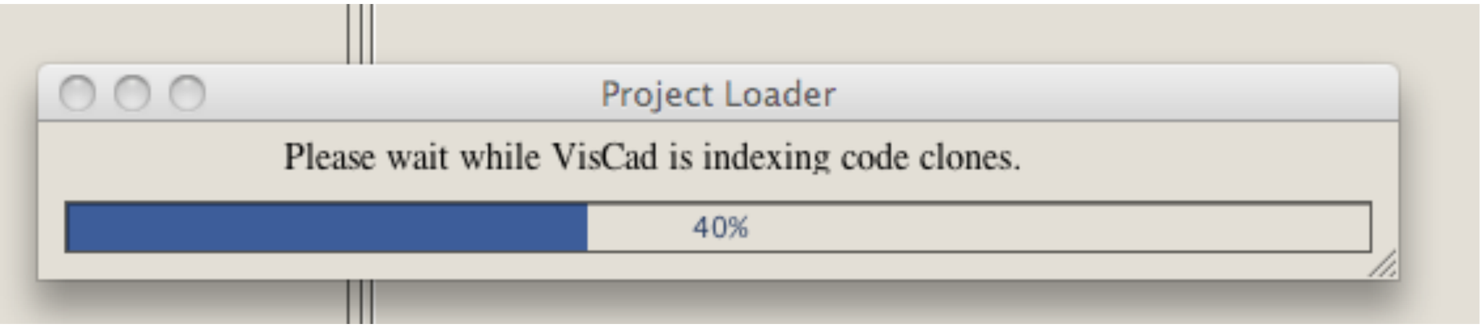
Language: JAVA

Granularity: Both

/Users/muhammad/880Research/Simian/JHotDraw7.6

/Users/muhammad/880Research/Simian/jhotdrawResultSimian.txt

Loading clone detection result

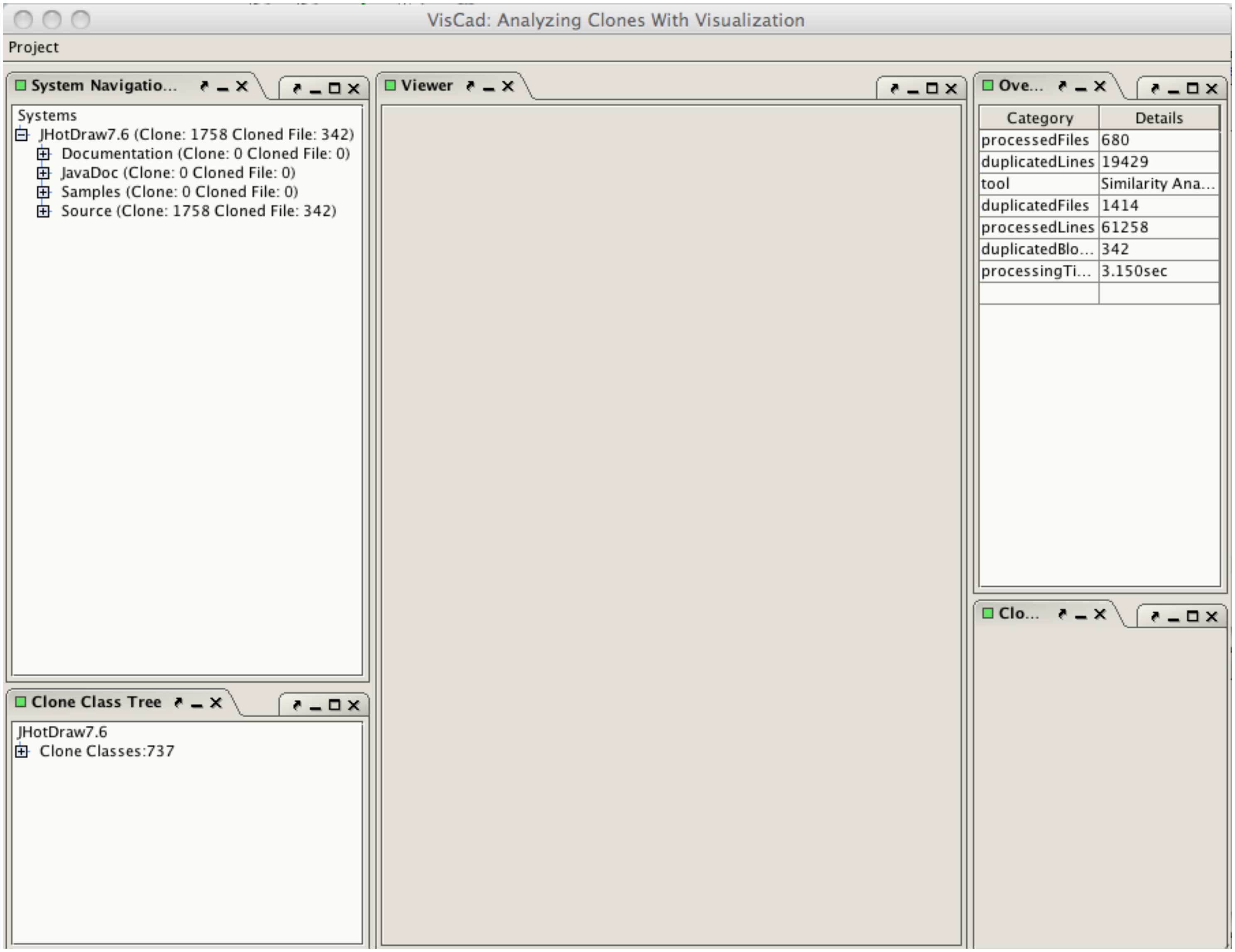


Please wait while VisCad is indexing code clones.

40%

VisCad is now loading the clone detection result.

Complete the loading operation



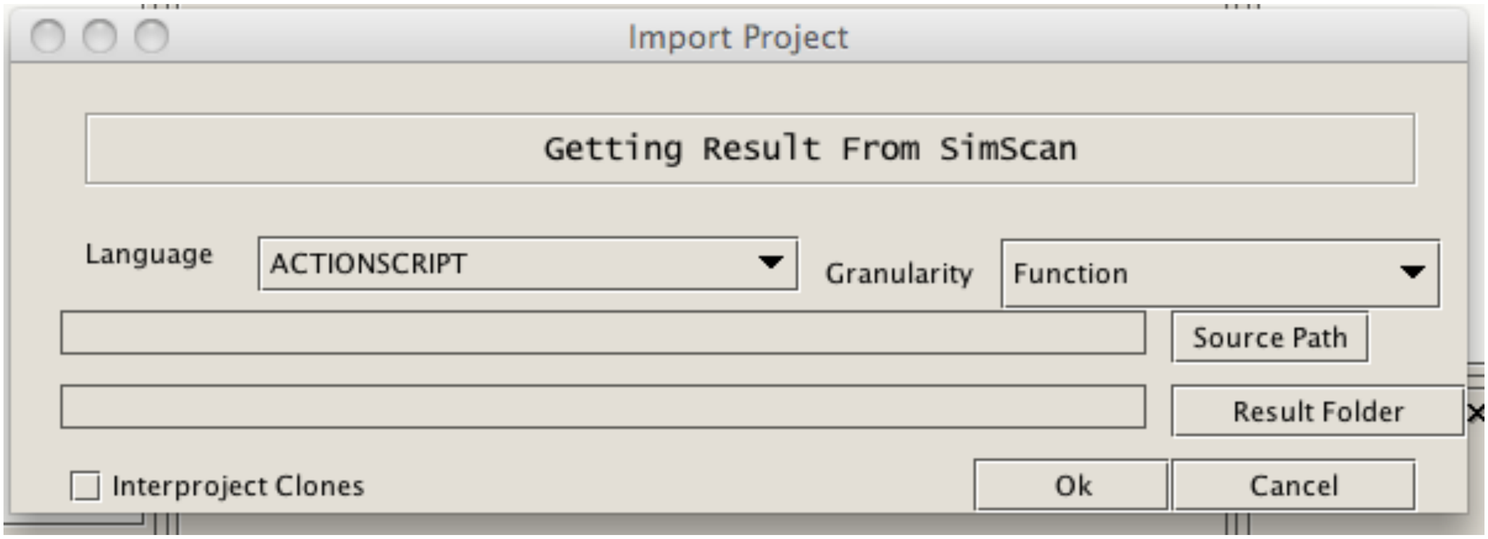
Systems
- JHotDraw7.6 (Clone: 1758 Cloned File: 342)
  - Documentation (Clone: 0 Cloned File: 0)
  - JavaDoc (Clone: 0 Cloned File: 0)
  - Samples (Clone: 0 Cloned File: 0)
  - Source (Clone: 1758 Cloned File: 342)

Category	Details
processedFiles	680
duplicatedLines	19429
tool	Similarity Ana...
duplicatedFiles	1414
processedLines	61258
duplicatedBlo...	342
processingTi...	3.150sec

JHotDraw7.6
- Clone Classes:737

The result is loaded into VisCad.

Import result from SimScan



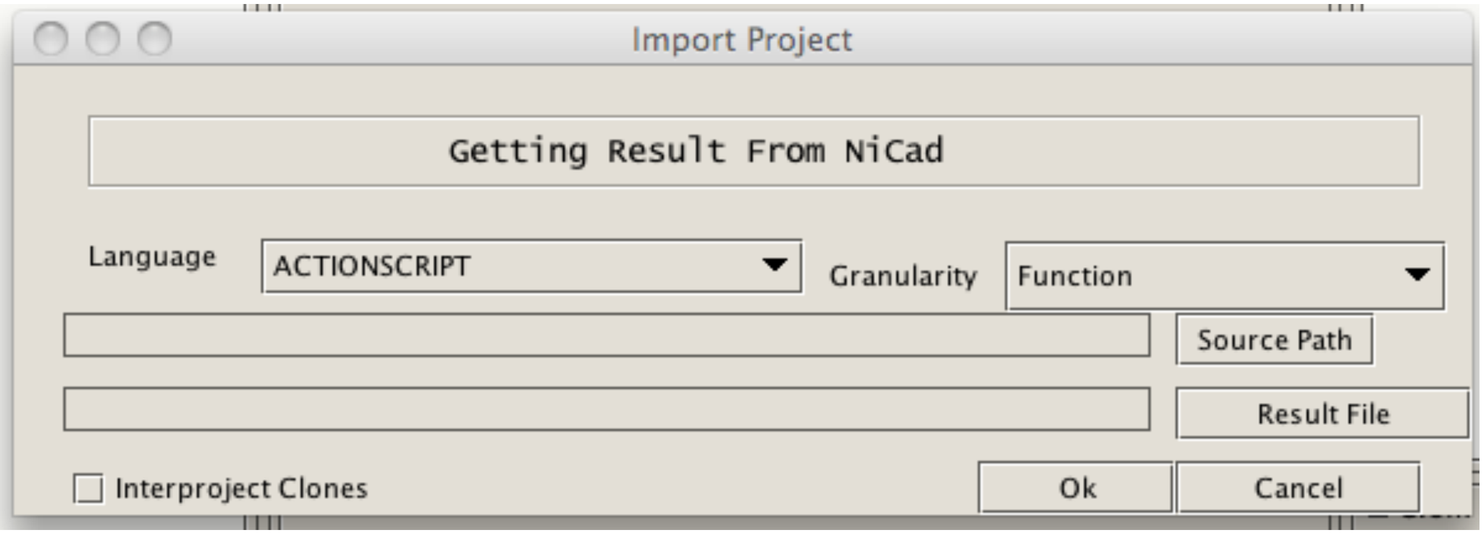
Language: ACTIONSCRIPT
Granularity: Function

SimScan creates a folder that contains the output of the clone detection. A sample clone detection result (jhotdrawReportSimScan.zip) for JHotDraw (Version 7.6) is included in the VisCadBeta.zip file. You can also find the source code of the subject system in a zip file (jhotdraw-7.6.nested.zip).

Follow the steps listed below:

1. Unzip the jhotdraw-7.6.nested.zip file and rename the folder to JHotDraw7.6.
2. Unzip the jhotdrawReportSimScan.zip file. This will create a folder (jhotdrawReportSimScan) that contains the clone detection results. Put the folder in the same place where JHotDraw7.6 folder resides.
3. Select the ‘Import Result from SimScan’ menu item from the ‘Project’ menu (located on the top left of the VisCad interface). This opens a dialog to import the result.
4. Click on the Source Path button and select the source code directory (In this case, JHotDraw7.6 folder). This should be the folder on which you apply clone detection.
5. Now, click on the Result Folder button and select the clone detection result folder (In this case, jhotdrawReportSimScan).
6. Select the language of the subject system and also the granularity of clone detection. If you are not sure, you can leave them as default.
7. Now, click on the Ok button to load the subject system by VisCad. The loading time may vary depending on the size of the system.
8. You can click on the Cancel button to cancel the import operation.

Import result from NiCad

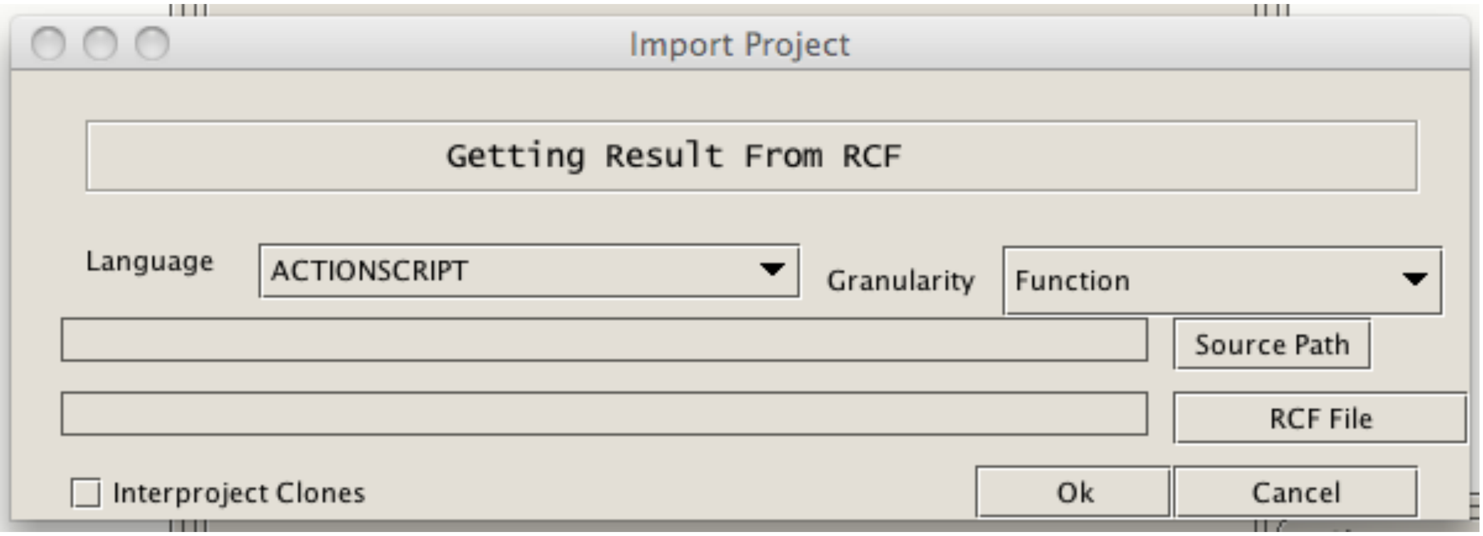


A sample clone detection result (JHotDraw7.6_functions-clones-0.3.xml) for JHotDraw (Version 7.6) is included in the zip file. You can also find the source code of the subject system in a zip file (jhotdraw-7.6.nested.zip). Clones were detected using NiCad.

Follow the steps listed below:

1. Unzip the jhotdraw-7.6.nested.zip file and rename the folder to JHotDraw7.6.
2. Put the clone detection result file in the same directory where JHotDraw7.6 folder resides.
3. Select the ‘Import Result from NiCad’ menu item from the ‘Project’ menu (located on the top left of the VisCad interface). This opens a dialog to import the result.
4. Click on the Source Path button and select the source code directory (In this case, JHotDraw7.6 folder). This should be the folder on which you apply clone detection.
5. Now, Click on the Result File button and select the clone detection result file (In this case, JHotDraw7.6_functions-clones-0.3.xml).
6. Select the language of the subject system and also the granularity of clone detection. If you are not sure, you can leave them as default.
7. Now, click on the Ok button to load the subject system by VisCad. The loading time may vary depending on the size of the system.
8. You can click on the Cancel button to cancel the import operation.

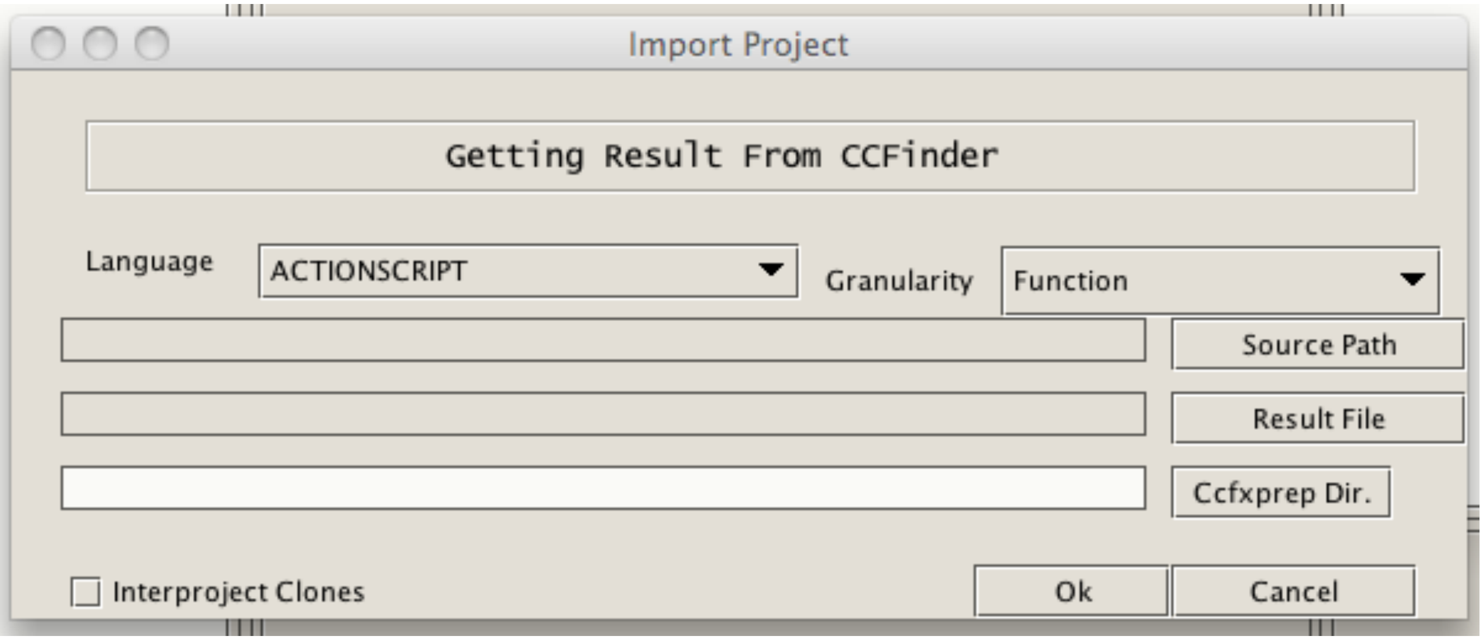
Import RCF file



RCF is a data format that can store clone data for a number of versions of a subject system.
For this example, we have used GNU Wget. Download both the subject system and the RCF file from here. The RCF file is packaged with the source code of the subject system.
Follow the steps listed below:

1. Extract the content of the archive and rename the folder to rcfResult. You now have two folders inside the rcfResult folder. One is the wget that contains the source code of seven different versions of the system. Another one is the wget.rcf that contains the clone detection results of those systems in the RCF format.
2. Select the ‘Import RCF’ menu item from the ‘Project’ menu (located on the top left of the VisCad interface). This opens a dialog to import the result.
3. Click on the Source Path button and select the wget folder located inside the rcfResult folder.
4. Now, click on the RCF File button and select the wget.rcf file.
5. Select the language of the subject system and also the granularity of clone detection. If you are not sure, you can leave them as default.
6. Now, click on the Ok button to load all seven-subject subjects by VisCad. The loading time may vary depending on the number of subject systems.
7. You can click on the Cancel button to cancel the import operation.

Import result from CCFinder

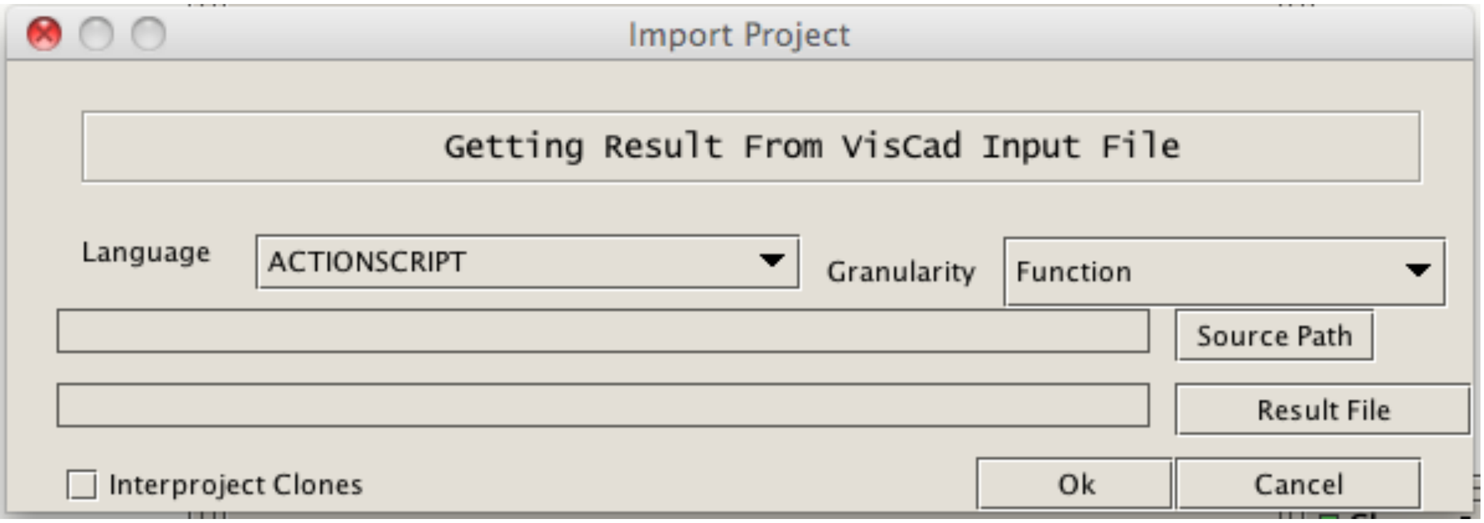


During the clone detection, CCFinder creates a directory (.ccfxprepdir). If you apply clone detection on JHotDraw7.6 directory, then the .ccfxprepdir will be created right inside that directory.

Follow the steps listed below:

1. Unzip the jhotdraw-7.6.nested.zip file and rename the folder to JHotDraw7.6. Detect clones with CCFinder and then rename the .ccfxprepdir directory to ccfxprepdir.
2. The output file is a binary one (such as a.ccfxd). You need to convert that into a text file (see here). From now, we will refer the text file as the result file.
3. Put the clone detection result file in the same directory where JHotDraw7.6 folder resides.
4. Select the 'Import CCFinder Project' menu item from the 'Project' menu (located on the top left of the VisCad interface). This opens a dialog to import the result.
5. Click on the Source Path button and select the source code directory (In this case, JHotDraw7.6 folder). This should be the folder on which you apply clone detection.
6. Now, click on the Result File button and select the clone detection result file.
7. Click on the Ccfxprep Dir. button and select the ccfxprepdir directory.
8. Select the language of the subject system and also the granularity of clone detection. If you are not sure, you can leave them as default.
9. Now, click on the Ok button to load the subject system by VisCad. The loading time may vary depending on the size of the system.
10. You can click on the Cancel button to cancel the import operation.

Import VisCad input file



Language ACTIONSCRIPT

Granularity Function

☐ Interproject Clones

For any other clone detection tools, the result file needs to be converted to the VisCad input file format. A sample clone detection result (resultViscadInputFile.txt) for JHotDraw (Version 7.6) is included in the VisCadBeta.zip file. Clones were detected using Simian.

Follow the steps listed below:

1. Unzip the jhotdraw-7.6.nested.zip file and rename the folder to JHotDraw7.6.
2. Put the clone detection result file in the same directory where JHotDraw7.6 folder resides.
3. Select the ‘Import VisCad Input File’ menu item from the ‘Project’ menu (located on the top left of the VisCad interface). This opens a dialog to import the result.
4. Click on the Source Path button and select the source code directory (In this case, JHotDraw7.6 folder). This should be the folder on which you apply clone detection.
5. Now, click on the Result File button and select the clone detection result file (In this case, resultViscadInputFile.txt).
6. Select the language of the subject system and also the granularity of clone detection. If you are not sure, you can leave them as default.
7. Now, click on the Ok button to load the subject system by VisCad. The loading time may vary depending on the size of the system.
8. You can click on the Cancel button to cancel the import operation.

User Interface Components

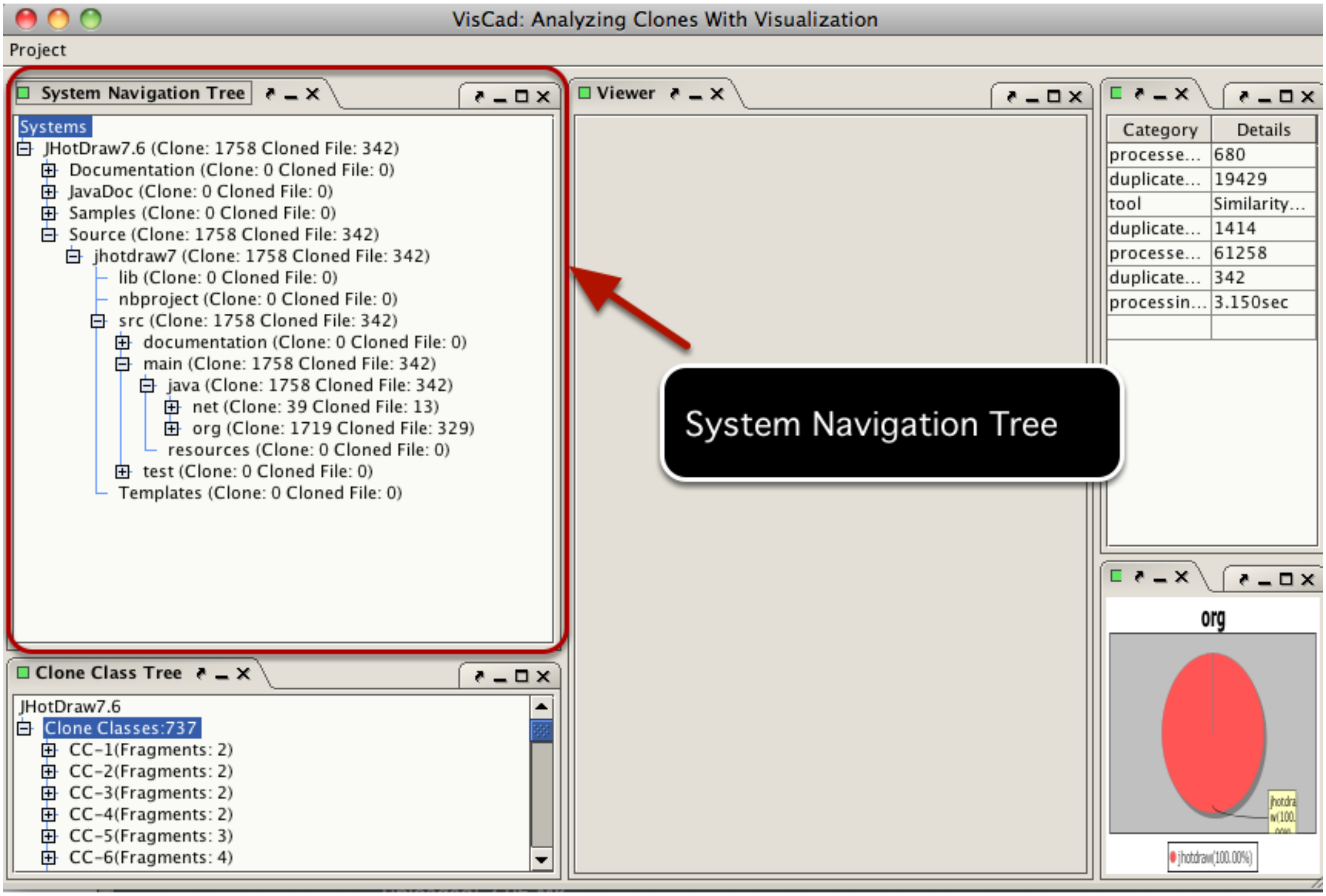
The main user interface of VisCad can be divided into three parts.

1. Left Part: The left part accommodates the *clone browser*. The clone browser has two parts, one of which displays the distribution of clones over the directories and sub- directories in the subject system, known as *System Navigation Tree*. The other part, located on the bottom of the clone browser, lists all *clone classes* and the number of clone snippets in each class, called *Clone Class Tree*.

2. Middle Part: The middle part of VisCad accommodates different views in separate tabs. We will refers this part as *Viewer*.

3. Right Part: The top- right window shows the clone detection specific information VisCad obtained while parsing the result file for the selected
subject system. For any selected directory in the *system navigation tree*, the bottom-right window shows the distribution of clones in its sub-directories through a pie chart.

System Navigation Tree



```
Systems
⊟ JHotDraw7.6 (Clone: 1758 Cloned File: 342)
   ⊞ Documentation (Clone: 0 Cloned File: 0)
   ⊞ JavaDoc (Clone: 0 Cloned File: 0)
   ⊞ Samples (Clone: 0 Cloned File: 0)
   ⊟ Source (Clone: 1758 Cloned File: 342)
      ⊟ jhotdraw7 (Clone: 1758 Cloned File: 342)
         lib (Clone: 0 Cloned File: 0)
         nbproject (Clone: 0 Cloned File: 0)
         ⊟ src (Clone: 1758 Cloned File: 342)
            ⊞ documentation (Clone: 0 Cloned File: 0)
            ⊟ main (Clone: 1758 Cloned File: 342)
               ⊟ java (Clone: 1758 Cloned File: 342)
                  ⊞ net (Clone: 39 Cloned File: 13)
                  ⊞ org (Clone: 1719 Cloned File: 329)
                  resources (Clone: 0 Cloned File: 0)
            ⊞ test (Clone: 0 Cloned File: 0)
         Templates (Clone: 0 Cloned File: 0)
```

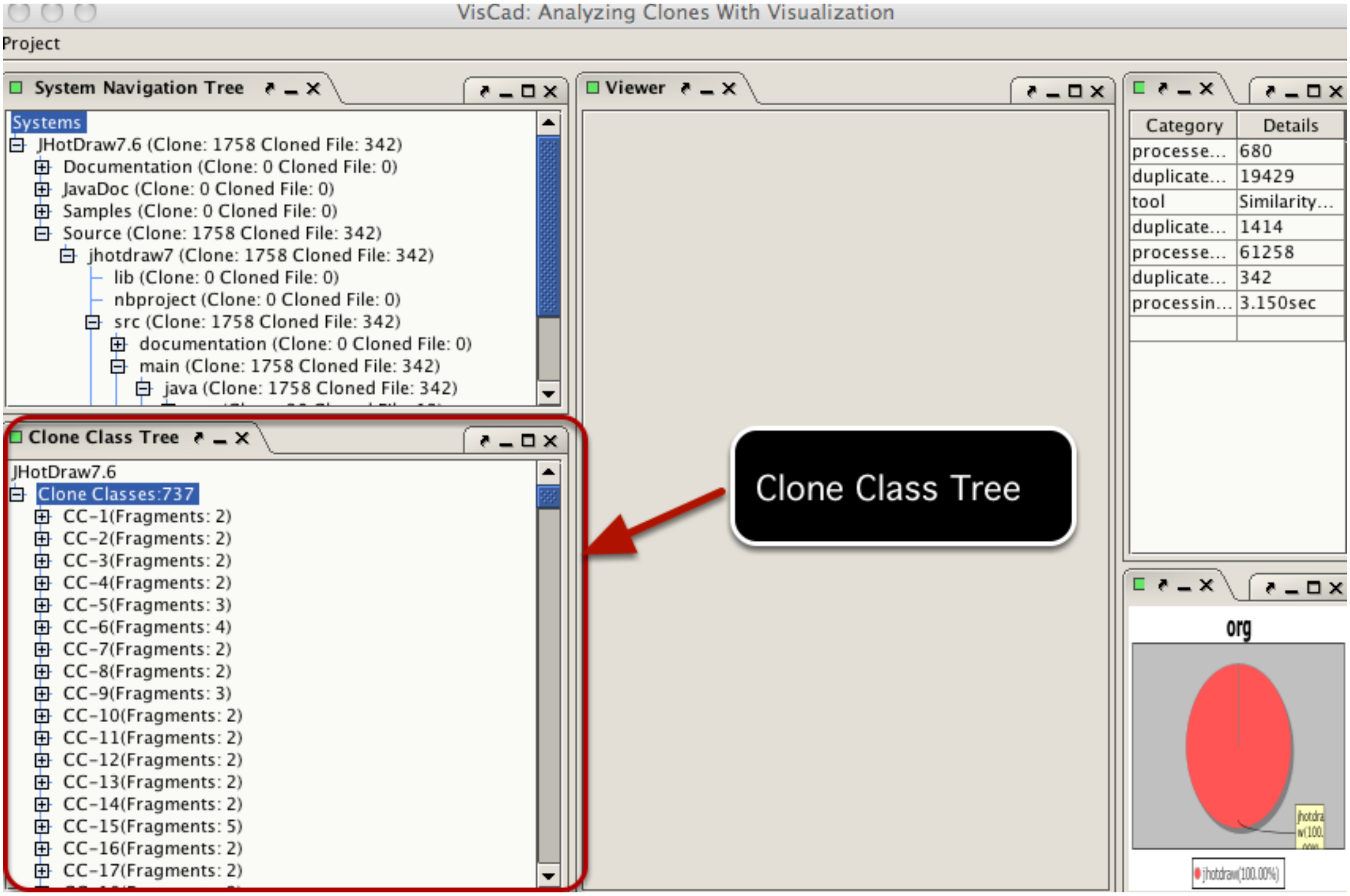
```
JHotDraw7.6
⊟ Clone Classes:737
   ⊞ CC-1(Fragments: 2)
   ⊞ CC-2(Fragments: 2)
   ⊞ CC-3(Fragments: 2)
   ⊞ CC-4(Fragments: 2)
   ⊞ CC-5(Fragments: 3)
   ⊞ CC-6(Fragments: 4)
```

Category	Details
processe...	680
duplicate...	19429
tool	Similarity...
duplicate...	1414
processe...	61258
duplicate...	342
processin...	3.150sec

System Navigation Tree

It maps clones to files and directories

Clone Class Tree



Systems
JHotDraw7.6 (Clone: 1758 Cloned File: 342)
  Documentation (Clone: 0 Cloned File: 0)
  JavaDoc (Clone: 0 Cloned File: 0)
  Samples (Clone: 0 Cloned File: 0)
  Source (Clone: 1758 Cloned File: 342)
    jhotdraw7 (Clone: 1758 Cloned File: 342)
      lib (Clone: 0 Cloned File: 0)
      nbproject (Clone: 0 Cloned File: 0)
      src (Clone: 1758 Cloned File: 342)
        documentation (Clone: 0 Cloned File: 0)
        main (Clone: 1758 Cloned File: 342)
          java (Clone: 1758 Cloned File: 342)

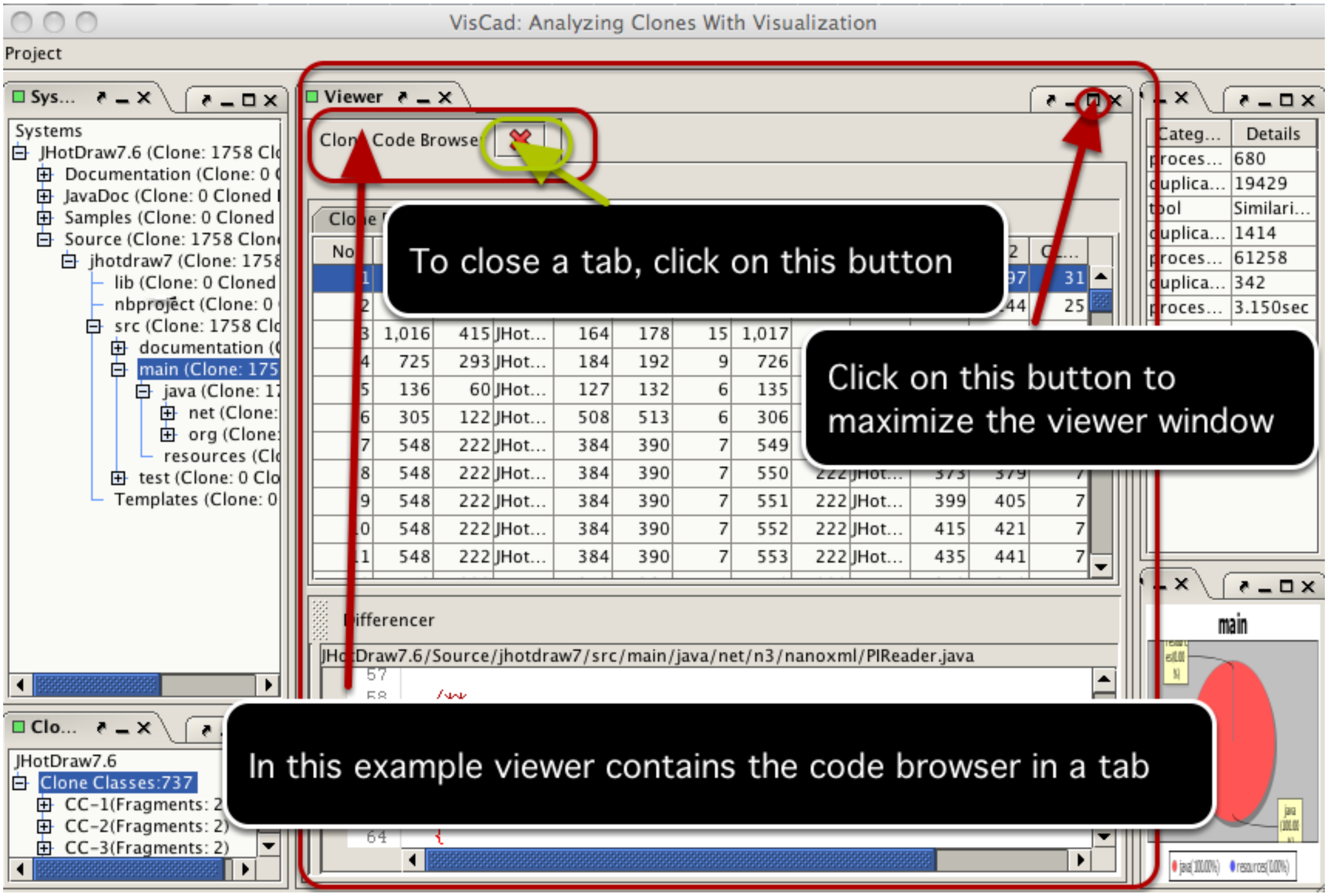
Category	Details
processe...	680
duplicate...	19429
tool	Similarity...
duplicate...	1414
processe...	61258
duplicate...	342
processin...	3.150sec

JHotDraw7.6
Clone Classes:737
  CC-1(Fragments: 2)
  CC-2(Fragments: 2)
  CC-3(Fragments: 2)
  CC-4(Fragments: 2)
  CC-5(Fragments: 3)
  CC-6(Fragments: 4)
  CC-7(Fragments: 2)
  CC-8(Fragments: 2)
  CC-9(Fragments: 3)
  CC-10(Fragments: 2)
  CC-11(Fragments: 2)
  CC-12(Fragments: 2)
  CC-13(Fragments: 2)
  CC-14(Fragments: 2)
  CC-15(Fragments: 5)
  CC-16(Fragments: 2)
  CC-17(Fragments: 2)

Clone Class Tree

Clone class tree groups the detected clones into different classes.

Viewer



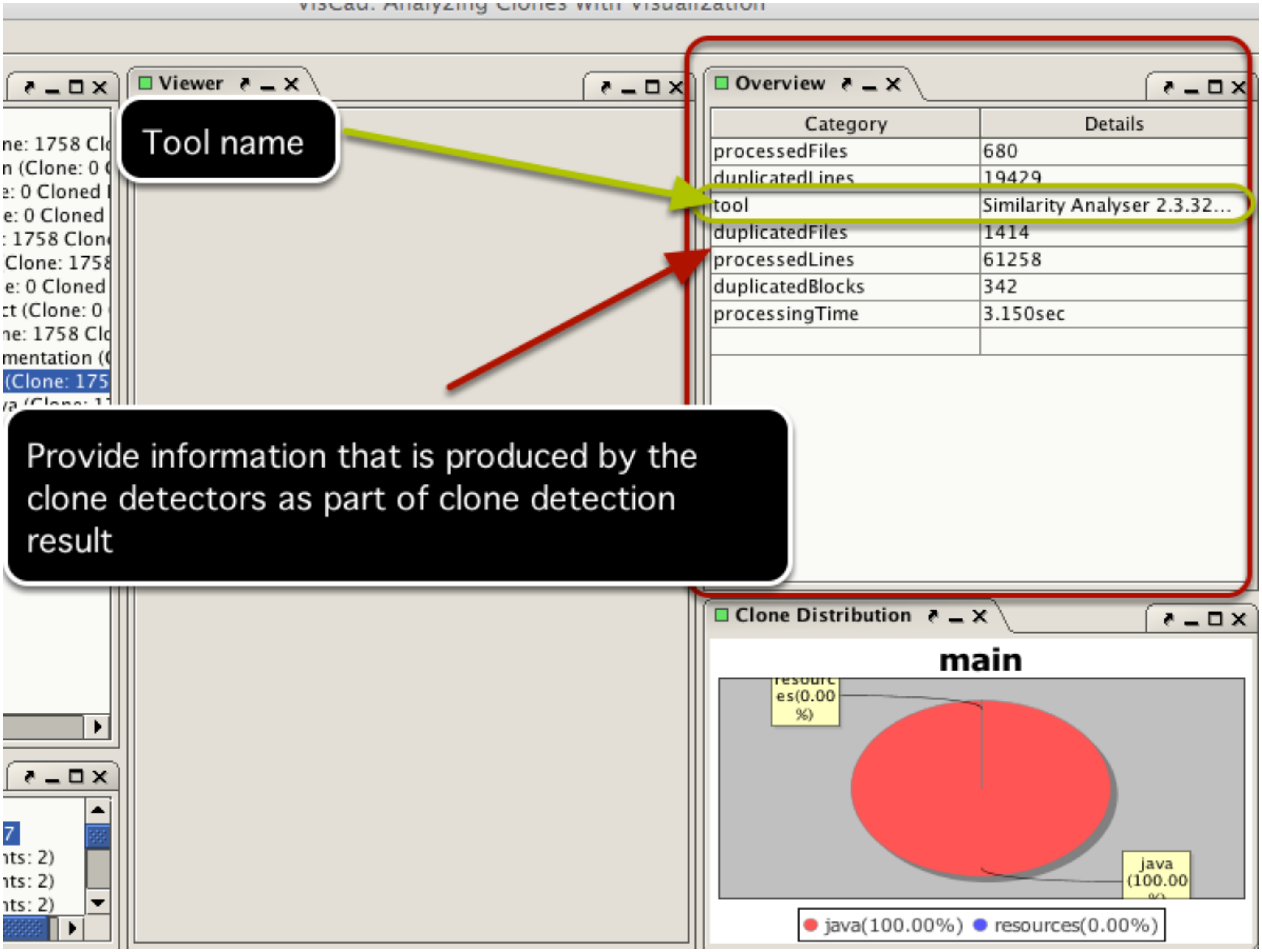
To close a tab, click on this button

Click on this button to maximize the viewer window

In this example viewer contains the code browser in a tab

Depending on the user's selection, it accomodates different views(such as the scatter plot, treemap, hierarchical dependency graph, source code browser etc.)

Obtaining information specific to clone detection



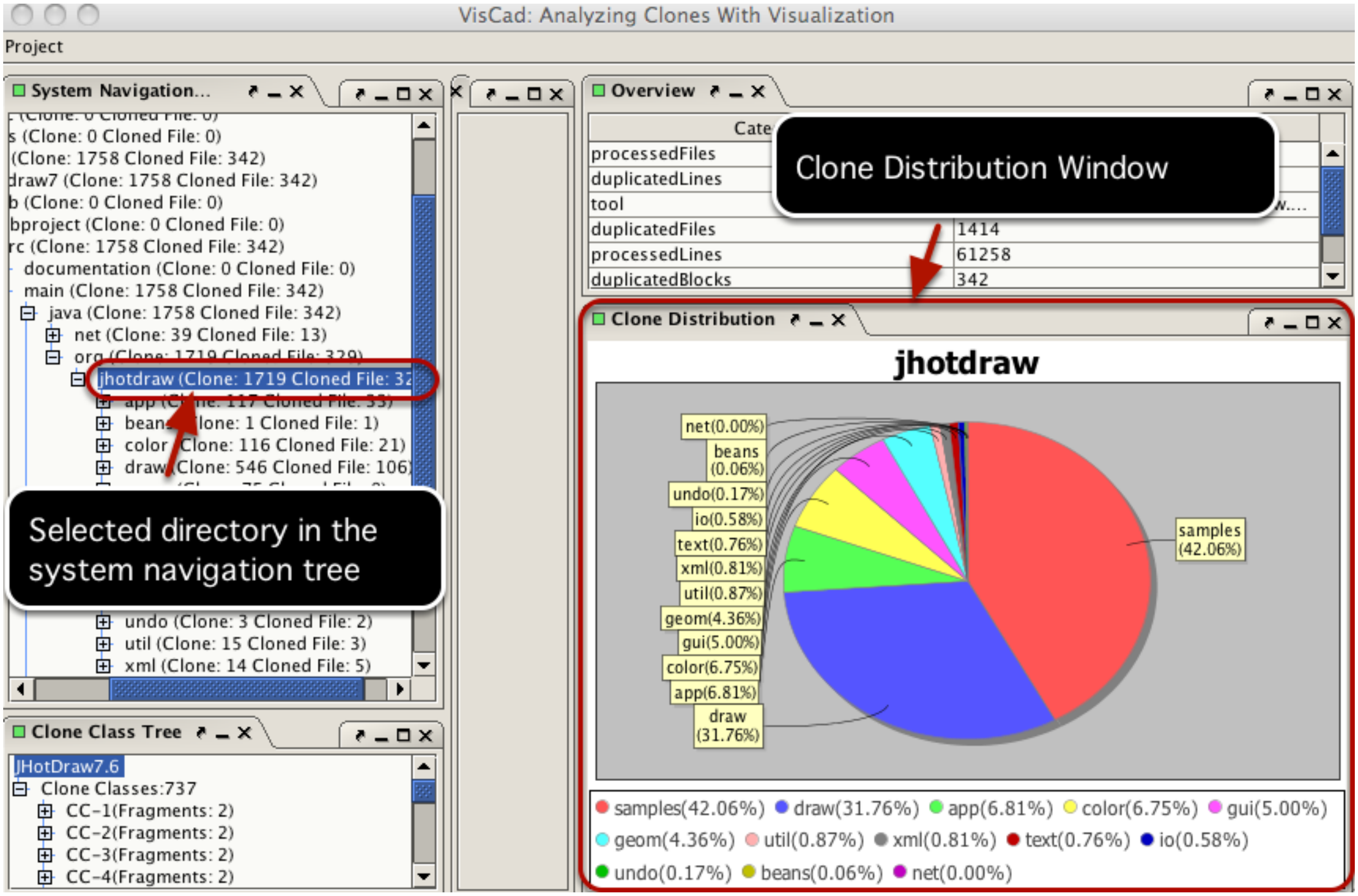
Tool name

Provide information that is produced by the clone detectors as part of clone detection result

Category	Details
processedFiles	680
duplicatedLines	19429
tool	Similarity Analyser 2.3.32...
duplicatedFiles	1414
processedLines	61258
duplicatedBlocks	342
processingTime	3.150sec

This part contains information such as tool name, overview of clone detection result.

Clone distribution window



s (Clone: 0 Cloned File: 0)
(Clone: 1758 Cloned File: 342)
draw7 (Clone: 1758 Cloned File: 342)
b (Clone: 0 Cloned File: 0)
bproject (Clone: 0 Cloned File: 0)
rc (Clone: 1758 Cloned File: 342)
documentation (Clone: 0 Cloned File: 0)
main (Clone: 1758 Cloned File: 342)
java (Clone: 1758 Cloned File: 342)
net (Clone: 39 Cloned File: 13)
org (Clone: 1719 Cloned File: 329)
jhotdraw (Clone: 1719 Cloned File: 32
app (Clone: 117 Cloned File: 33)
beans (Clone: 1 Cloned File: 1)
color (Clone: 116 Cloned File: 21)
draw (Clone: 546 Cloned File: 106)
undo (Clone: 3 Cloned File: 2)
util (Clone: 15 Cloned File: 3)
xml (Clone: 14 Cloned File: 5)

Cate	
processedFiles	
duplicatedLines	
tool	w....
duplicatedFiles	1414
processedLines	61258
duplicatedBlocks	342

JHotDraw7.6
Clone Classes:737
CC-1(Fragments: 2)
CC-2(Fragments: 2)
CC-3(Fragments: 2)
CC-4(Fragments: 2)

Clone Distribution Window

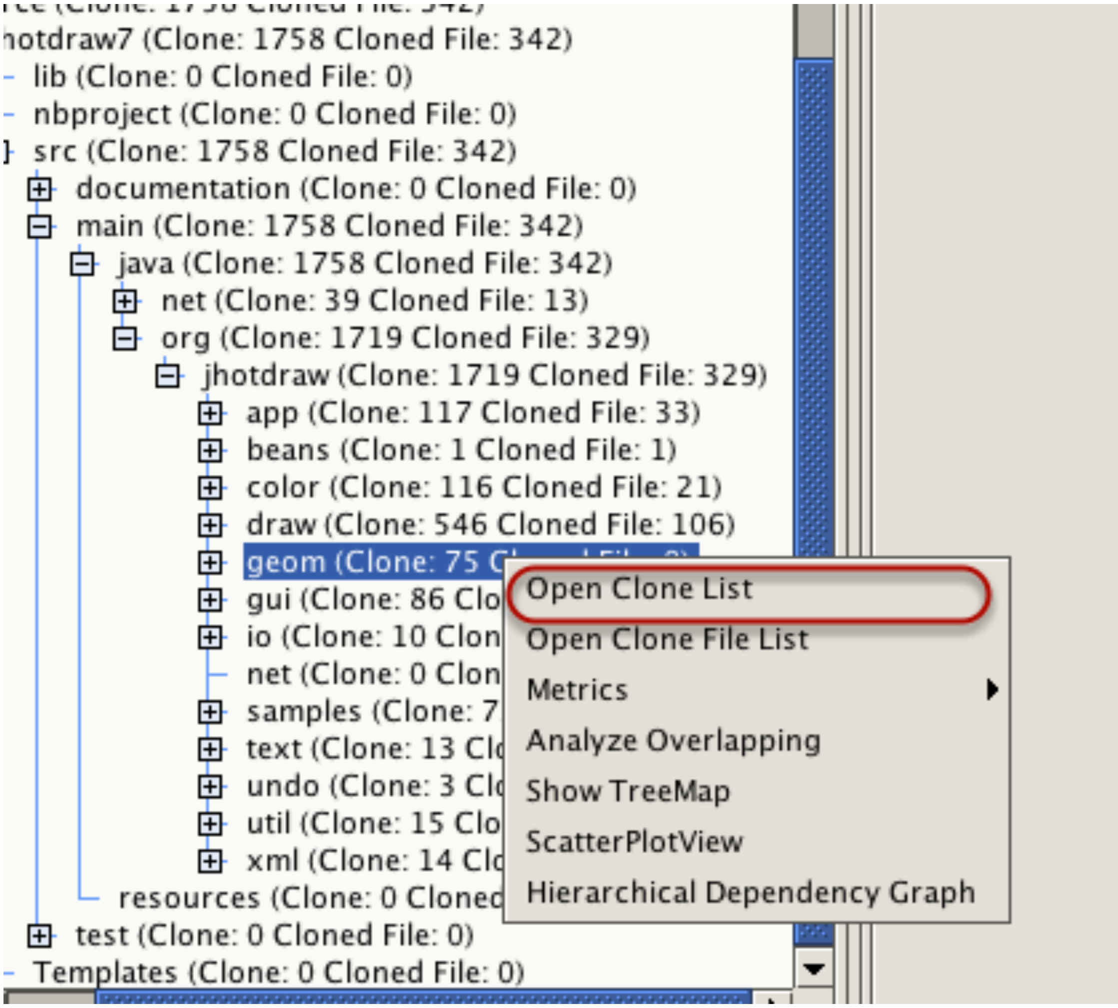
Selected directory in the system navigation tree

Click on a directory in the system navigation tree to select it. Based on the selection, the clone distribution window updates and shows the distribution of clones in its subdirectories through a pie chart.

Analyze Clone Fragments

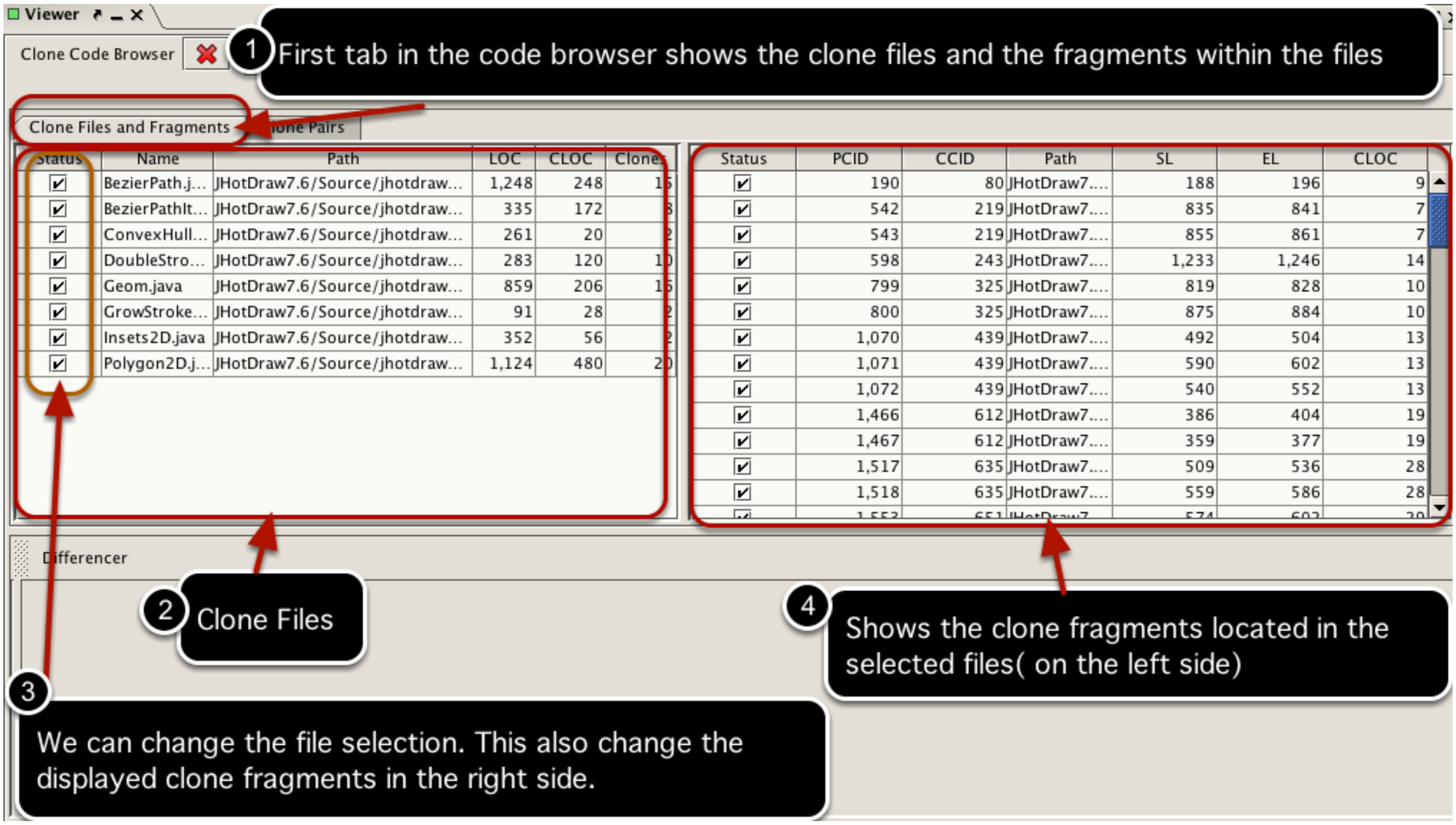
You can analyze the source code of the clone fragments using the *code browser*. The same component is used for analyzing clone code fragments in other places also.

Open the popup menu and make selection



If we want to analyze the clone fragments on the *geom directory*, we need to select the directory, right click on the mouse button to bring up the popup menu and select the *Open clone list* menu item.

The code browser



① First tab in the code browser shows the clone files and the fragments within the files

Status	Name	Path	LOC	CLOC	Clones
☑	BezierPath.j...	JHotDraw7.6/Source/jhotdraw...	1,248	248	15
☑	BezierPathIt...	JHotDraw7.6/Source/jhotdraw...	335	172	8
☑	ConvexHull...	JHotDraw7.6/Source/jhotdraw...	261	20	2
☑	DoubleStro...	JHotDraw7.6/Source/jhotdraw...	283	120	10
☑	Geom.java	JHotDraw7.6/Source/jhotdraw...	859	206	15
☑	GrowStroke...	JHotDraw7.6/Source/jhotdraw...	91	28	2
☑	Insets2D.java	JHotDraw7.6/Source/jhotdraw...	352	56	2
☑	Polygon2D.j...	JHotDraw7.6/Source/jhotdraw...	1,124	480	20

Status	PCID	CCID	Path	SL	EL	CLOC
☑	190	80	JHotDraw7....	188	196	9
☑	542	219	JHotDraw7....	835	841	7
☑	543	219	JHotDraw7....	855	861	7
☑	598	243	JHotDraw7....	1,233	1,246	14
☑	799	325	JHotDraw7....	819	828	10
☑	800	325	JHotDraw7....	875	884	10
☑	1,070	439	JHotDraw7....	492	504	13
☑	1,071	439	JHotDraw7....	590	602	13
☑	1,072	439	JHotDraw7....	540	552	13
☑	1,466	612	JHotDraw7....	386	404	19
☑	1,467	612	JHotDraw7....	359	377	19
☑	1,517	635	JHotDraw7....	509	536	28
☑	1,518	635	JHotDraw7....	559	586	28

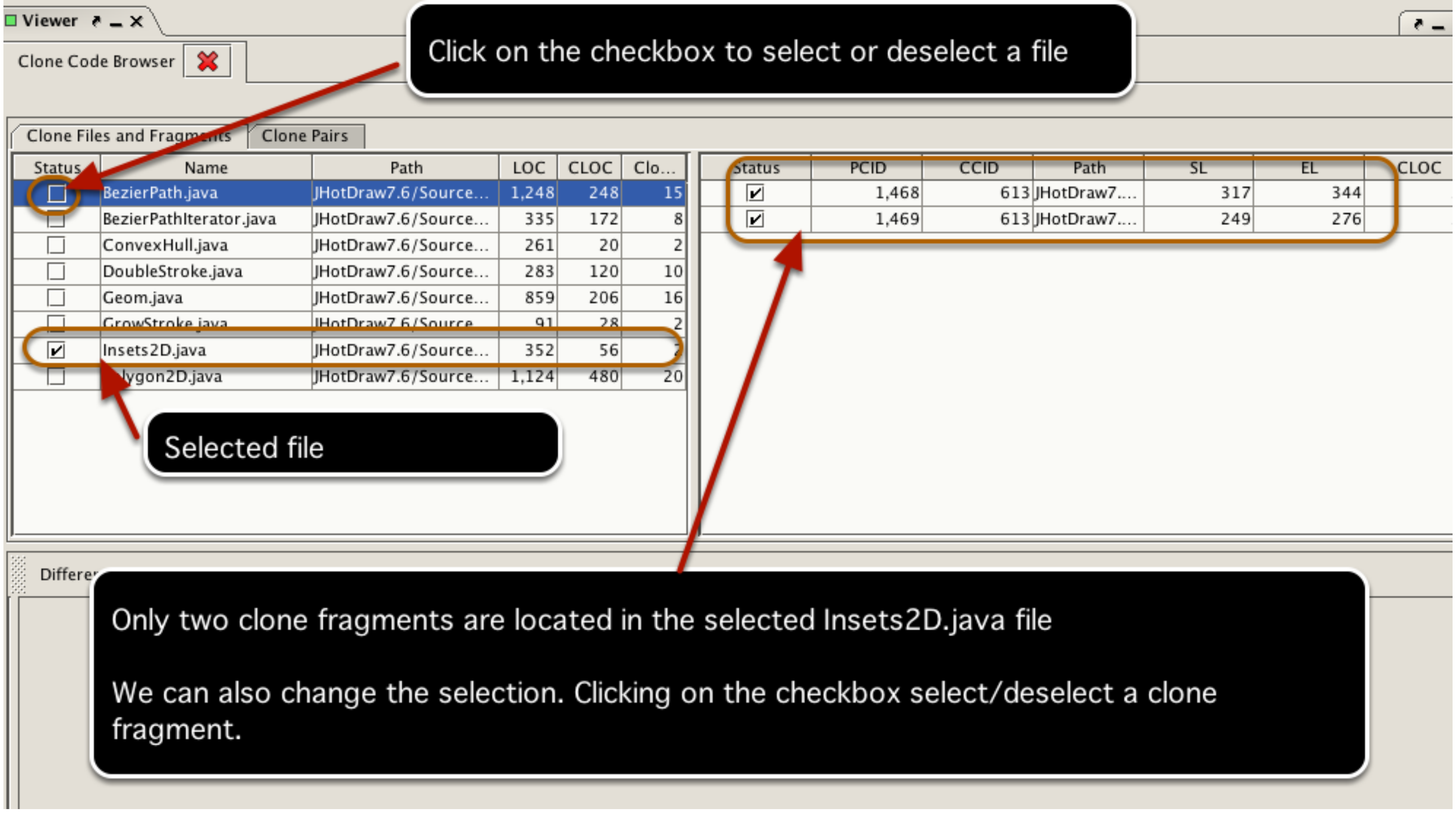
② Clone Files

③ We can change the file selection. This also change the displayed clone fragments in the right side.

④ Shows the clone fragments located in the selected files(on the left side)

The previous selection open the code browser in the *viewer* window in a new tab.

An Example



Click on the checkbox to select or deselect a file

Status	Name	Path	LOC	CLOC	Clo...
☐	BezierPath.java	JHotDraw7.6/Source...	1,248	248	15
☐	BezierPathIterator.java	JHotDraw7.6/Source...	335	172	8
☐	ConvexHull.java	JHotDraw7.6/Source...	261	20	2
☐	DoubleStroke.java	JHotDraw7.6/Source...	283	120	10
☐	Geom.java	JHotDraw7.6/Source...	859	206	16
☐	GrowStroke.java	JHotDraw7.6/Source...	91	28	2
☑	Insets2D.java	JHotDraw7.6/Source...	352	56	2
☐	[illegible]lygon2D.java	JHotDraw7.6/Source...	1,124	480	20

Selected file

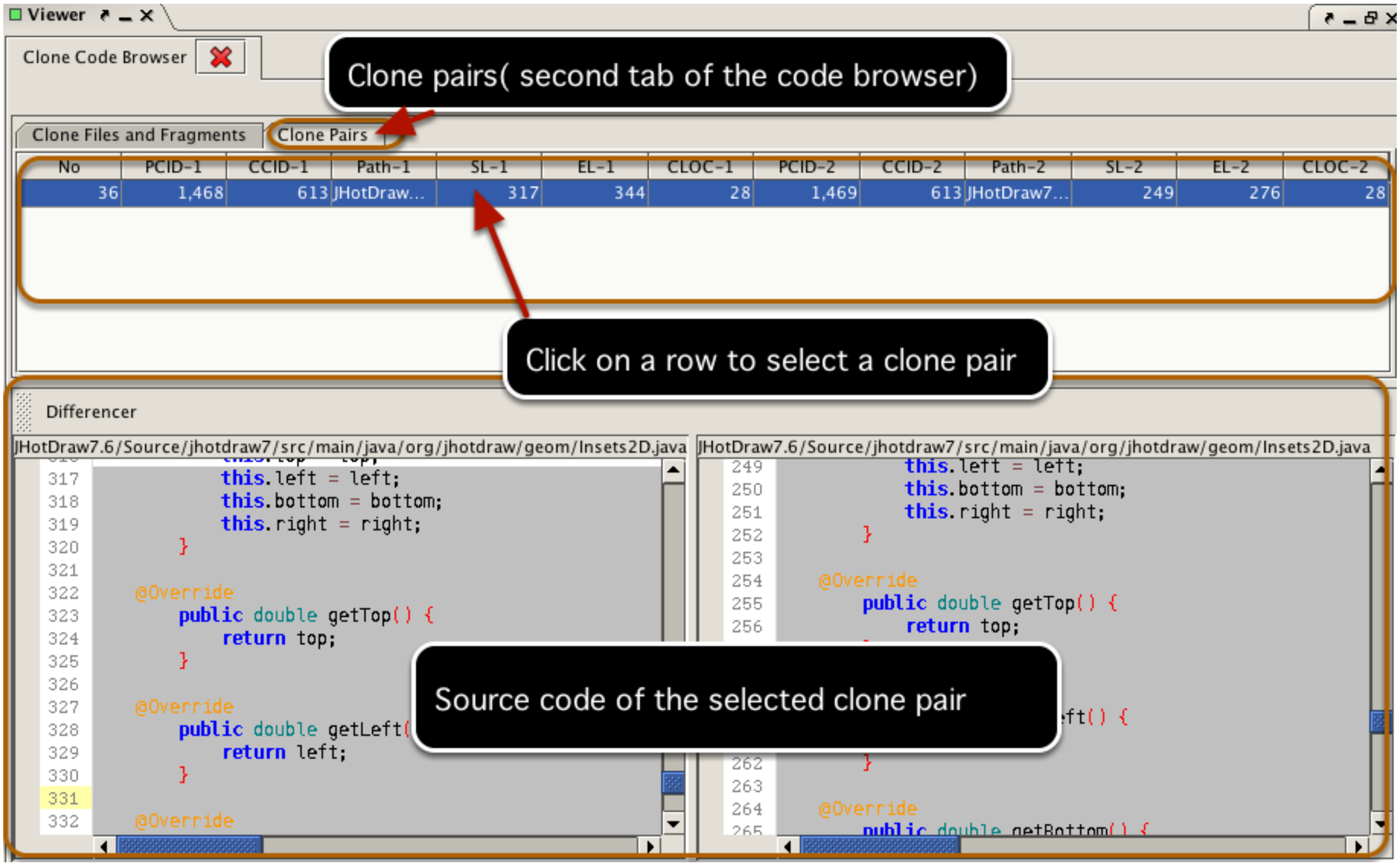
Status	PCID	CCID	Path	SL	EL	CLOC
☑	1,468	613	JHotDraw7....	317	344	
☑	1,469	613	JHotDraw7....	249	276	

Only two clone fragments are located in the selected Insets2D.java file

We can also change the selection. Clicking on the checkbox select/deselect a clone fragment.

Suppose, we want to analyze the clone fragments located only in the Insets2D. java file.
We need to deselect all files except Insets2D.java file by clicking on the checkboxes. From the right side, we can see that only two code fragments are located in this file.

Examining clone pairs



Clone pairs(second tab of the code browser)

No	PCID-1	CCID-1	Path-1	SL-1	EL-1	CLOC-1	PCID-2	CCID-2	Path-2	SL-2	EL-2	CLOC-2
36	1,468	613	JHotDraw...	317	344	28	1,469	613	JHotDraw7...	249	276	28

Click on a row to select a clone pair

JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/geom/Insets2D.java

```
        this.left = left;
        this.bottom = bottom;
        this.right = right;
    }

    @Override
    public double getTop() {
        return top;
    }

    @Override
    public double getLeft(
        return left;
    }

    @Override
```

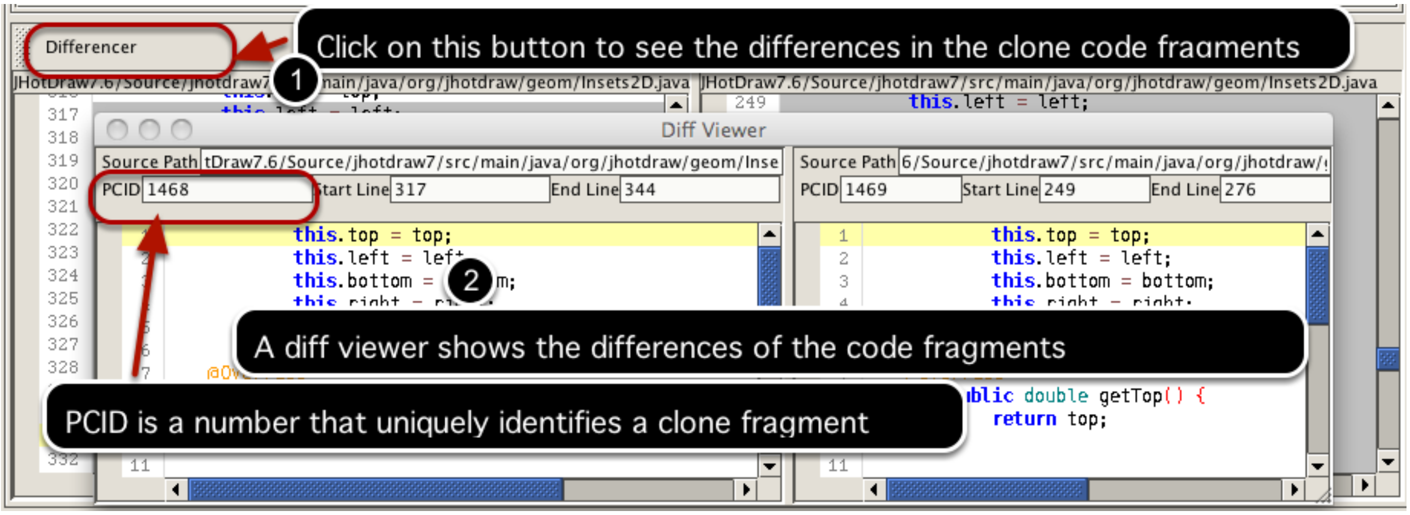
JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/geom/Insets2D.java

```
        this.left = left;
        this.bottom = bottom;
        this.right = right;
    }

    @Override
    public double getTop() {
        return top;
```

Source code of the selected clone pair

For the selected clone fragments, the next tab in the code browser shows the clone pairs. Selecting a clone pair also displays the code fragments in the bottom panel.

Source code difference analysis



Click on this button to see the differences in the clone code fragments

Source Path	tDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/geom/Inse		
PCID	1468	Start Line 317	End Line 344

Source Path	6/Source/jhotdraw7/src/main/java/org/jhotdraw/		
PCID	1469	Start Line 249	End Line 276

```
        this.top = top;
        this.left = left;
        this.bottom = bottom;
```

A diff viewer shows the differences of the code fragments

PCID is a number that uniquely identifies a clone fragment

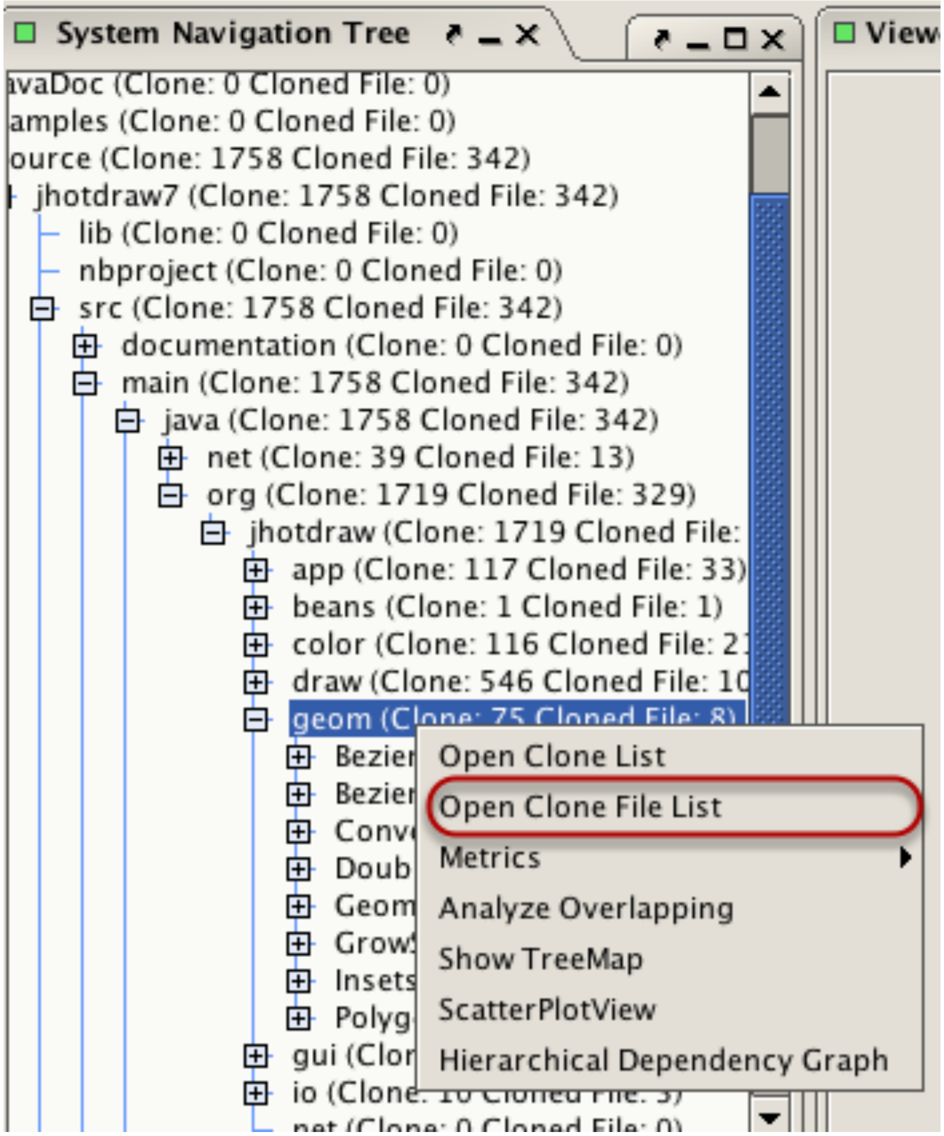
We can understand the source code differences using the diff viewer. In the above figure, the two clone fragments are exact copy of each other.

Similar to PCID, another number(CCID) is used to uniquely identify each clone class.

Analyze Clone Files

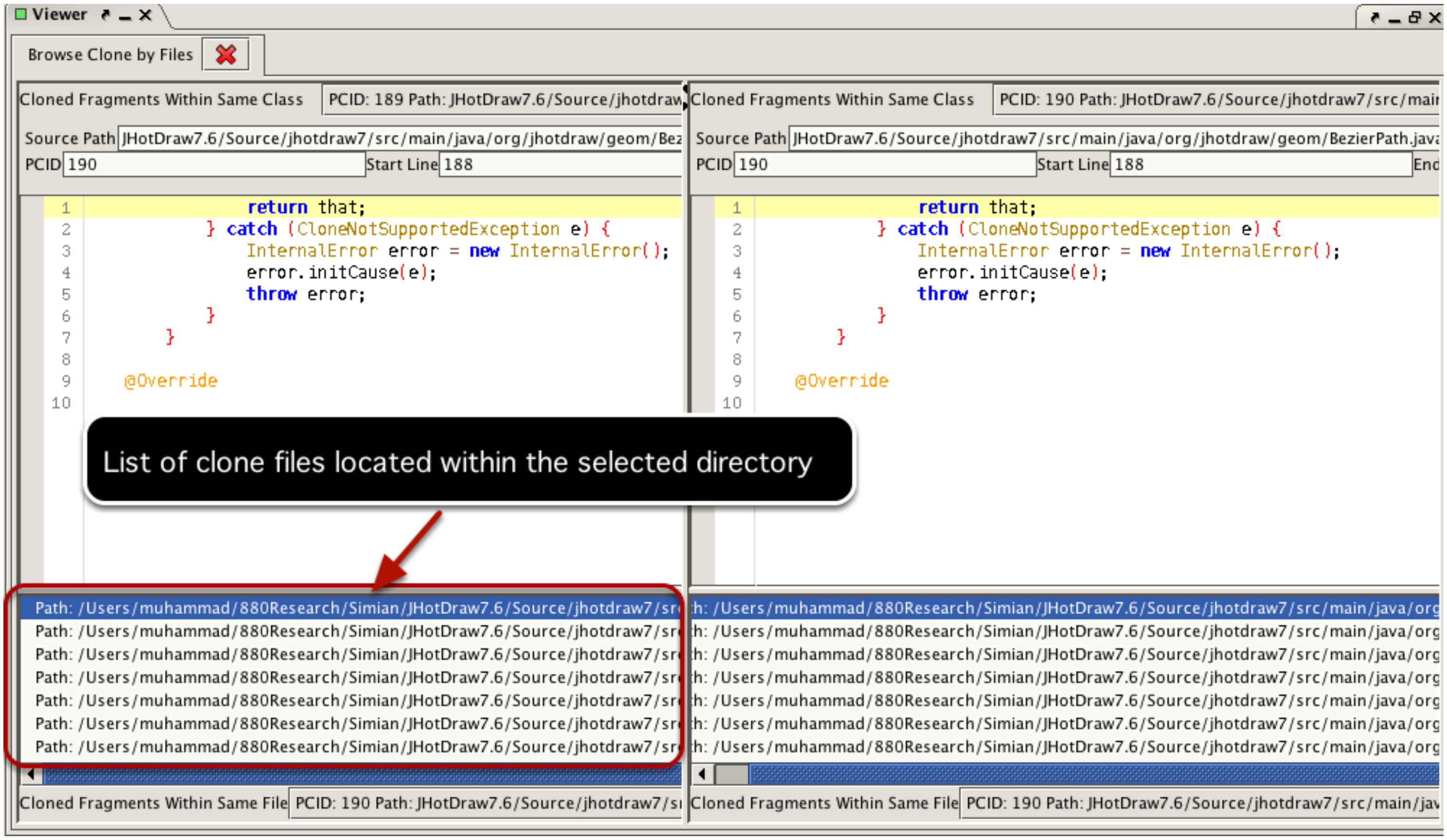
You can use this view to analyze and compare clone files with grouping and selection features.

Make selection



Select a target directory. Right click on the selected directory to open the popup menu and select *Open Clone File List* menu item.

List of clone files



Cloned Fragments Within Same Class | PCID: 189 Path: JHotDraw7.6/Source/jhotdraw

Source Path JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/geom/Bez

PCID 190 Start Line 188

```
                return that;
            } catch (CloneNotSupportedException e) {
                InternalError error = new InternalError();
                error.initCause(e);
                throw error;
            }
        }

    @Override
```

Cloned Fragments Within Same Class | PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/src/mair

Source Path JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/geom/BezierPath.java

PCID 190 Start Line 188 Enc

```
                return that;
            } catch (CloneNotSupportedException e) {
                InternalError error = new InternalError();
                error.initCause(e);
                throw error;
            }
        }

    @Override
```

List of clone files located within the selected directory

Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/sr

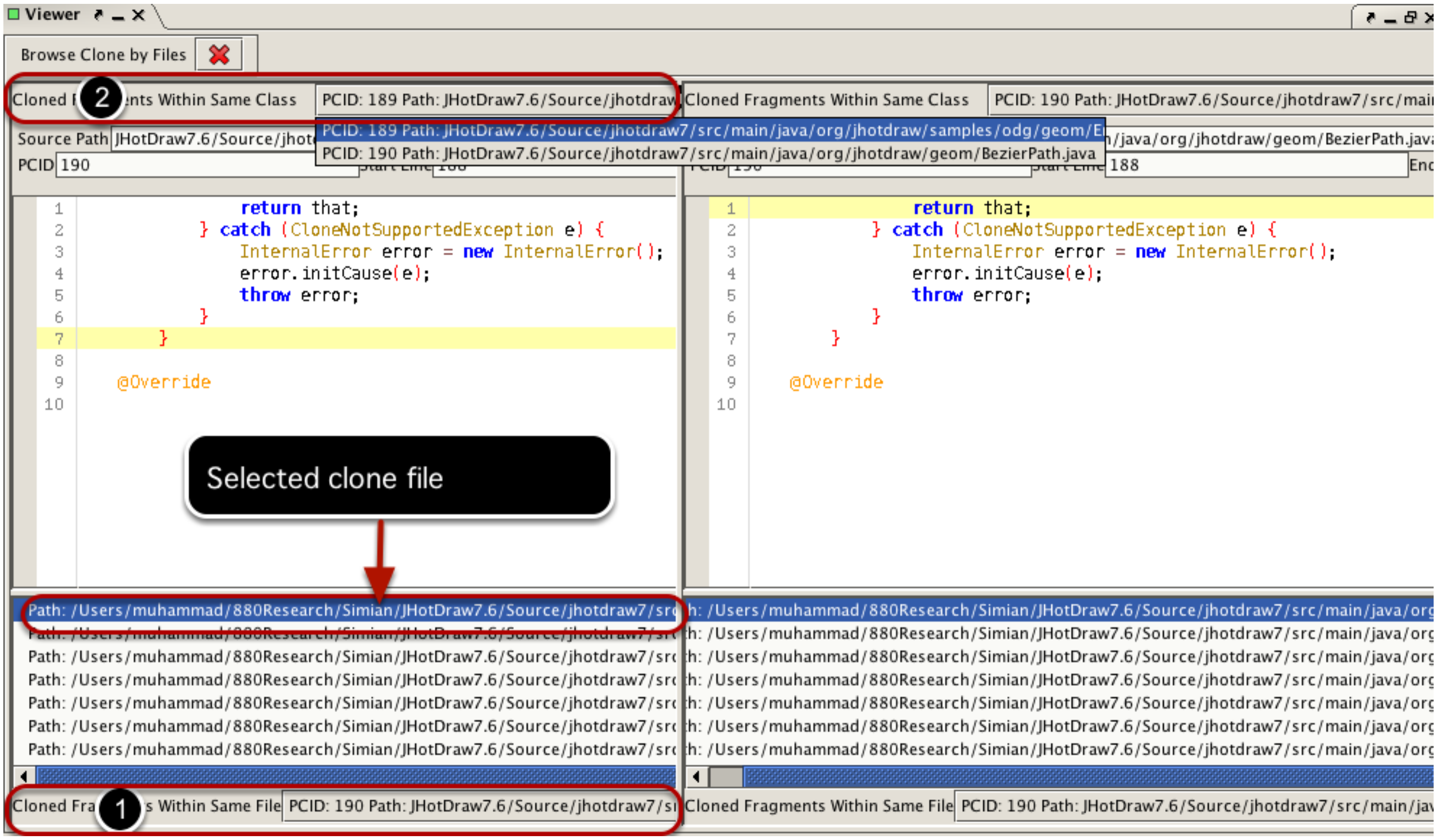
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org

Cloned Fragments Within Same File | PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/sı

Cloned Fragments Within Same File | PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/src/main/jav

We can select a clone file from the list of files.

Features



PCID: 189 Path: JHotDraw7.6/Source/jhotdraw

PCID: 189 Path: JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/samples/odg/geom/E

PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/geom/BezierPath.java

PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/src/mai

Source Path JHotDraw7.6/Source/jhot

PCID 190

n/java/org/jhotdraw/geom/BezierPath.java

188

```
            return that;
        } catch (CloneNotSupportedException e) {
            InternalError error = new InternalError();
            error.initCause(e);
            throw error;
        }
    }

    @Override
```

```
            return that;
        } catch (CloneNotSupportedException e) {
            InternalError error = new InternalError();
            error.initCause(e);
            throw error;
        }
    }

    @Override
```

```
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
Path: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src
```

```
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
h: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org
```

PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/si

PCID: 190 Path: JHotDraw7.6/Source/jhotdraw7/src/main/jav

This view has some advantages. It groups clone fragments located within the selected file(labelled with 1). For the selected clone fragment, it also groups all clone fragments that falls within the same clone class(labelled with 2).

The left and right part shows the same list of clone files. We can change the selection and compare the related clone fragments side by side.

Visualization

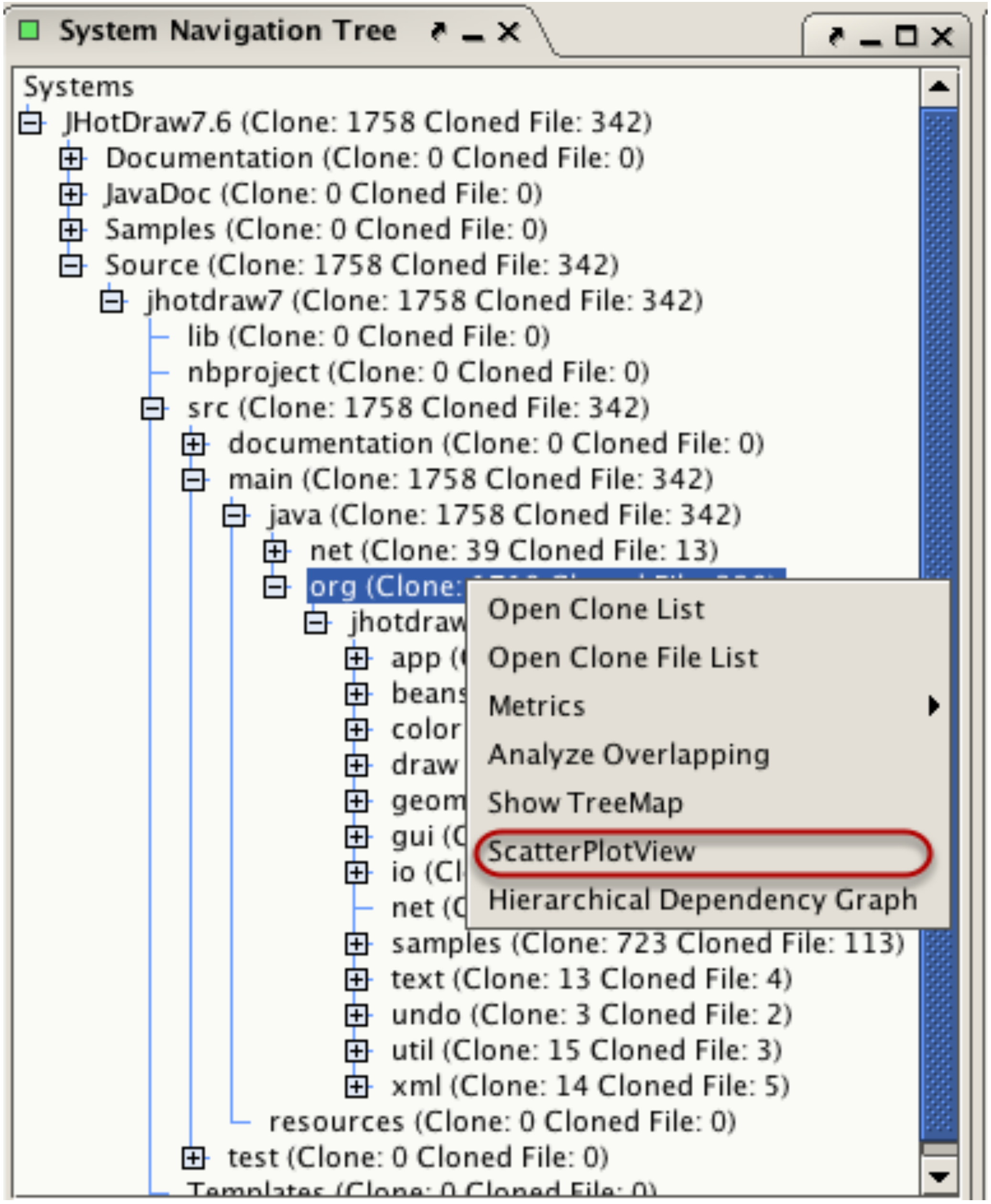
Introduction

Visualization plays an important role in code clone analysis since it can provide high level overview of cloning in a system. At present, VisCad supports three different visualizations which are scatter plot, treemap and hierarchical dependency graph.

Scatter Plot

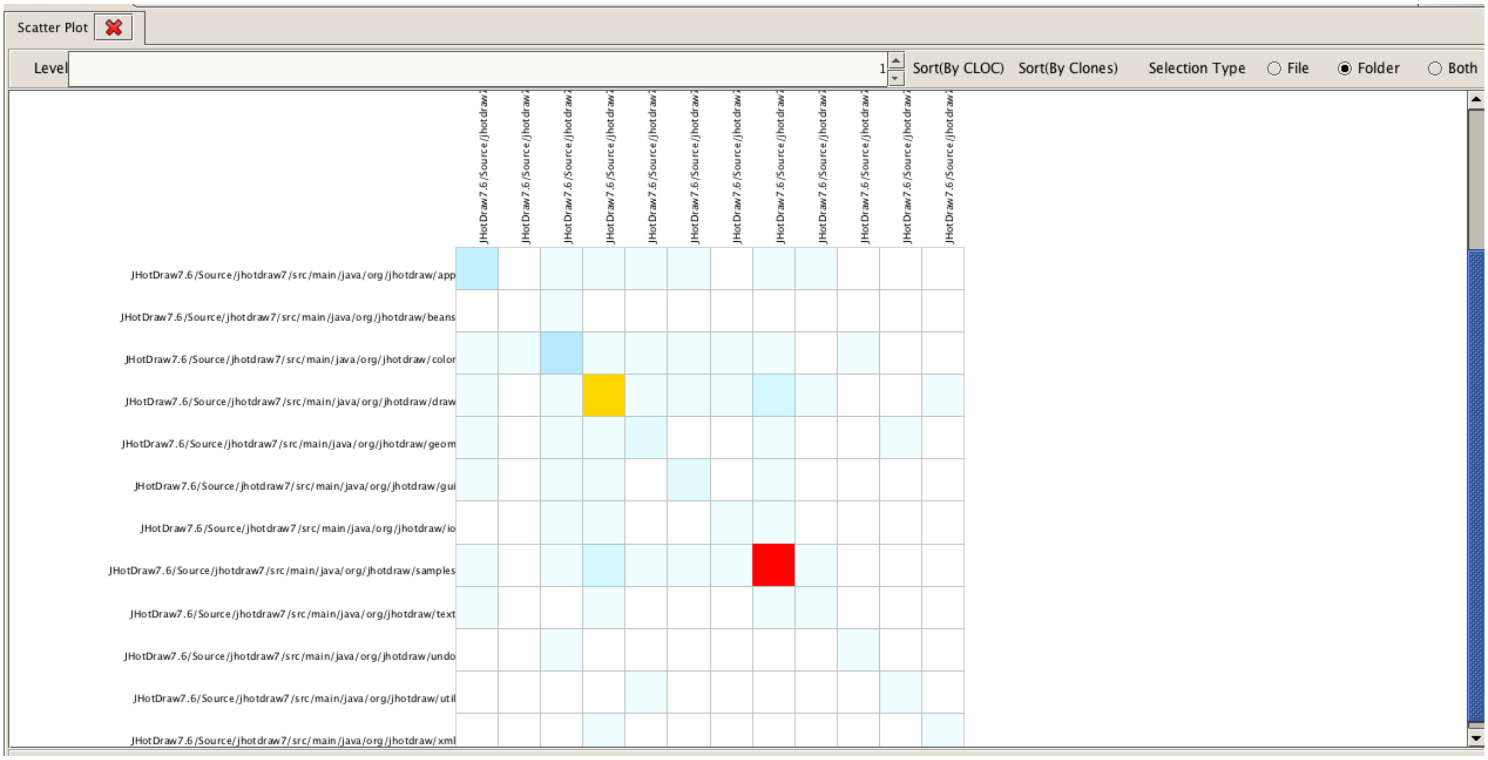
A scatter plot can be viewed as a two dimensional matrix where each cell represents the cloning status between a pair of files or directories. In VisCad, cells render the clone pairs distributed between a pair of files or directories using a color heatmap. Cells are also labelled in the horizontal and vertical axes.

Opening a scatter plot



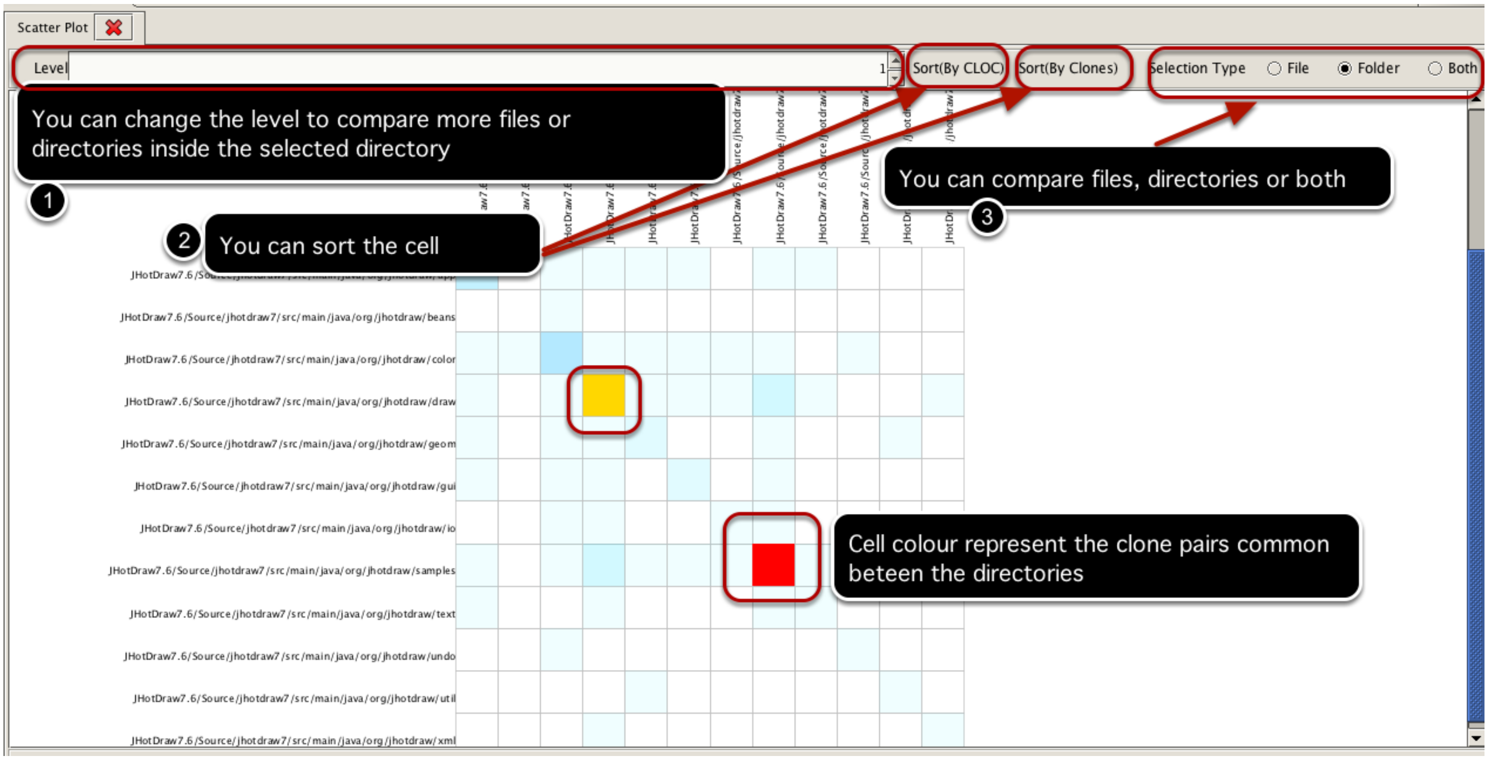
Select a directory from the system navigation tree and right click on it to open a popup menu. Click on the *Scatter Plot View* menu item.

An example



The previous selection opens the above scatter plot on a new tab in the *viewer* view panel (in the middle of the VisCad user interface).

Options



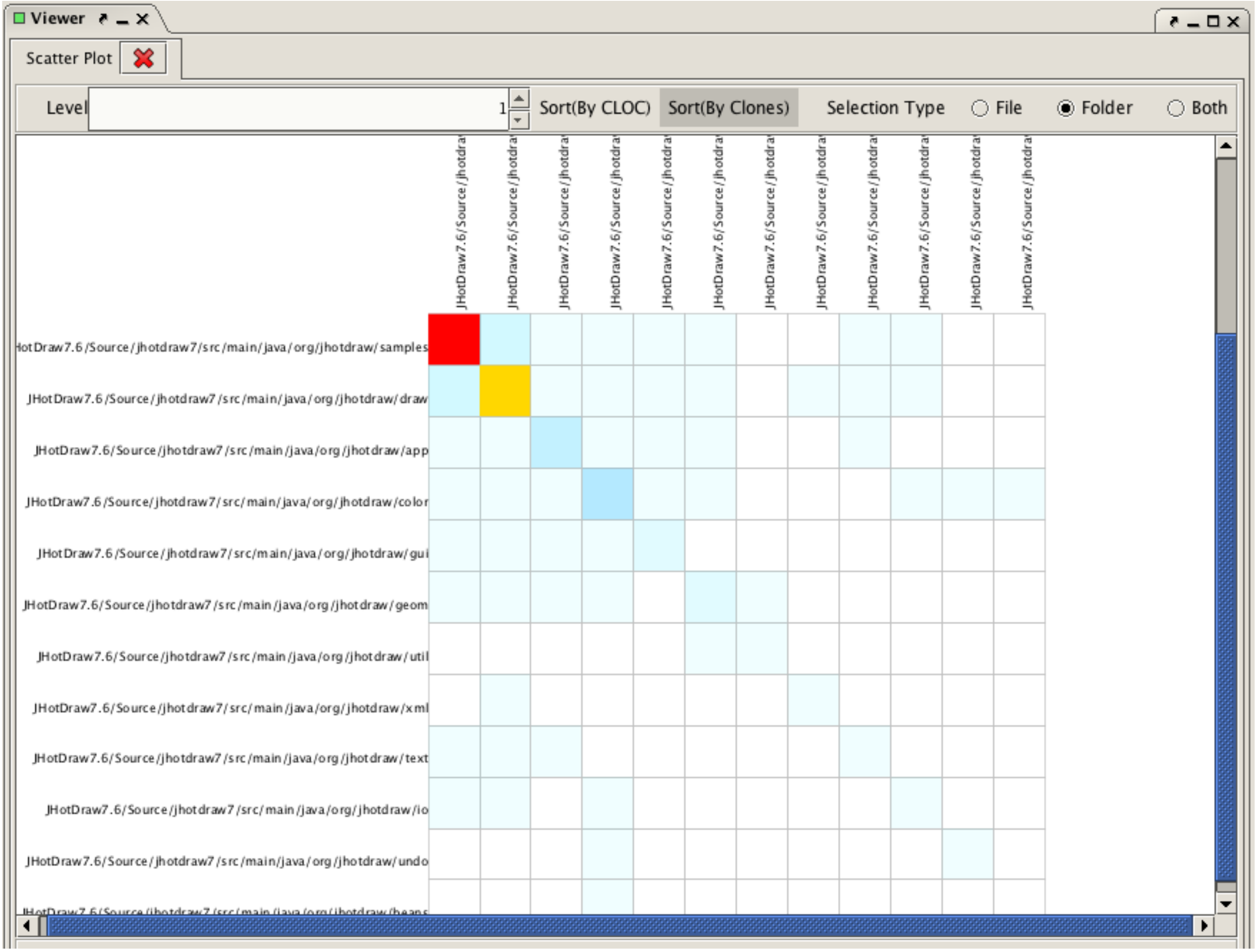
① You can change the level to compare more files or directories inside the selected directory

② You can sort the cell

③ You can compare files, directories or both

Cell colour represent the clone pairs common beteen the directories

Example of Sorting

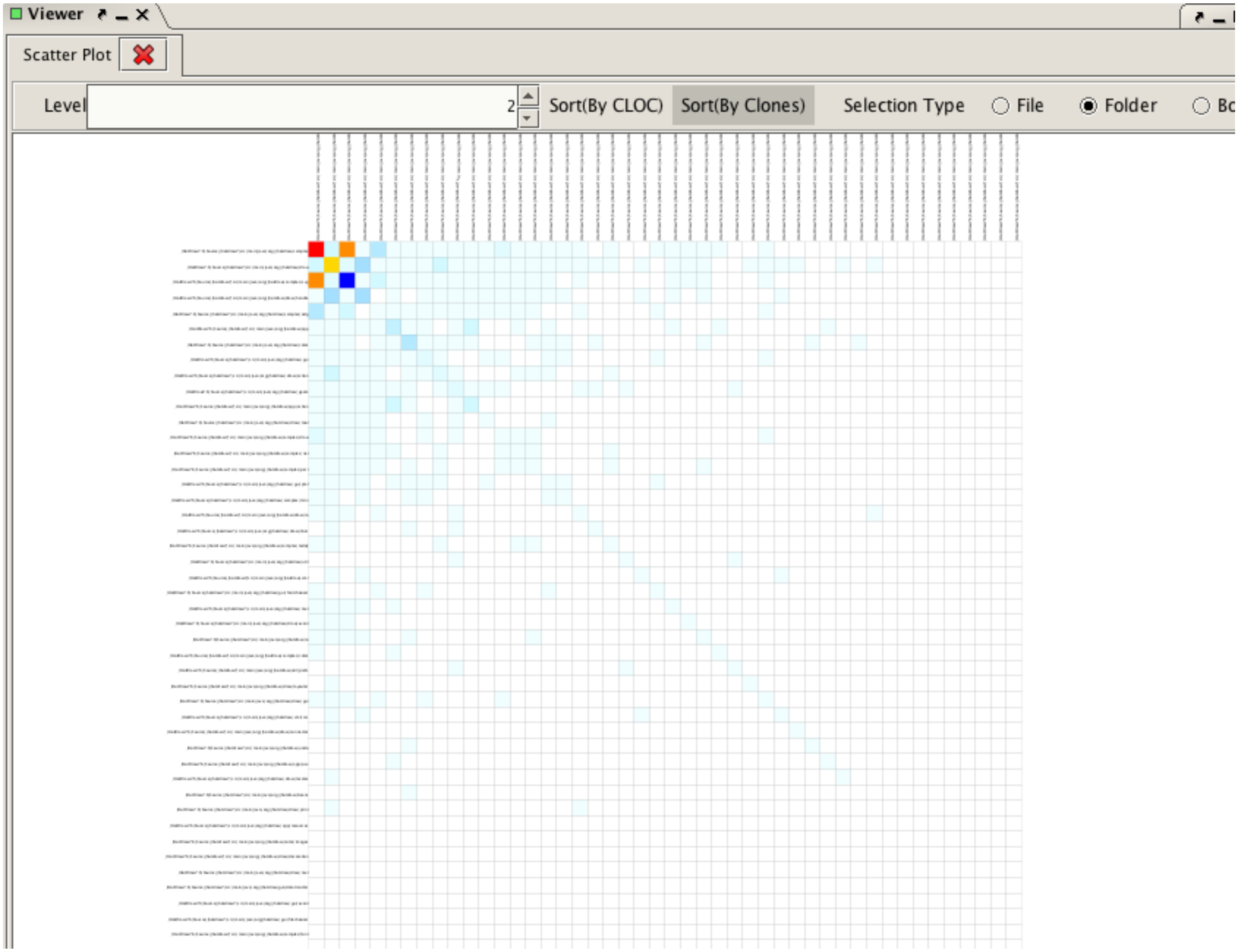


Level 1

Selection Type ○ File ● Folder ○ Both

Sorting allows to identify cloning patterns easily.

Zoom in or zoom out

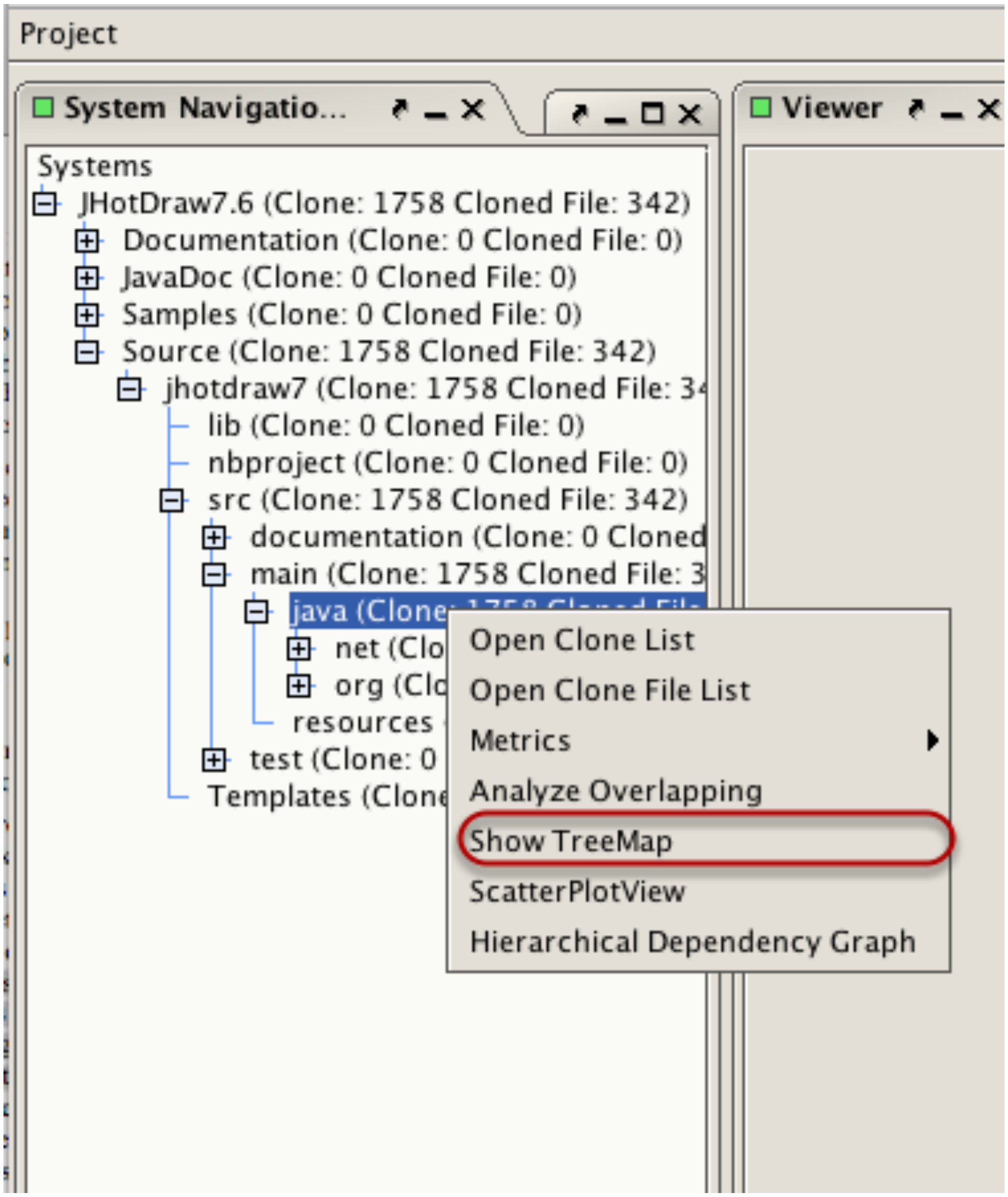


You can hold the right mouse button and move it in the inward or outward direction to perform zoom in or zoom out operations. The above figure shows an example of zoom out operation. You can also identify the files or directories involved with a cell with tooltip by holding the mouse pointer on the cell.

Treemap

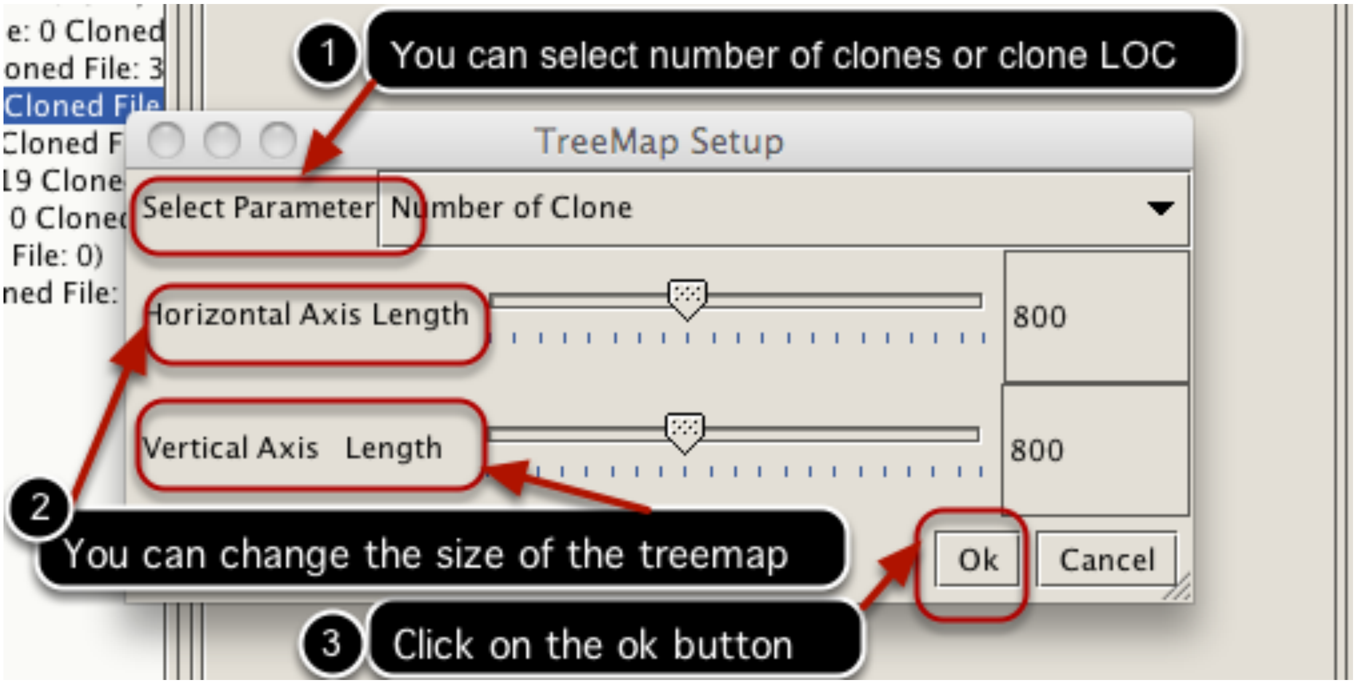
Treemap preserves the hierarchical structure of subject systems where each rectangle represents a file or directory. The rectangles representing the files are aggregated to indicate the cloning status of a directory in the system hierarchy.

Opening treemap



Select a directory and right clink on it to open the popup menu. Click on the *Show TreeMap* menu item.

Configure the treemap



① You can select number of clones or clone LOC

TreeMap Setup

Select Parameter: Number of Clone

Horizontal Axis Length: 800

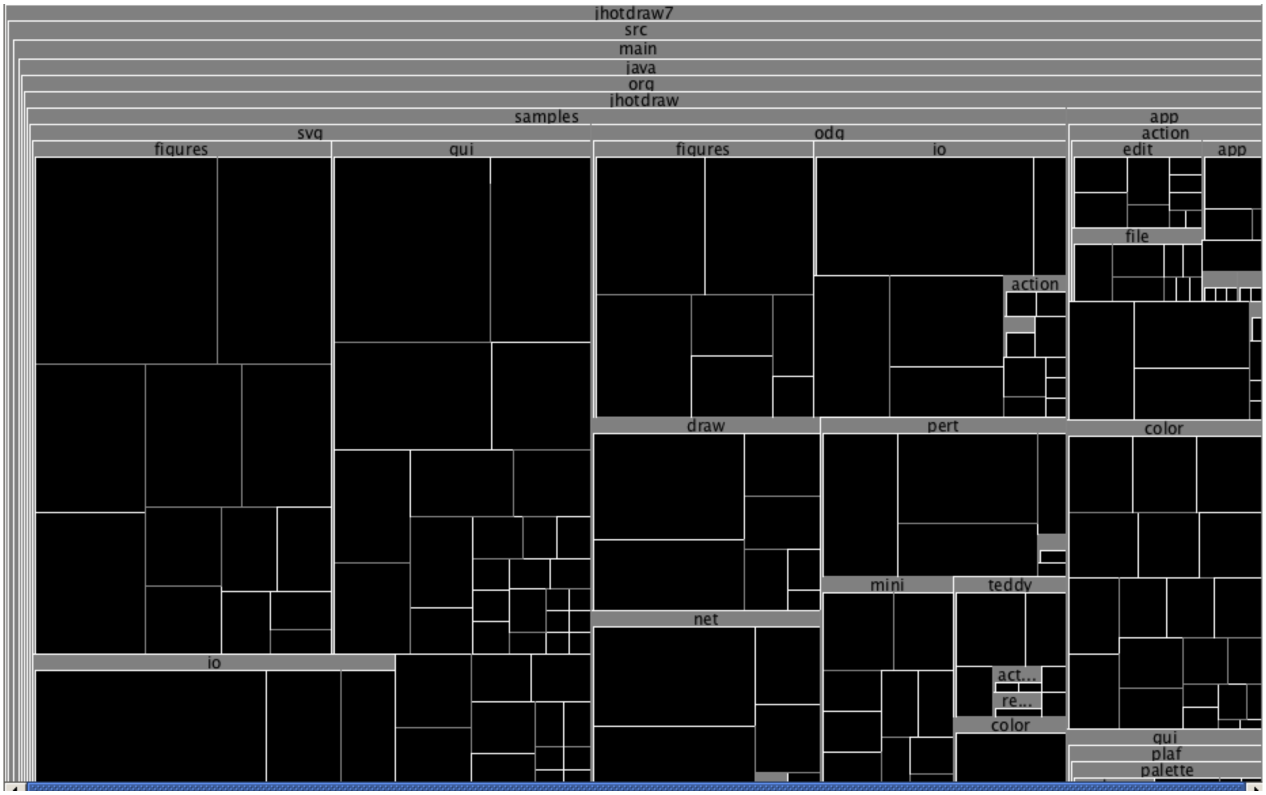
Vertical Axis Length: 800

② You can change the size of the treemap

③ Click on the ok button

A dialog box appears that can be used to configure thre treemap. Click on the ok button to open the treemap.

An Example

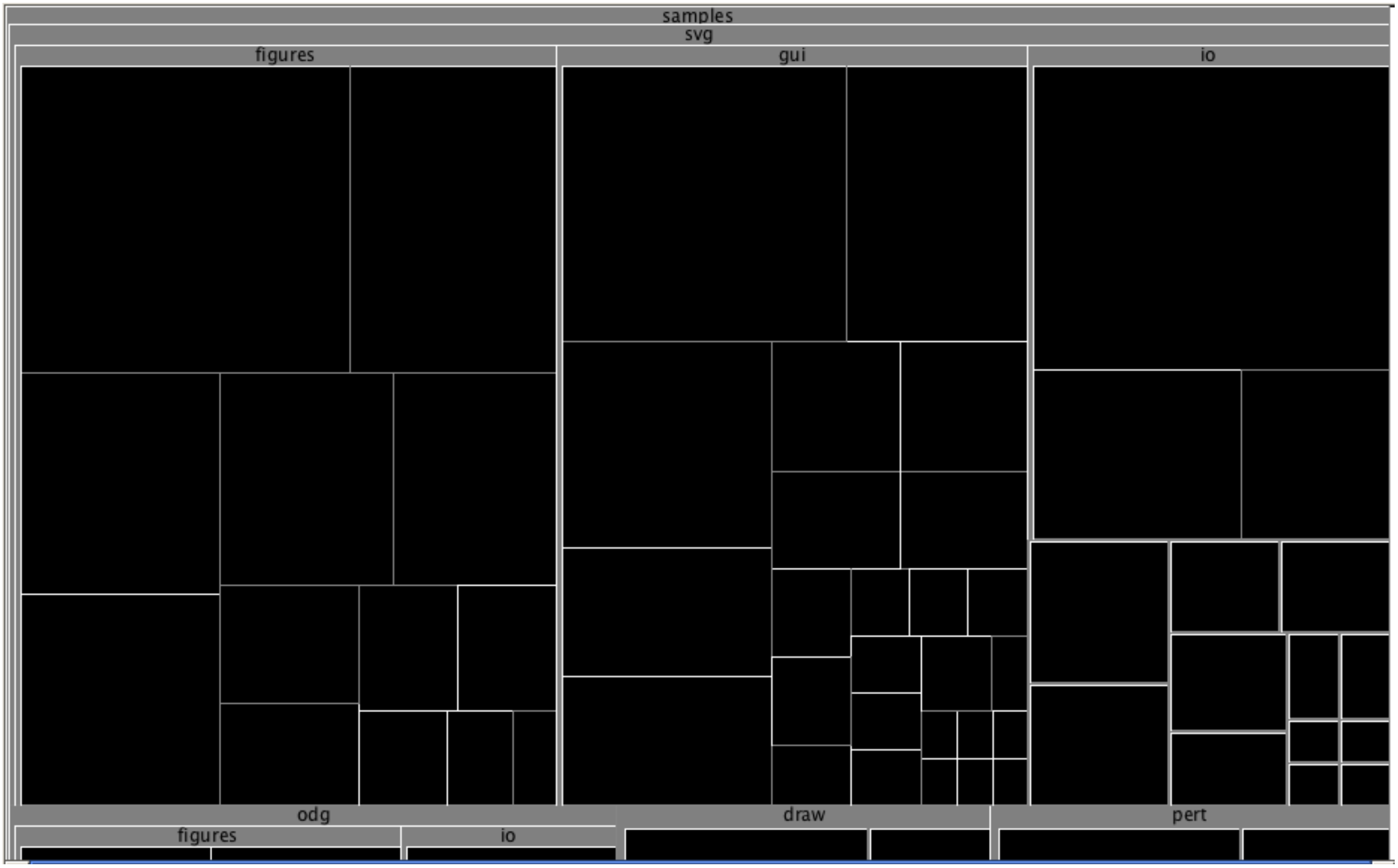


An example of a treemap for the JHotDraw system(Version 7.6). Putting the mouse over any rectangle will provide details information about that using a tooltip.

Zoom in or zoom out

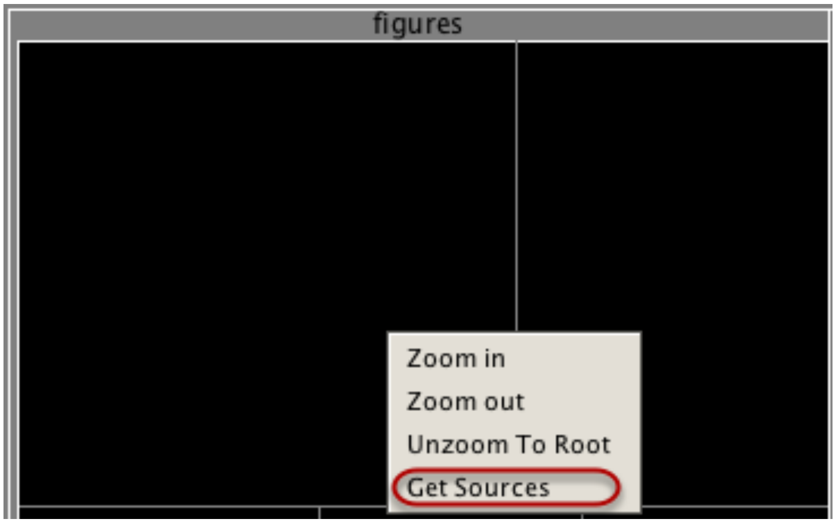
Right click on rectangle representing a directory opens a popup menu. You can then perform zoom in or zoom out operation.

An example of zoom in operation



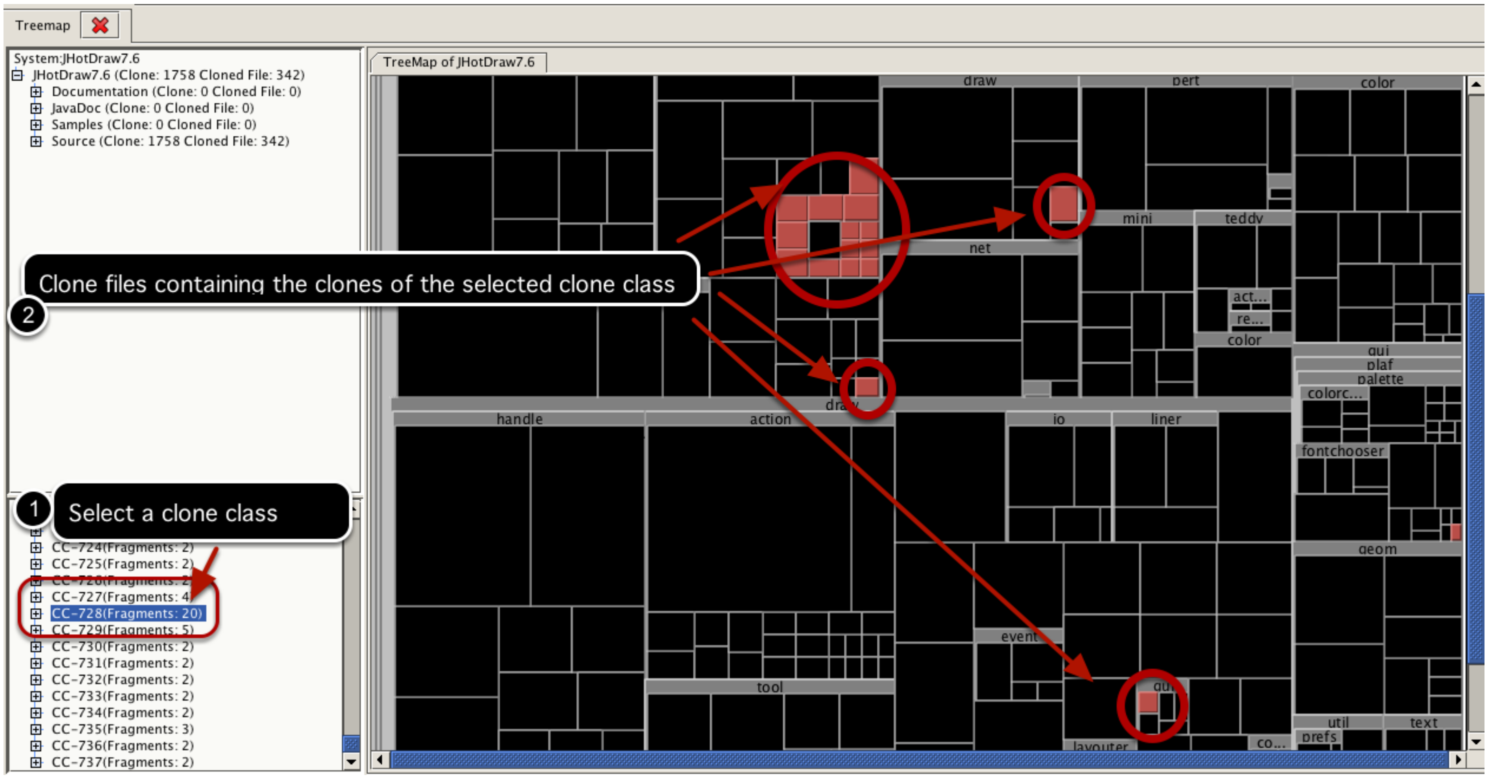
An example of zoom in operation on the *samples* directory. We can select the zoom out operation to move one step back or can select the *unzoom to root* option to move back to the beginning state.

Obtaining source code of the clone fragment



Right click on a rectangle and select the *Get Source* menu item. This will open the clone code fragments of the file(s) in a new tab using the clone code browser. For details about the code browser, see Analyze Clone Fragments.

Obtaining clone distribution information



2 Clone files containing the clones of the selected clone class

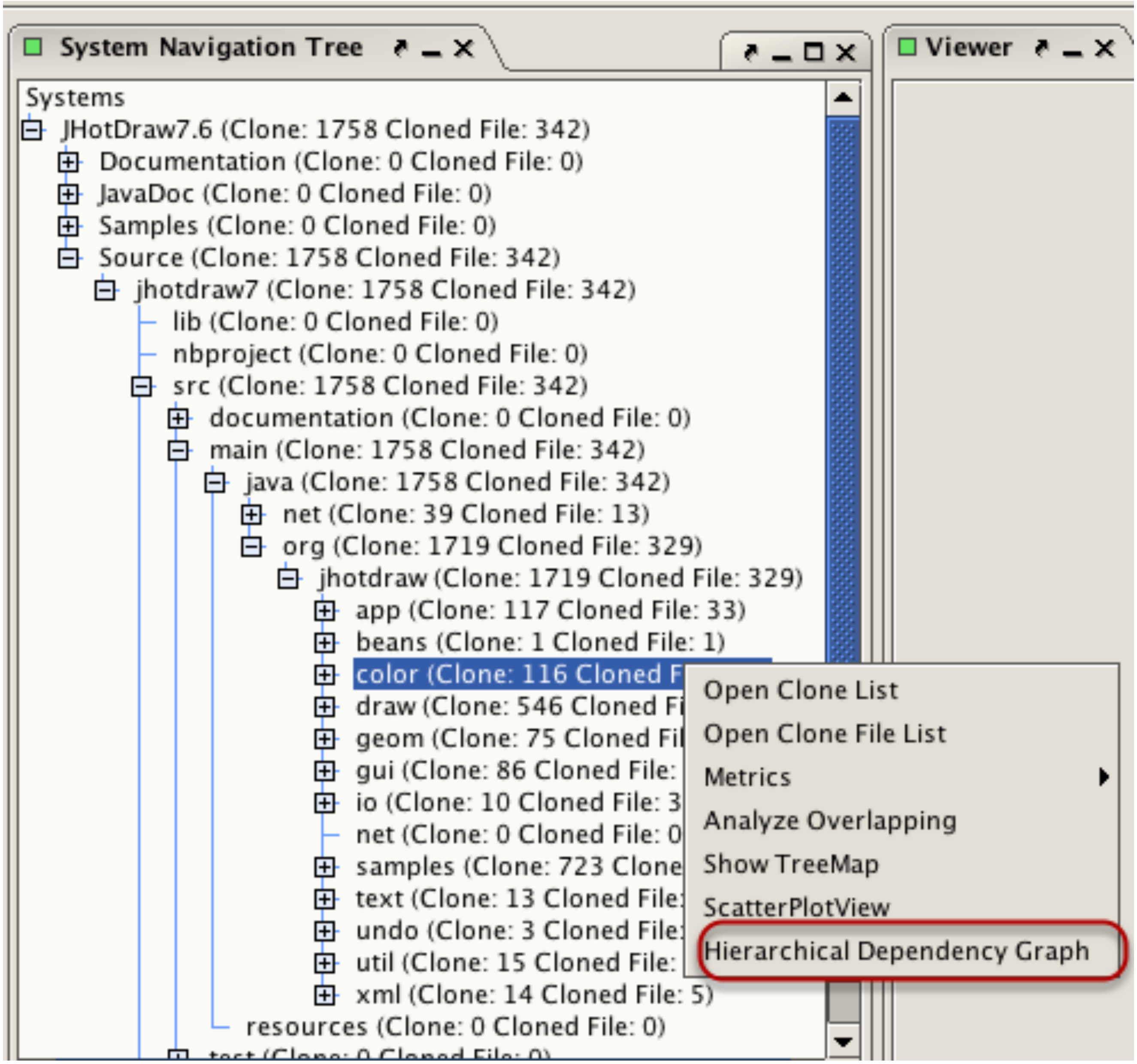
1 Select a clone class

You can understand how the clone fragments of a clone class are distributed with the help of a treemap. Select a clone class from the bottom-left panel and the files containing those clones will be marked with red color in the treemap.

Hierarchical Dependency Graph

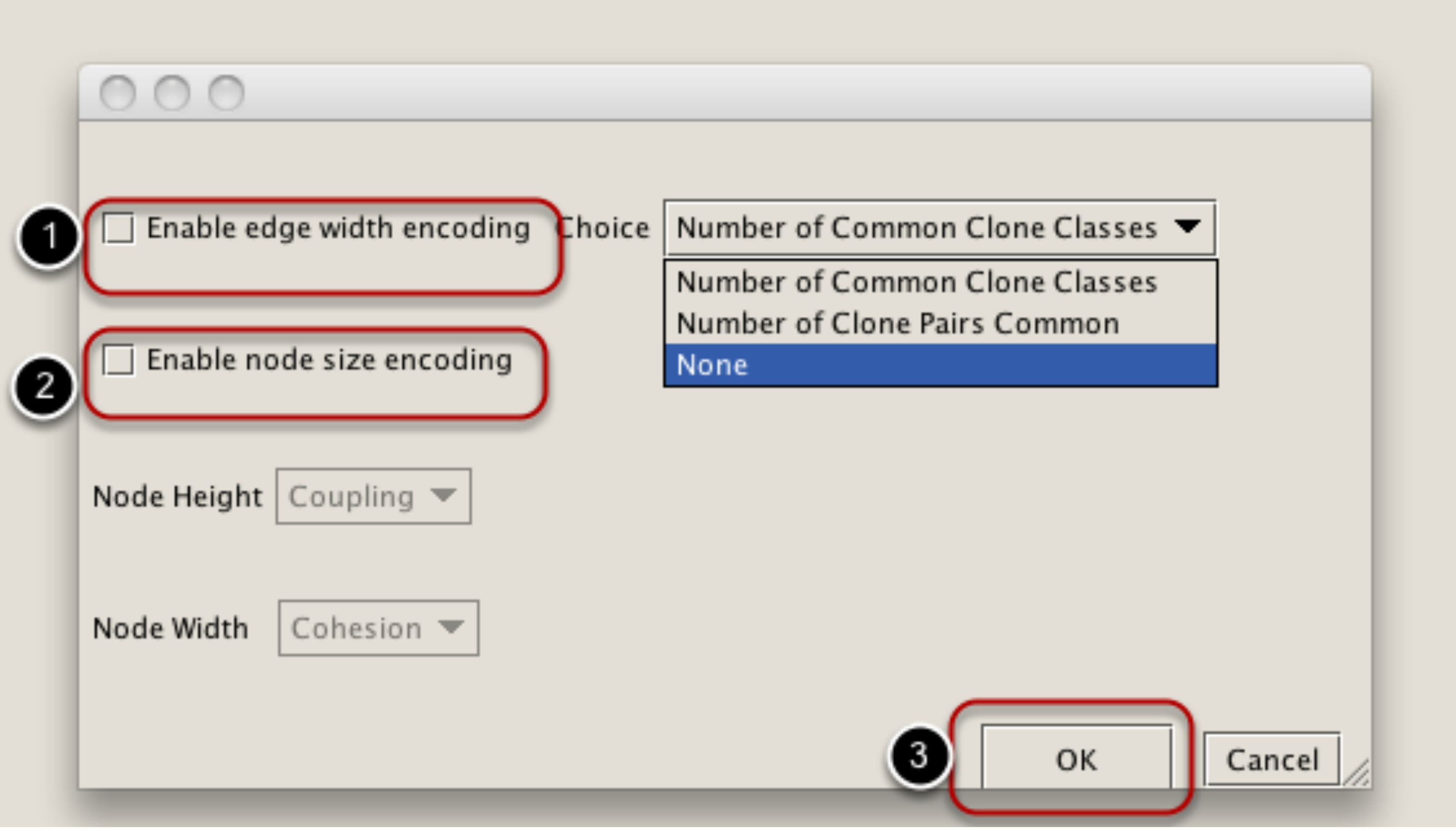
Clones are more problematic when members of a clone class scattered in different parts of a software system because this requires changes need to be made in different parts of the system. Thus, it is required to discover how clone fragments are distributed across subsystems/directories. Moreover, understanding cloning relationships among different subsystems can also reveal their dependencies. VisCad can render the hierarchical organization of a software system along with the distribution of clones using a hierarchical dependency graph.

Opening the graph



Select a directory from the *system navigation tree* and right click on it to open a popup menu. Click on the *Hierarchical Dependency Graph* menu item.

Configure the graph



☐ Enable edge width encoding

Choice: Number of Common Clone Classes

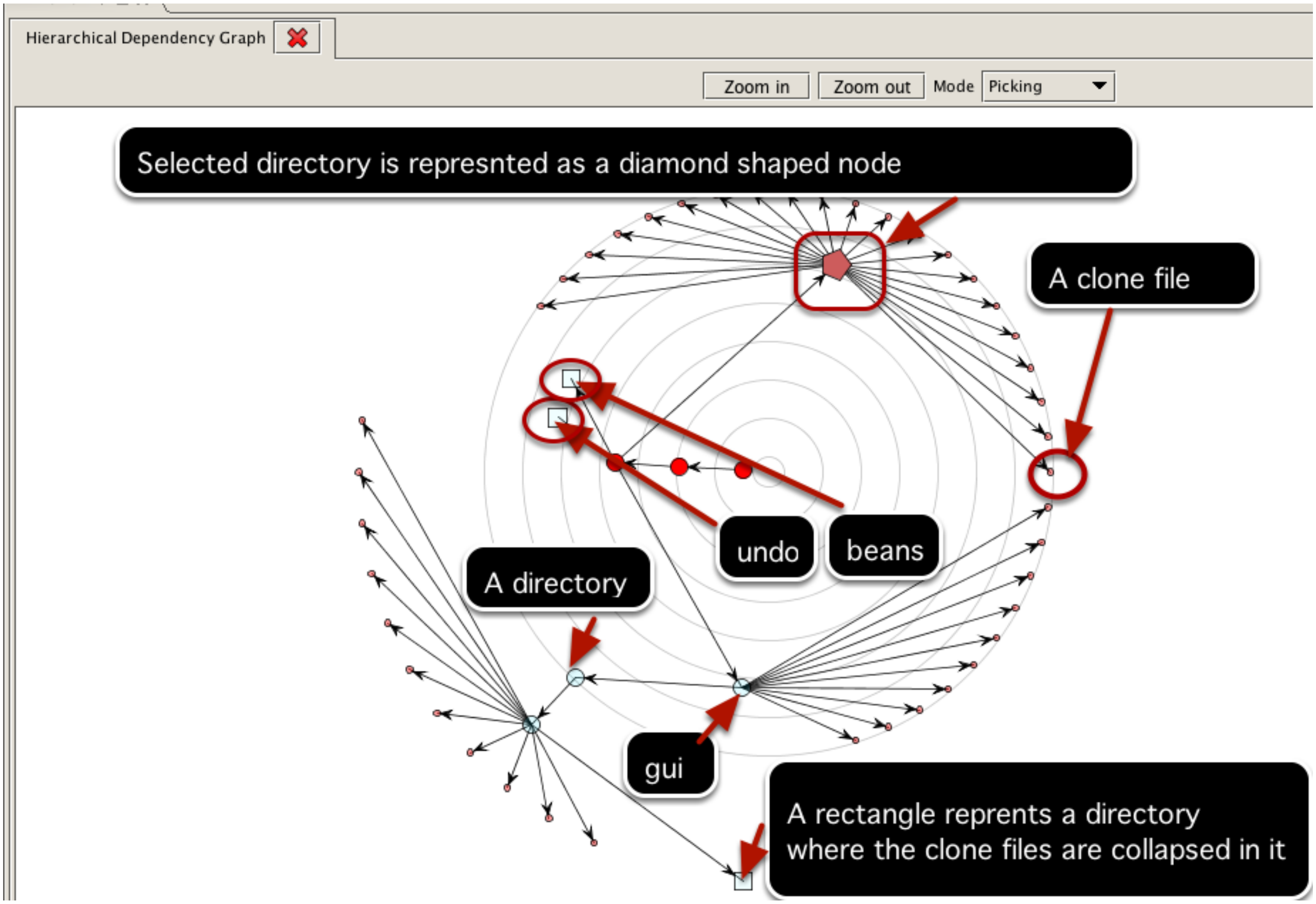
☐ Enable node size encoding

Node Height: Coupling

Node Width: Cohesion

Edge width can represent number clone classes/clone pairs distribued between a pair of directories. Nodes width and height can represent cohesion and coupling value. Click on the ok button when you are done.

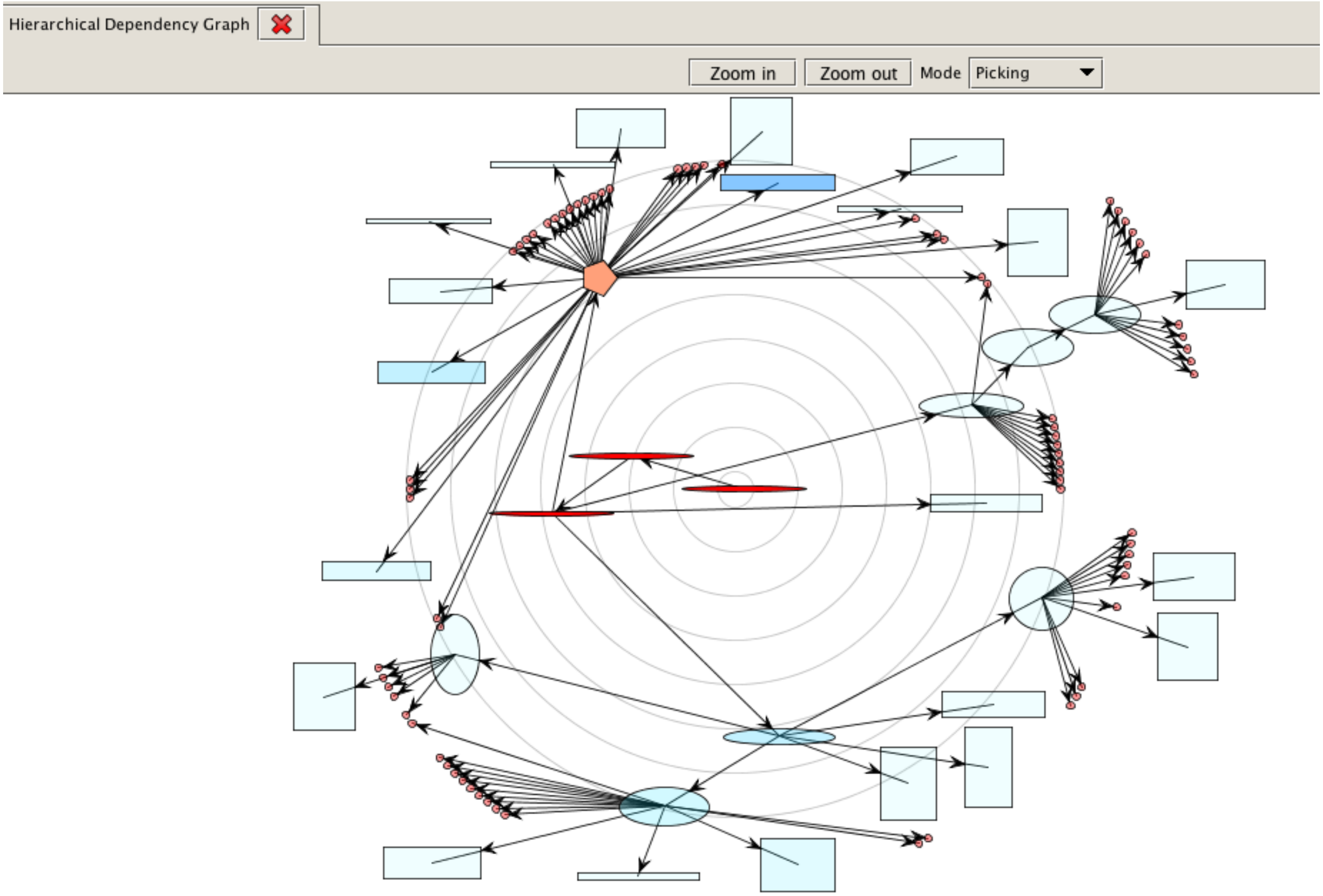
An example



Suppose we want to understand which directories in the system contain the external clones of the color directory. We select the *color* directory from the system navigation tree, open the popup menu and select the *Hierarchical Dependency Graph.*
With the default settings, we get a graph as shown in the above figure. From the figure we can understand that the color directory has external cloning relationship with the *gui*, *undo* and *beans* directories.

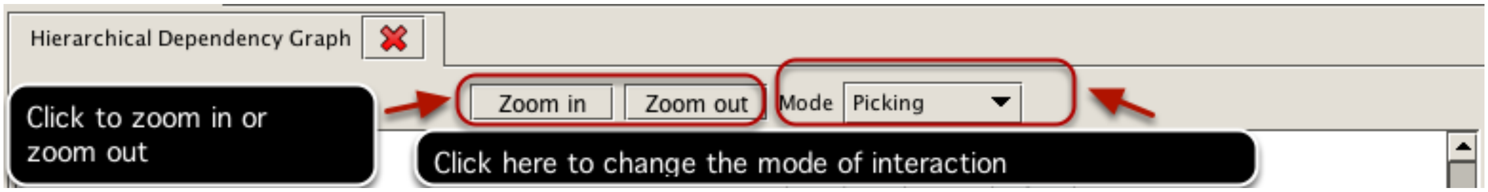
VisCad visualizes the graph by filtering those nodes that do not have any cloning relation with the selected directory.

Another example



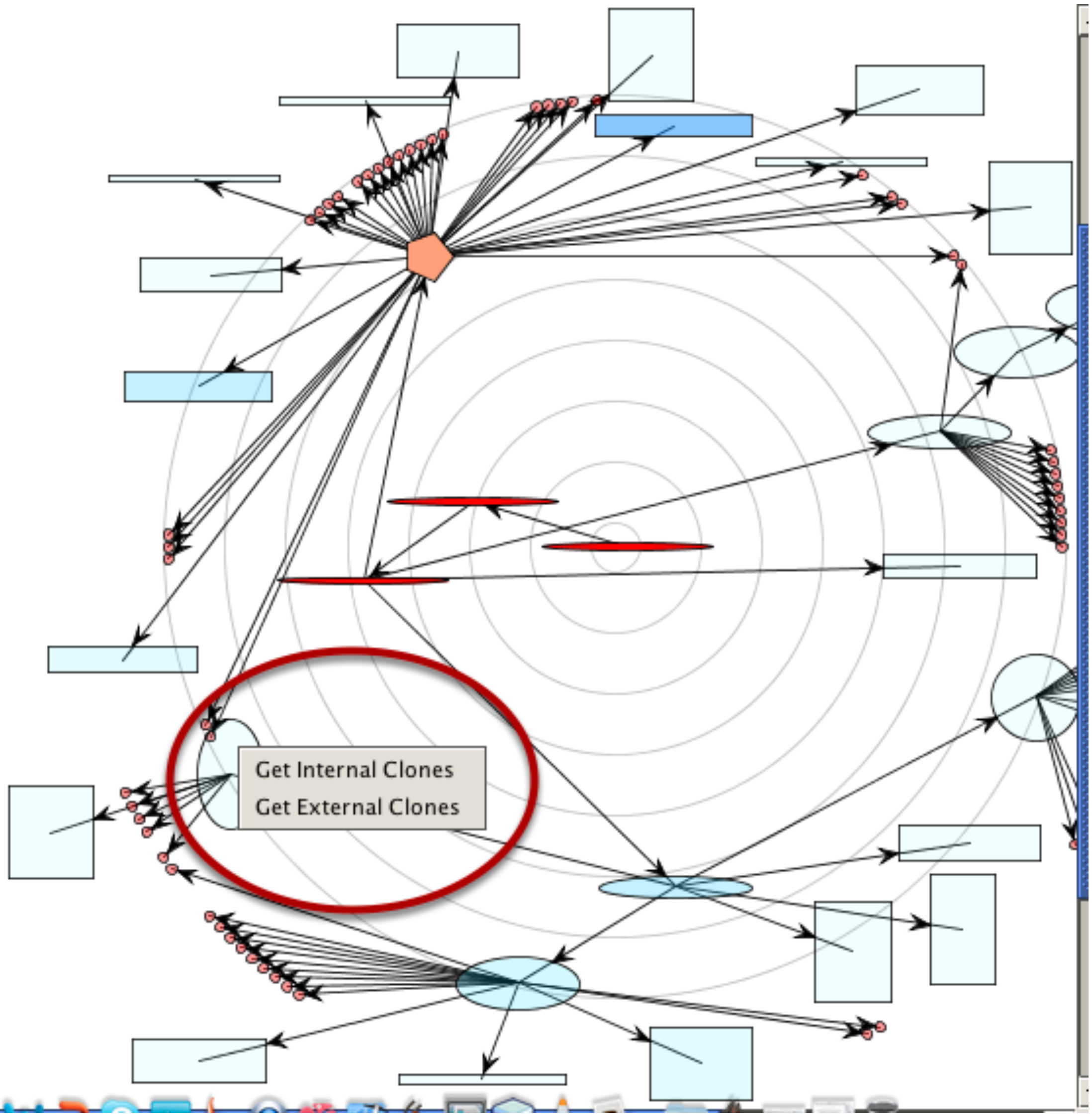
The above figure shows the hierarchical dependency graph for the *jhotdraw/draw* directory. In this case, node's (represnting the directory) width and height represent the clone cohesion and coupling value. Node color value represent the external cloning relation level with the target directory.

Two mode of interaction



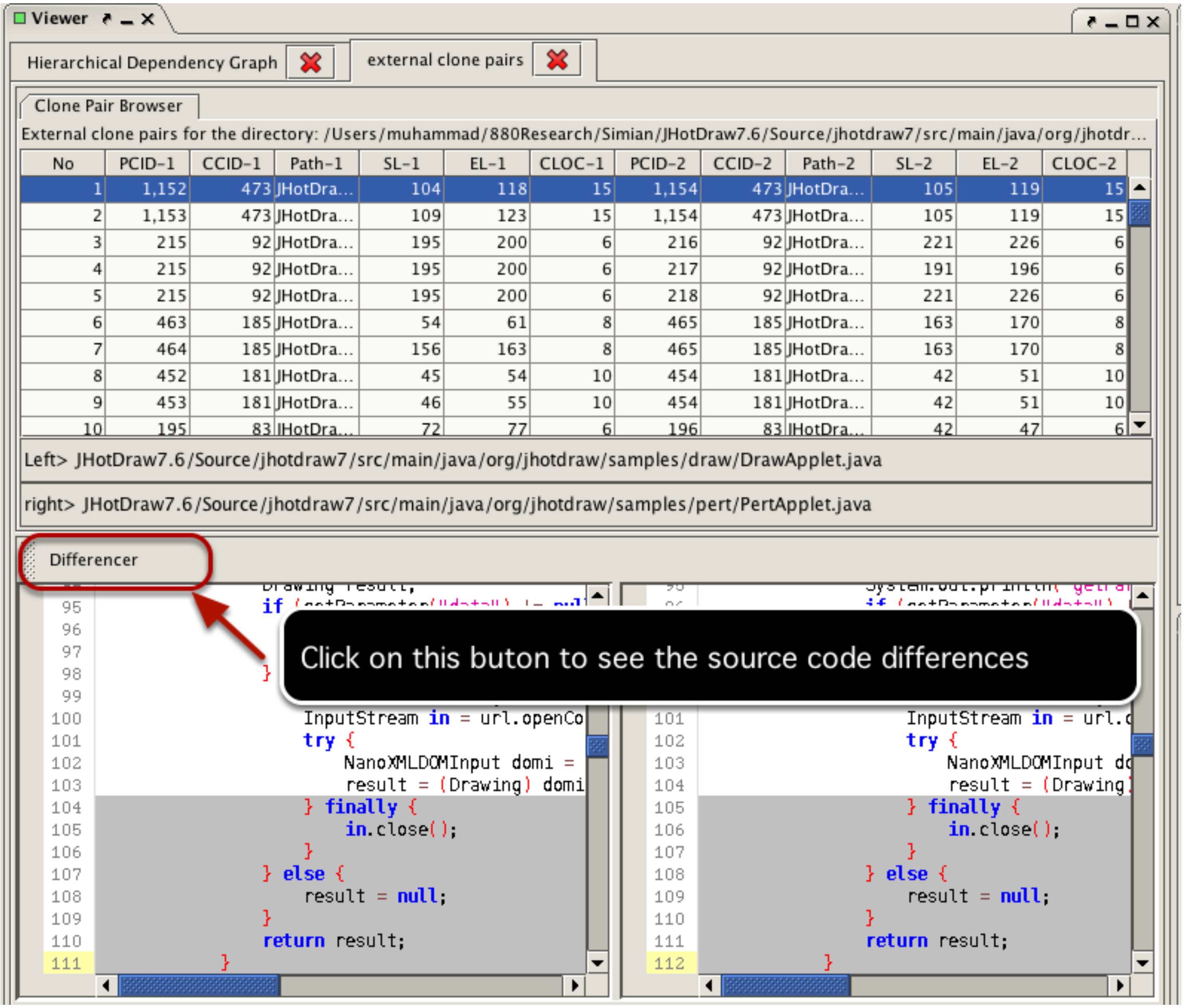
In the *picking* mode, you can select individual node and move them. In the *transforming* mode, you can move the entire graph.

Obtaining external or internal clone pairs



Right click on a node(represnting a directory) opens a popup menu. You can then select the target operation. The clone pairs(internal/external) will be viewed in a new tab using the *clone pair browser* component (see the next figure).

Example of a clone pair browser



External clone pairs for the directory: /Users/muhammad/880Research/Simian/JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdr...

No	PCID-1	CCID-1	Path-1	SL-1	EL-1	CLOC-1	PCID-2	CCID-2	Path-2	SL-2	EL-2	CLOC-2
1	1,152	473	JHotDra...	104	118	15	1,154	473	JHotDra...	105	119	15
2	1,153	473	JHotDra...	109	123	15	1,154	473	JHotDra...	105	119	15
3	215	92	JHotDra...	195	200	6	216	92	JHotDra...	221	226	6
4	215	92	JHotDra...	195	200	6	217	92	JHotDra...	191	196	6
5	215	92	JHotDra...	195	200	6	218	92	JHotDra...	221	226	6
6	463	185	JHotDra...	54	61	8	465	185	JHotDra...	163	170	8
7	464	185	JHotDra...	156	163	8	465	185	JHotDra...	163	170	8
8	452	181	JHotDra...	45	54	10	454	181	JHotDra...	42	51	10
9	453	181	JHotDra...	46	55	10	454	181	JHotDra...	42	51	10
10	195	83	JHotDra...	72	77	6	196	83	JHotDra...	42	47	6

Left> JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/samples/draw/DrawApplet.java

right> JHotDraw7.6/Source/jhotdraw7/src/main/java/org/jhotdraw/samples/pert/PertApplet.java

```
            }

                InputStream in = url.openCo
                try {
                    NanoXMLDOMInput domi =
                    result = (Drawing) domi
                } finally {
                    in.close();
                }
            } else {
                result = null;
            }
            return result;
        }
```

```
                InputStream in = url.c
                try {
                    NanoXMLDOMInput dc
                    result = (Drawing)
                } finally {
                    in.close();
                }
            } else {
                result = null;
            }
            return result;
        }
```

Click on this buton to see the source code differences

Code Clone Metrics

Introduction

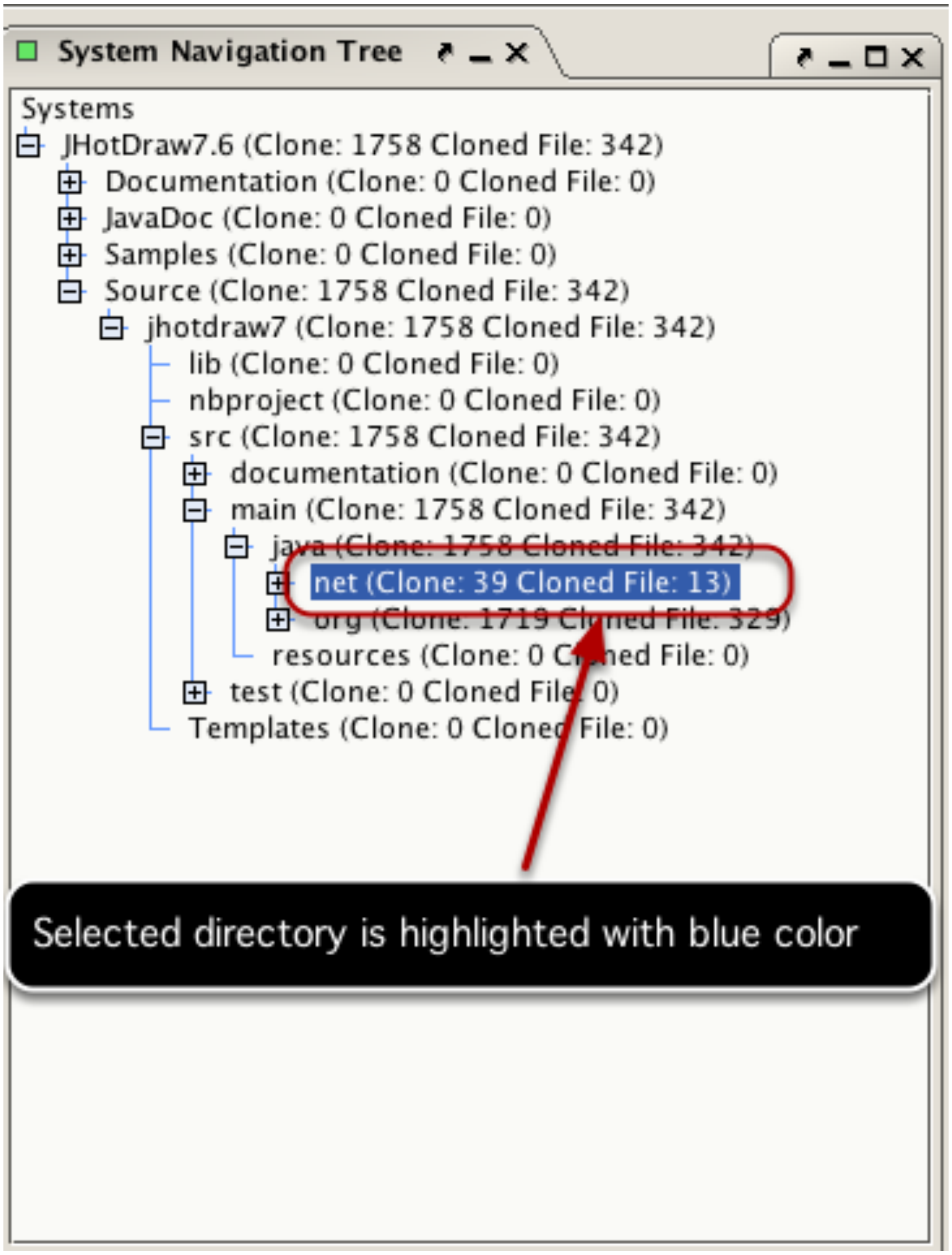
For supporting in-depth clone analysis, VisCad can compute a set of metrics. We can divide the metrics into two broard categories. The first set of metrics (*clone system metric set*) relate clones with the organizational structure of the subject system and can be computed for different system boundaries, such as for the entire system, for subsystems/directories or for source files, as per the user's choice. Depending on the granularity of operation, we can again subdivide them into two groups, the *file metric set* and the *directory metric set*. The next set of metrics (*clone class metrics set*) deals with the clone classes.

VisCad supports four operations for each metric set. These are:
1. Exporting : Results of metric computations can be exported in CSV (comma separated values)format.
2. Plotting : Although metrics are important for quantitative analysis, identifying important patterns from a large set of data is difficult. To avoid such difficulties, users can plot the metrics values with a bar chart which helps in identifying an anomaly within clone patterns easily.
3. Browsing Clone Code: Depending on the metrics values, user may be interested to explore the clone fragments located within a file or directory. VisCad also supports such operation.
4. Sorting : Values can be sorted to locate the maximum or the minimum value easily.

Obtaining Metrics For Files

This section discusses the steps for obtaining various metrics values for the clone files located in a directory.

Select a directory

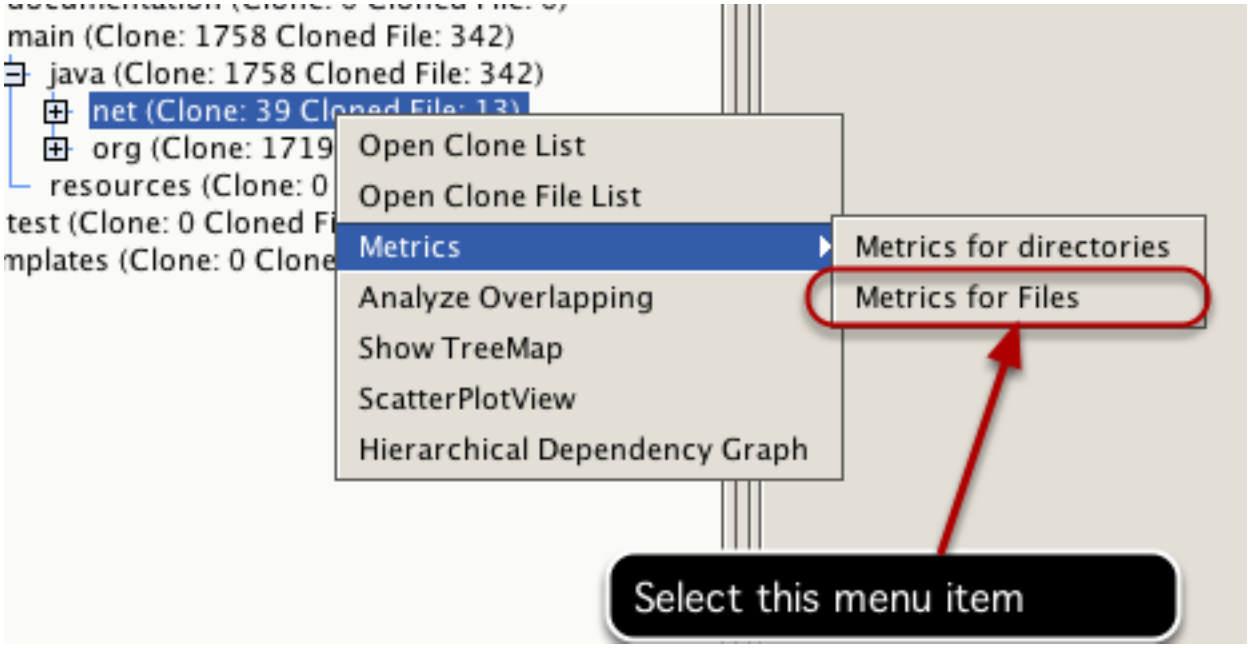


- Systems
  - JHotDraw7.6 (Clone: 1758 Cloned File: 342)
    - Documentation (Clone: 0 Cloned File: 0)
    - JavaDoc (Clone: 0 Cloned File: 0)
    - Samples (Clone: 0 Cloned File: 0)
    - Source (Clone: 1758 Cloned File: 342)
      - jhotdraw7 (Clone: 1758 Cloned File: 342)
        - lib (Clone: 0 Cloned File: 0)
        - nbproject (Clone: 0 Cloned File: 0)
        - src (Clone: 1758 Cloned File: 342)
          - documentation (Clone: 0 Cloned File: 0)
          - main (Clone: 1758 Cloned File: 342)
            - java (Clone: 1758 Cloned File: 342)
              - net (Clone: 39 Cloned File: 13)
              - org (Clone: 1719 Cloned File: 329)
            - resources (Clone: 0 Cloned File: 0)
          - test (Clone: 0 Cloned File: 0)
        - Templates (Clone: 0 Cloned File: 0)

Selected directory is highlighted with blue color

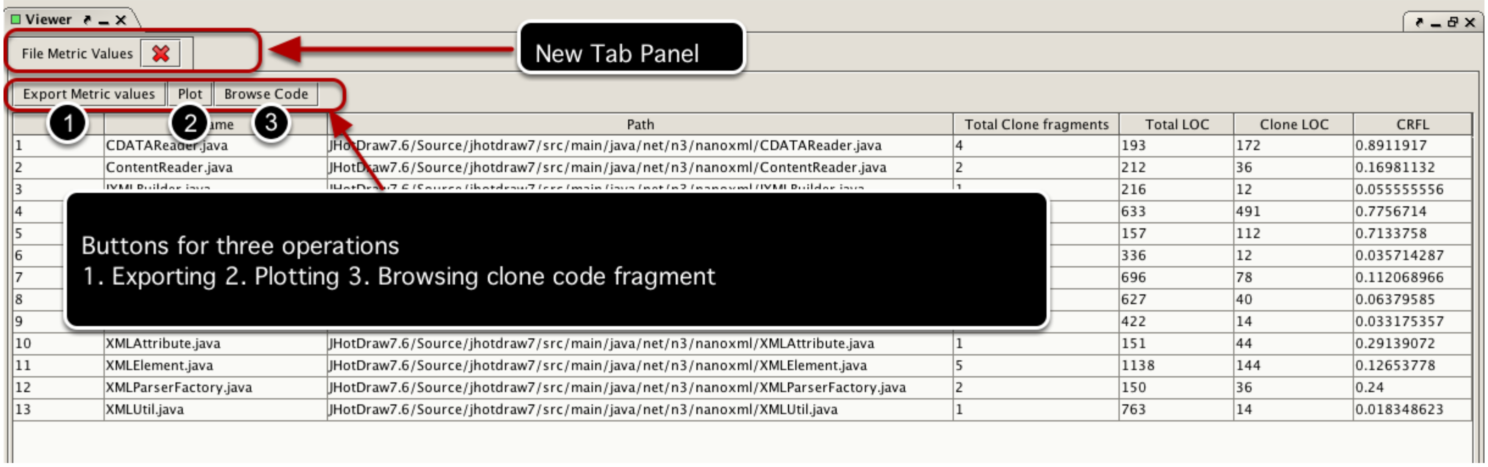
Click on a target directory to select it from the system navigation tree. Metrics will be computed for all clone files located within this directory

Open the popup menu and select the operation



Right click on the selected directory to open the popup menu. Select Metrics->Metrics for files from the menu.

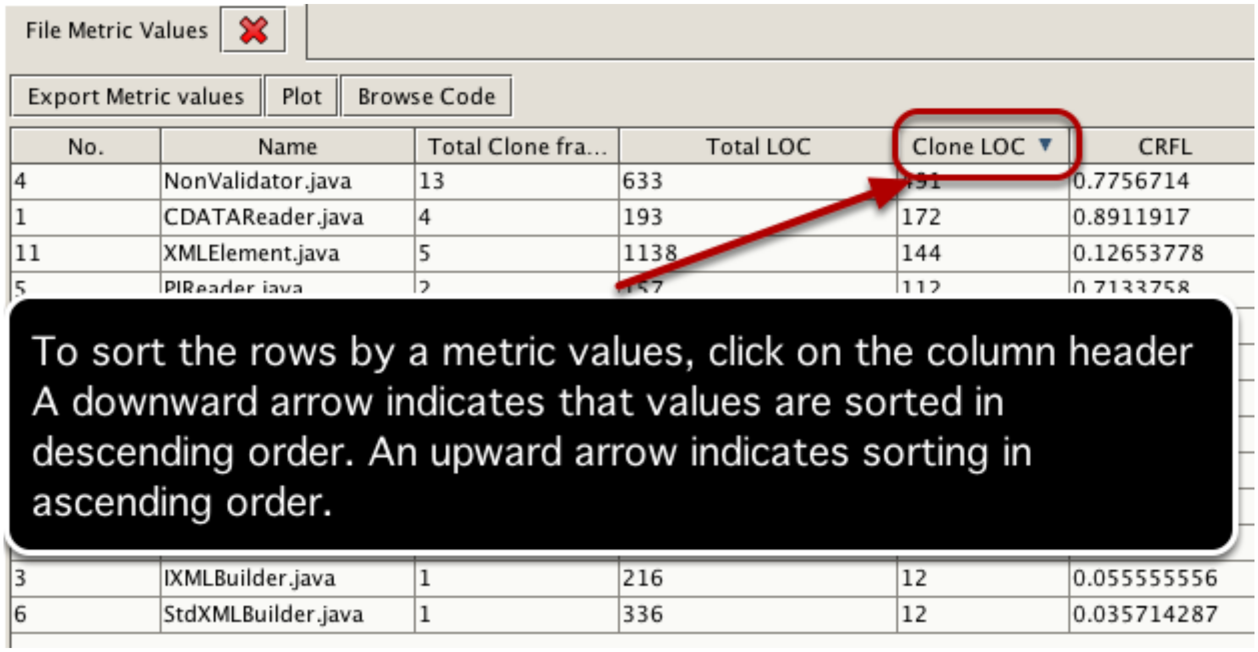
Result



	ame	Path	Total Clone fragments	Total LOC	Clone LOC	CRFL
1	CDATAReader.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/CDATAReader.java	4	193	172	0.8911917
2	ContentReader.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/ContentReader.java	2	212	36	0.16981132
3	IXMLBuilder.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/IXMLBuilder.java	1	216	12	0.055555556
4				633	491	0.7756714
5				157	112	0.7133758
6				336	12	0.035714287
7				696	78	0.112068966
8				627	40	0.06379585
9				422	14	0.033175357
10	XMLAttribute.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/XMLAttribute.java	1	151	44	0.29139072
11	XMLElement.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/XMLElement.java	5	1138	144	0.12653778
12	XMLParserFactory.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/XMLParserFactory.java	2	150	36	0.24
13	XMLUtil.java	JHotDraw7.6/Source/jhotdraw7/src/main/java/net/n3/nanoxml/XMLUtil.java	1	763	14	0.018348623

The computed metrics values for all clone files will be displayed as a new tab panel(titled *File metric Values*) in the viewer window.

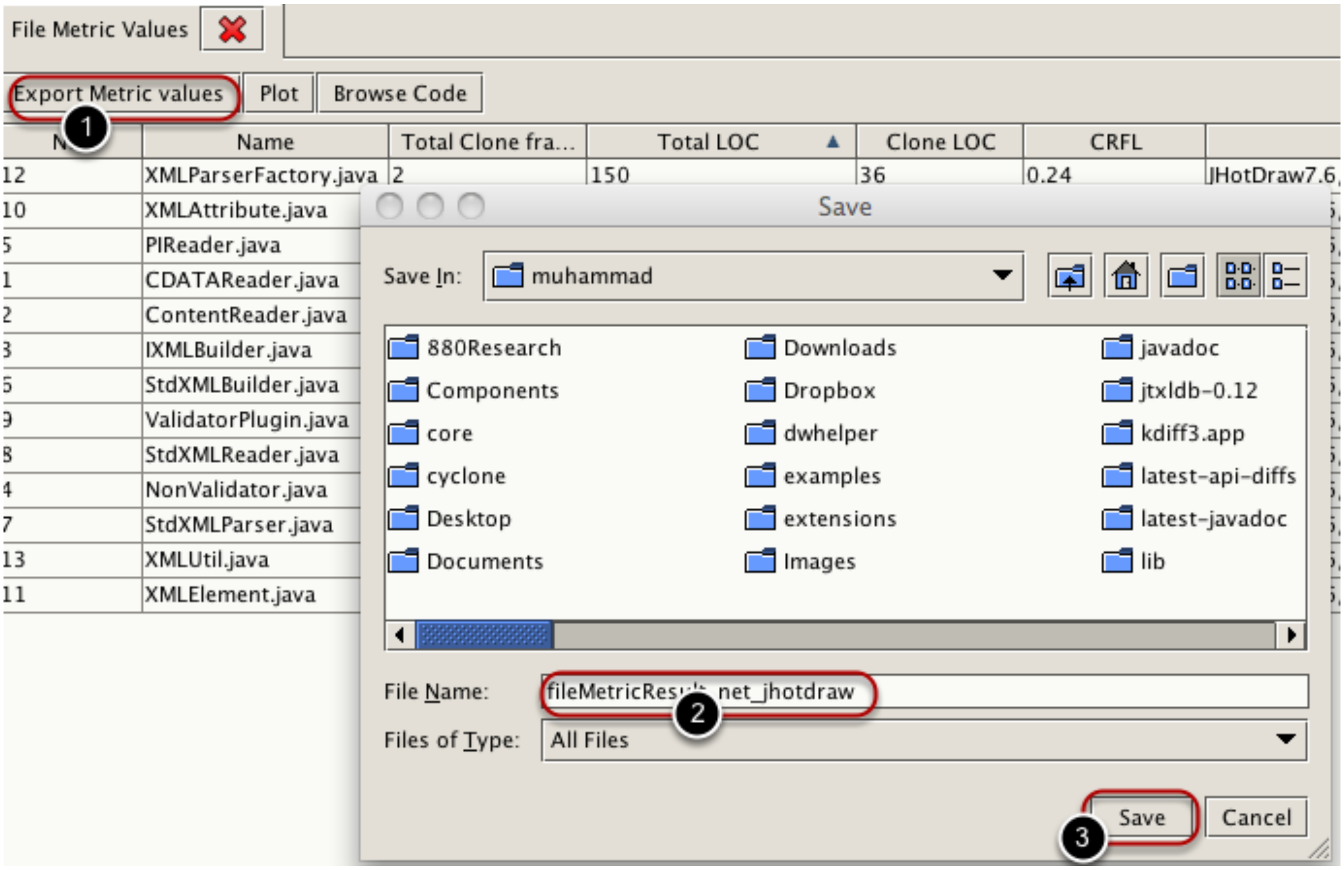
Sorting metric values



No.	Name	Total Clone fra...	Total LOC	Clone LOC ▼	CRFL
4	NonValidator.java	13	633	491	0.7756714
1	CDATAReader.java	4	193	172	0.8911917
11	XMLElement.java	5	1138	144	0.12653778
5	PIReader.java	2	157	112	0.7133758
3	IXMLBuilder.java	1	216	12	0.055555556
6	StdXMLBuilder.java	1	336	12	0.035714287

To sort the rows by a metric values, click on the column header A downward arrow indicates that values are sorted in descending order. An upward arrow indicates sorting in ascending order.

Exporting/Saving metric values



No.	Name	Total Clone fra...	Total LOC ▲	Clone LOC	CRFL	
12	XMLParserFactory.java	2	150	36	0.24	JHotDraw7.6,
10	XMLAttribute.java					
5	PIReader.java					
1	CDATAReader.java					
2	ContentReader.java					
3	IXMLBuilder.java					
5	StdXMLBuilder.java					
9	ValidatorPlugin.java					
8	StdXMLReader.java					
4	NonValidator.java					
7	StdXMLParser.java					
13	XMLUtil.java					
11	XMLElement.java					

File Name: fileMetricResult_net_jhotdraw

Files of Type: All Files

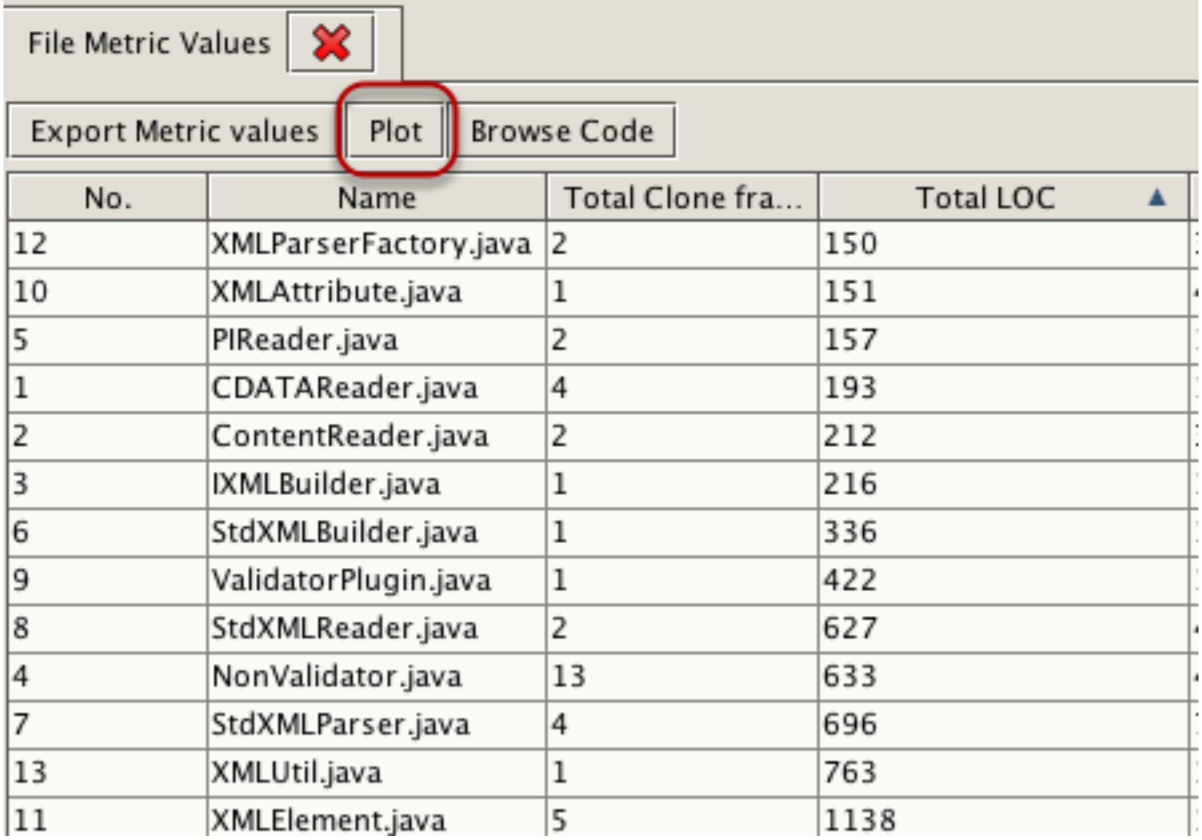
To save the result, follow the following steps:

Step-1: To save the result, click on the *Export Metric Values* button. This will open a dialog box. Select the folder where you want to save the result.

Step-2: Provide the name of the file you want to save .

Step-3: Click on the Save button to complete the operation.

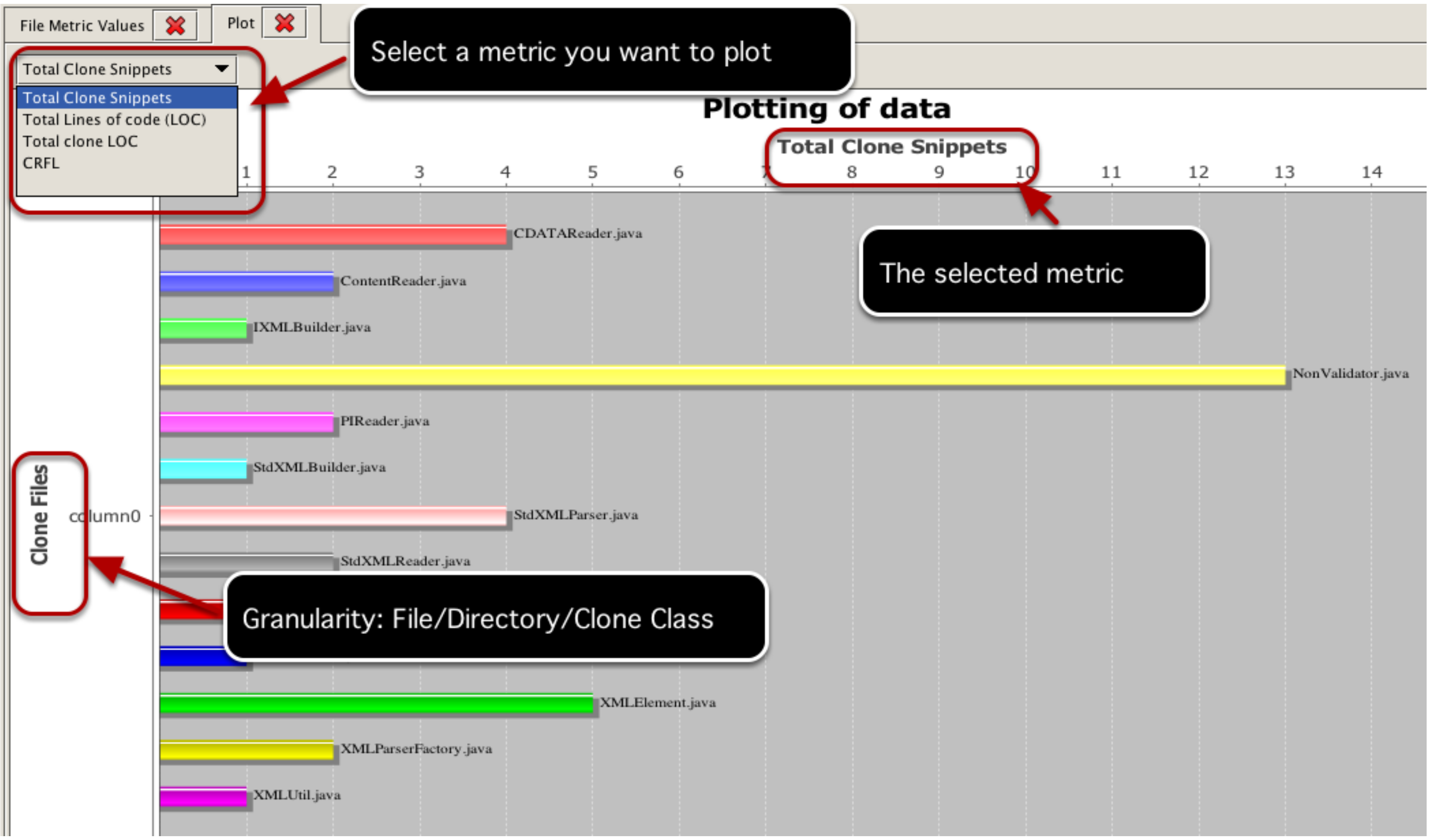
Plotting metric values



No.	Name	Total Clone fra...	Total LOC ▲
12	XMLParserFactory.java	2	150
10	XMLAttribute.java	1	151
5	PIReader.java	2	157
1	CDATAReader.java	4	193
2	ContentReader.java	2	212
3	IXMLBuilder.java	1	216
6	StdXMLBuilder.java	1	336
9	ValidatorPlugin.java	1	422
8	StdXMLReader.java	2	627
4	NonValidator.java	13	633
7	StdXMLParser.java	4	696
13	XMLUtil.java	1	763
11	XMLElement.java	5	1138

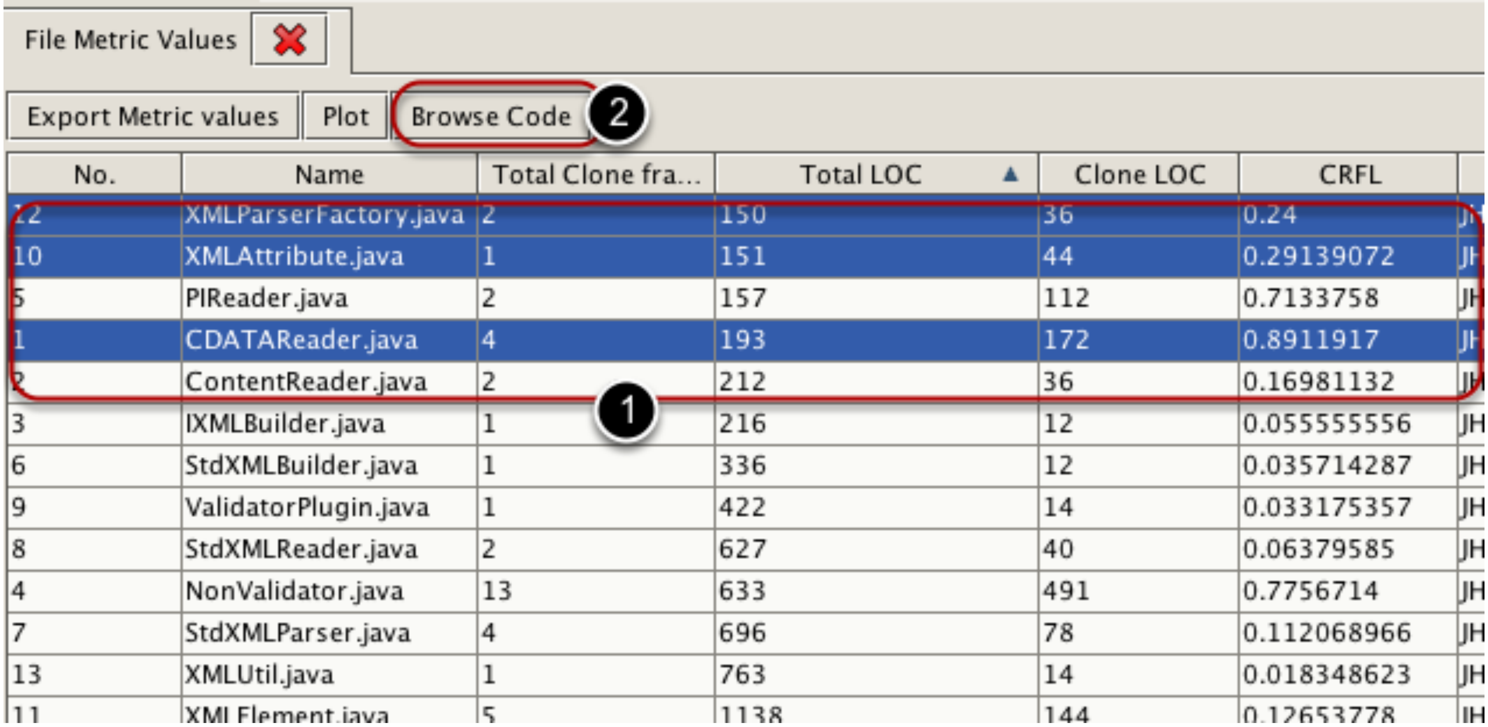
To plot the metric values, click the on the *Plot* button.

Plotting metric values (Continued)



You can change the selection to plot different metric values. Putting the cursor over a bar provides the detail path information.

Browsing clone code fragments



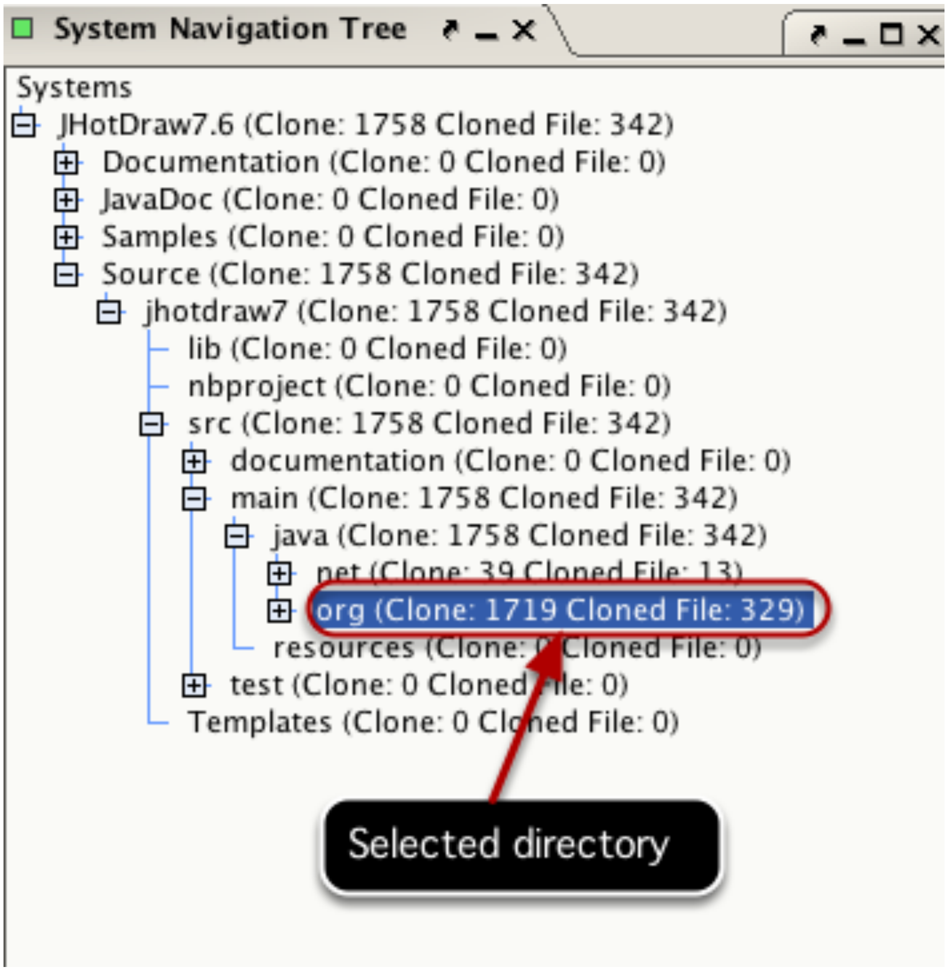
No.	Name	Total Clone fra...	Total LOC ▲	Clone LOC	CRFL	
12	XMLParserFactory.java	2	150	36	0.24	JH
10	XMLAttribute.java	1	151	44	0.29139072	JH
5	PIReader.java	2	157	112	0.7133758	JH
1	CDATAReader.java	4	193	172	0.8911917	JH
2	ContentReader.java	2	212	36	0.16981132	JH
3	IXMLBuilder.java	1	216	12	0.055555556	JH
6	StdXMLBuilder.java	1	336	12	0.035714287	JH
9	ValidatorPlugin.java	1	422	14	0.033175357	JH
8	StdXMLReader.java	2	627	40	0.06379585	JH
4	NonValidator.java	13	633	491	0.7756714	JH
7	StdXMLParser.java	4	696	78	0.112068966	JH
13	XMLUtil.java	1	763	14	0.018348623	JH
11	XMLElement.java	5	1138	144	0.12653778	JH

Step-1: Select one or more rows by holding the command key in MAC or control key in Windows.
Step-2: Now click on the *Browse Code* Button to analyze the clone code fragments in the selected file(s) with the *source code browser*.

Obtaining Metrics For Directories

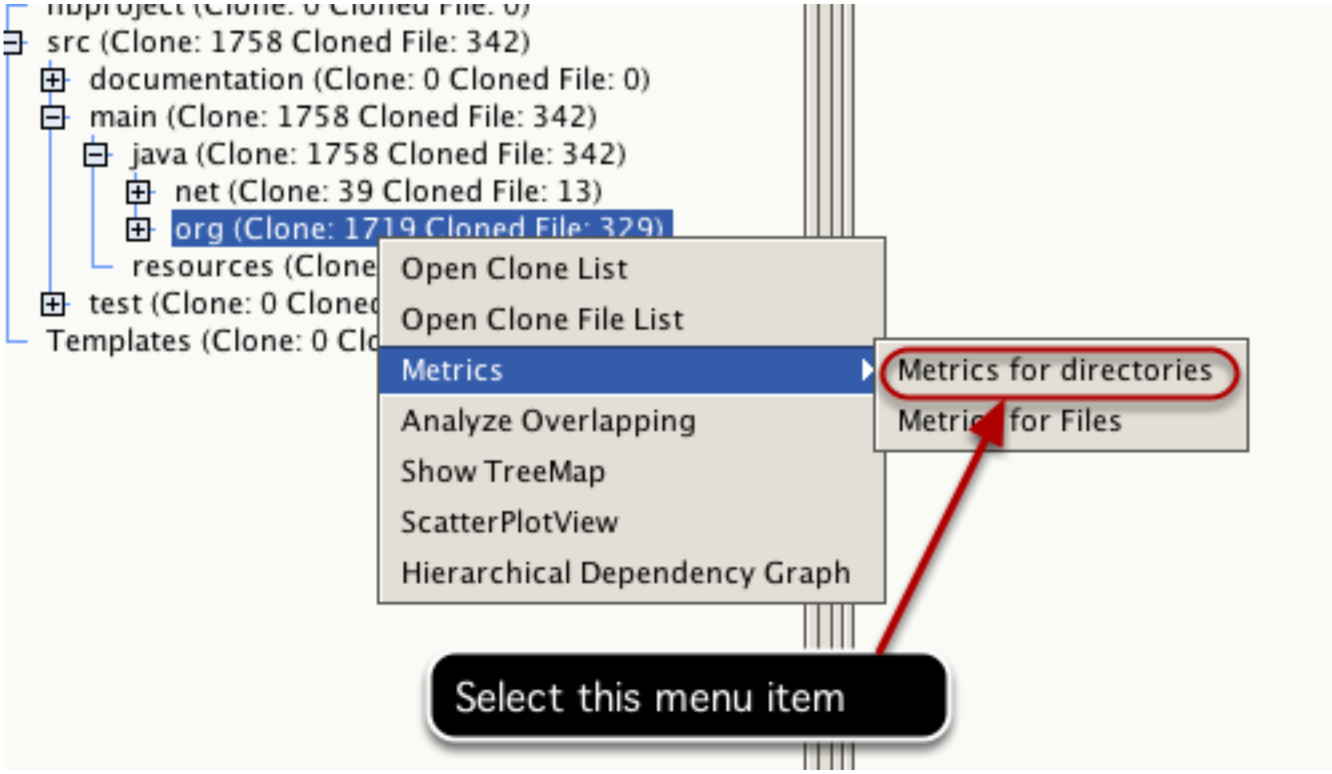
This section discusses the steps for obtaining various metrics values for all clone directories within a selected directory. A clone directory is a directory that contains at least one clone fragment.

Select a directory



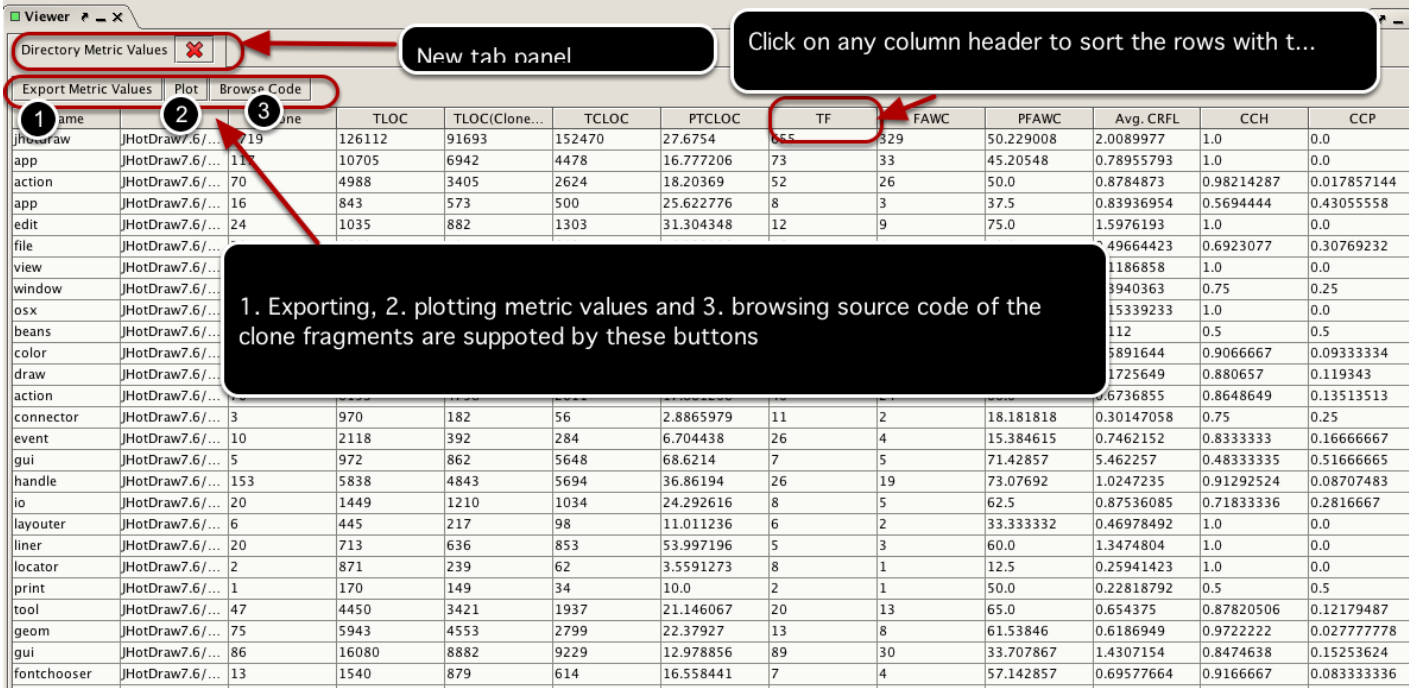
Click on a target directory to select it from the system navigation tree. Metrics will be computed for all clone directories within it.

Open the popup menu and select the operation



Right click on the selected directory to open the popup menu. Select *Metrics->Metrics for directories* from the menu.

Result



Click on any column header to sort the rows with t...

1. Exporting, 2. plotting metric values and 3. browsing source code of the clone fragments are suppoted by these buttons

ame		one	TLOC	TLOC(Clone...	TCLOC	PTCLOC	TF	FAWC	PFAWC	Avg. CRFL	CCH	CCP
[illegible]	JHotDraw7.6/...	[illegible]	126112	91693	152470	27.6754	655	329	50.229008	2.0089977	1.0	0.0
app	JHotDraw7.6/...	117	10705	6942	4478	16.777206	73	33	45.20548	0.78955793	1.0	0.0
action	JHotDraw7.6/...	70	4988	3405	2624	18.20369	52	26	50.0	0.8784873	0.98214287	0.017857144
app	JHotDraw7.6/...	16	843	573	500	25.622776	8	3	37.5	0.83936954	0.5694444	0.43055558
edit	JHotDraw7.6/...	24	1035	882	1303	31.304348	12	9	75.0	1.5976193	1.0	0.0
file	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.49664423	0.6923077	0.30769232
view	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1186858	1.0	0.0
window	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3940363	0.75	0.25
osx	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	15339233	1.0	0.0
beans	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	112	0.5	0.5
color	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	5891644	0.9066667	0.09333334
draw	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1725649	0.880657	0.119343
action	JHotDraw7.6/...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.6736855	0.8648649	0.13513513
connector	JHotDraw7.6/...	3	970	182	56	2.8865979	11	2	18.181818	0.30147058	0.75	0.25
event	JHotDraw7.6/...	10	2118	392	284	6.704438	26	4	15.384615	0.7462152	0.8333333	0.16666667
gui	JHotDraw7.6/...	5	972	862	5648	68.6214	7	5	71.42857	5.462257	0.48333335	0.51666665
handle	JHotDraw7.6/...	153	5838	4843	5694	36.86194	26	19	73.07692	1.0247235	0.91292524	0.08707483
io	JHotDraw7.6/...	20	1449	1210	1034	24.292616	8	5	62.5	0.87536085	0.71833336	0.2816667
layouter	JHotDraw7.6/...	6	445	217	98	11.011236	6	2	33.333332	0.46978492	1.0	0.0
liner	JHotDraw7.6/...	20	713	636	853	53.997196	5	3	60.0	1.3474804	1.0	0.0
locator	JHotDraw7.6/...	2	871	239	62	3.5591273	8	1	12.5	0.25941423	1.0	0.0
print	JHotDraw7.6/...	1	170	149	34	10.0	2	1	50.0	0.22818792	0.5	0.5
tool	JHotDraw7.6/...	47	4450	3421	1937	21.146067	20	13	65.0	0.654375	0.87820506	0.12179487
geom	JHotDraw7.6/...	75	5943	4553	2799	22.37927	13	8	61.53846	0.6186949	0.9722222	0.027777778
gui	JHotDraw7.6/...	86	16080	8882	9229	12.978856	89	30	33.707867	1.4307154	0.8474638	0.15253624
fontchooser	JHotDraw7.6/...	13	1540	879	614	16.558441	7	4	57.142857	0.69577664	0.9166667	0.083333336

The computed metrics values for all clone directories will be displayed as a new tab panel(titled *Directory Metric Values*) in the viewer window.

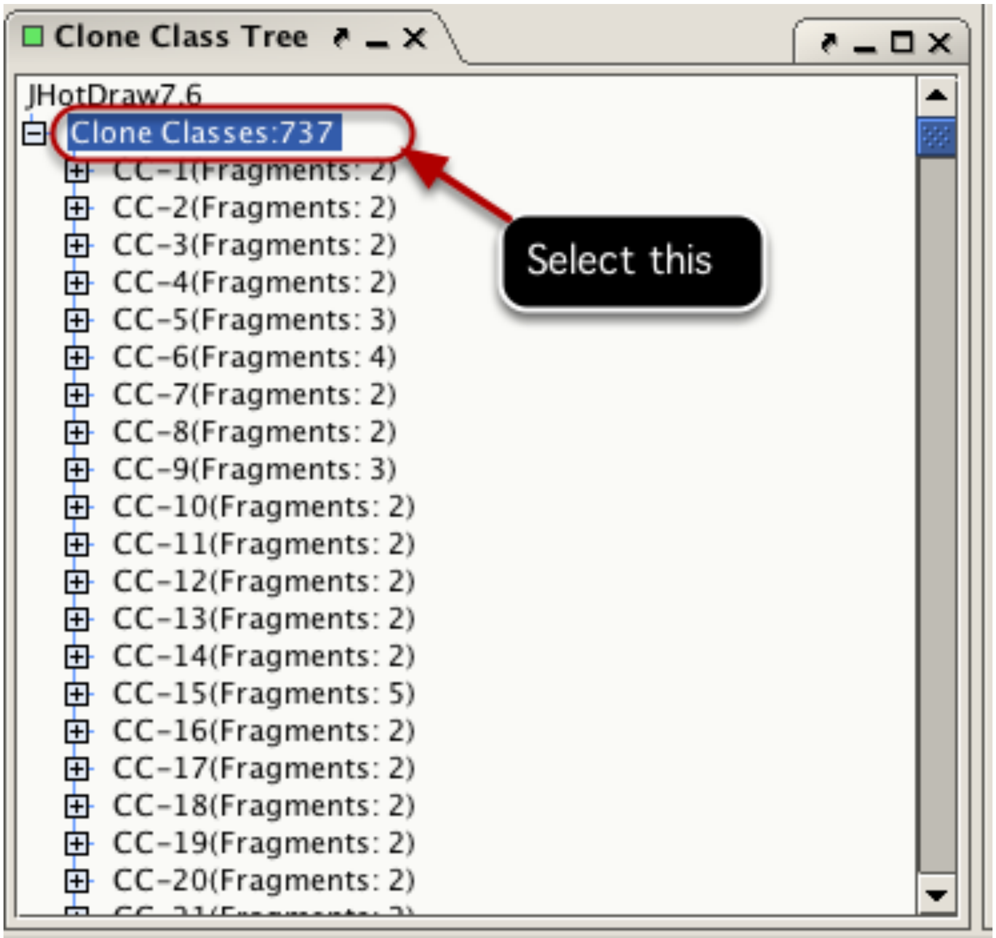
Other operations

Various operations such as sorting, exporting and plotting computed metrics values, and browsing source code of the clone fragments are also supported. For details, see those operations in the discussion for files (Obtaining Metrics For Files).

Obtaining Metrics For Clone Classes

This section discusses the steps for obtaining various metrics values for the clone classes.

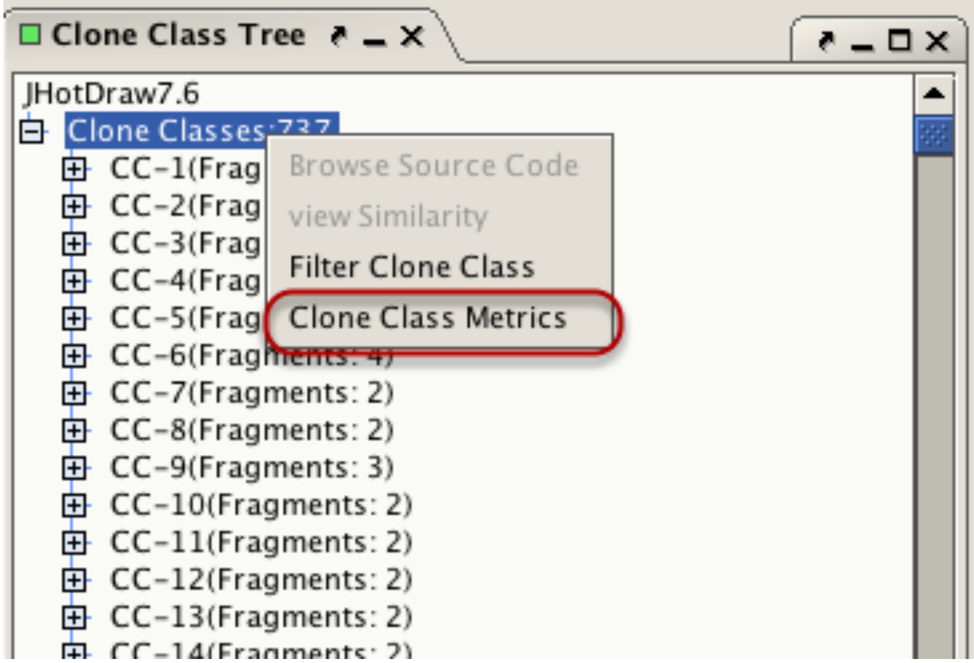
Make the selection



Select this

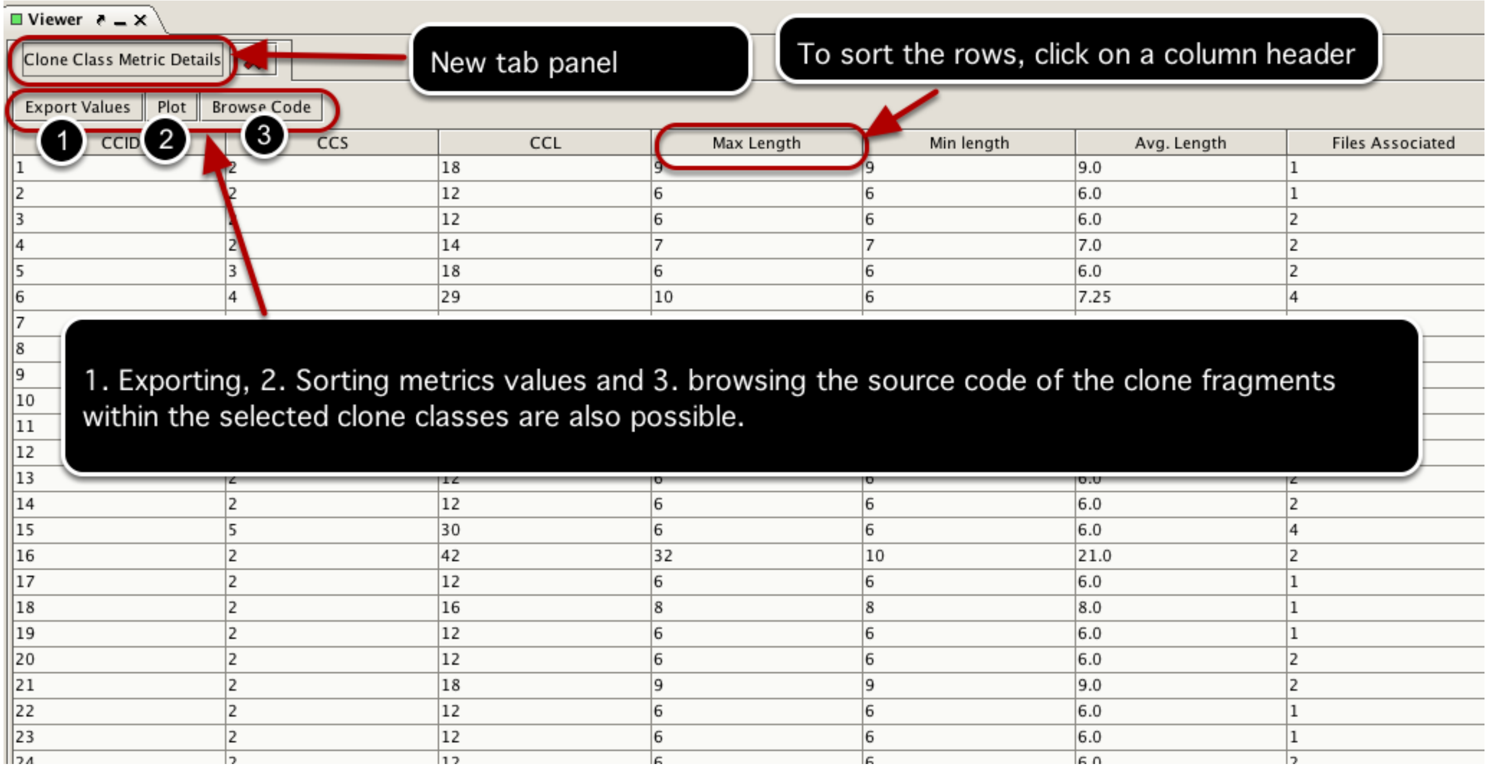
Click on the root node of the clone classes to select it from the clone class tree. Metrics will be computed for all clone classes within it.

Open the popup menu and select the operation



Right click on the selected node to open the popup menu. Select *Clone Class metrics* menu item from the popup menu.

Result



New tab panel

To sort the rows, click on a column header

1. Exporting, 2. Sorting metrics values and 3. browsing the source code of the clone fragments within the selected clone classes are also possible.

CCID	CCS	CCL	Max Length	Min length	Avg. Length	Files Associated
1	2	18	9	9	9.0	1
2	2	12	6	6	6.0	1
3	2	12	6	6	6.0	2
4	2	14	7	7	7.0	2
5	3	18	6	6	6.0	2
6	4	29	10	6	7.25	4
7						
8						
9						
10						
11						
12						
13	2	12	6	6	6.0	2
14	2	12	6	6	6.0	2
15	5	30	6	6	6.0	4
16	2	42	32	10	21.0	2
17	2	12	6	6	6.0	1
18	2	16	8	8	8.0	1
19	2	12	6	6	6.0	1
20	2	12	6	6	6.0	2
21	2	18	9	9	9.0	2
22	2	12	6	6	6.0	1
23	2	12	6	6	6.0	1
24	2	12	6	6	6.0	2

The computed metrics values for all clone classes will be displayed as a new tab panel(titled *Clone Class Metric Details*) in the viewer window.

Other operations

Various operations such as sorting, exporting and plotting computed metrics values, and browsing source code of the clone fragments are also supported. For details, see those operations in the discussion for files (Obtaining Metrics For Files).

Filtering

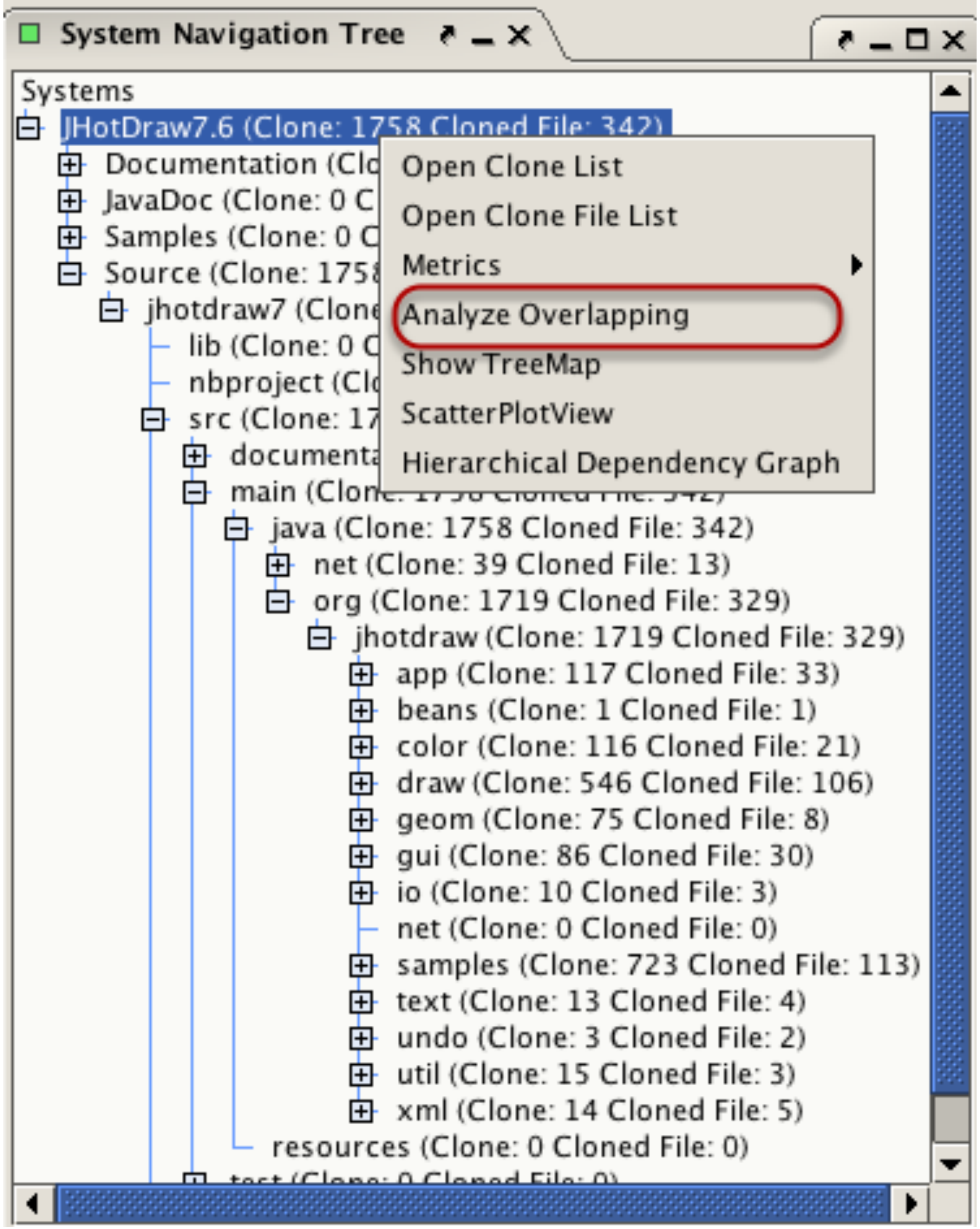
Introduction

The first and foremost challenge in clone analysis is the large volume of clone detection results. Not all clones are useful to the user and the objective of the analysis at hand governs the set of the useful clones. Here, the term ‘useful clones’ refers to those clone fragments that the maintenance engineers are looking for or are interested in. For example, when the objective is to analyze the inter-project clones, users may be more interested in the clone classes whose fragments are distributed across different projects and these clone fragments form the set of useful clones. In that case, we can filter out the clone pairs that are not distributed across different projects. VisCad supports a set of filtering operations to remove clones that are not useful/interesting to the users.

Overlapping Clone filterng

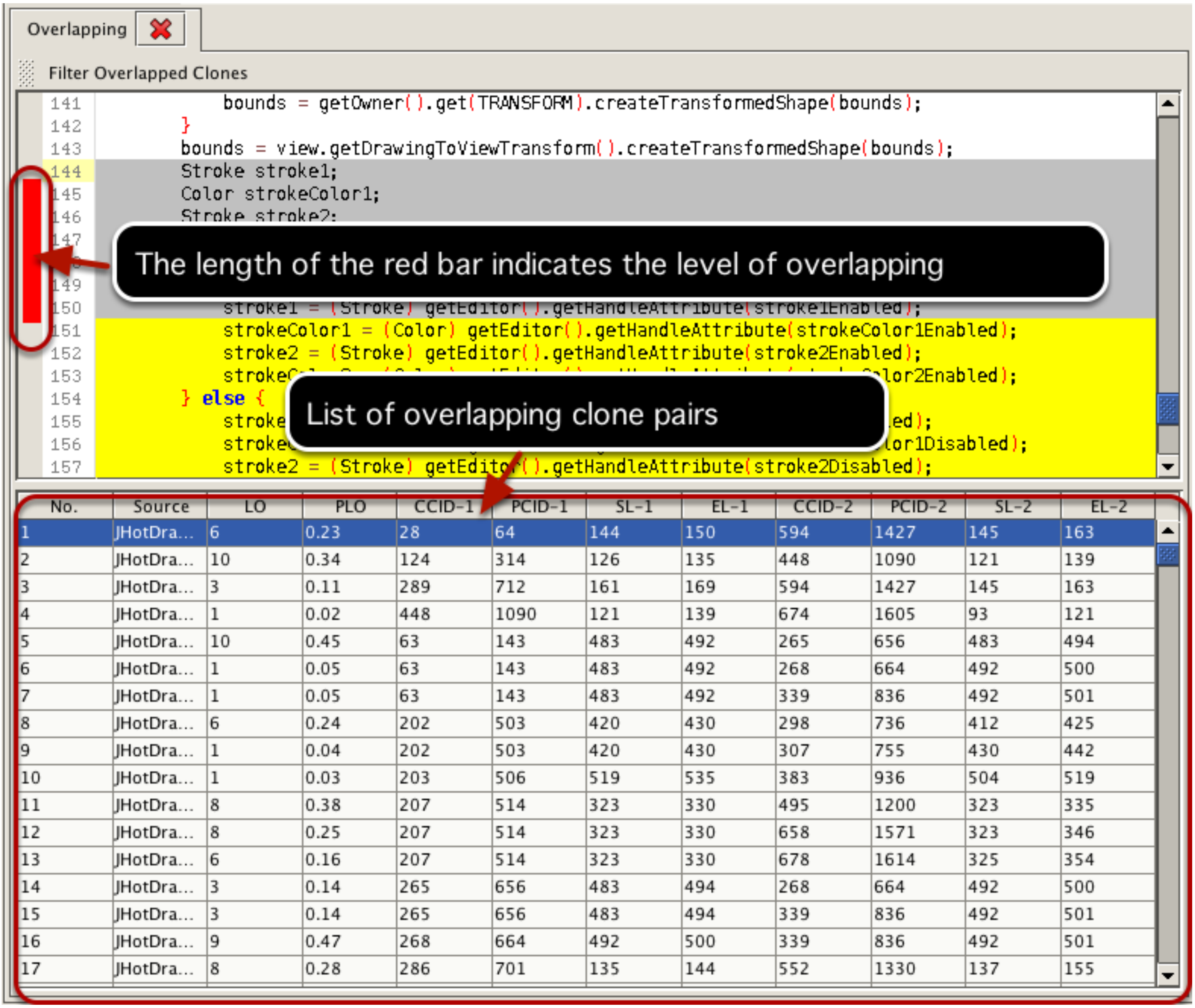
Clones may overlapped each other and removing those overlapping clones also reduces the size of the result set.

Analyze overlapping



Click on a directory to selct it. Right click on it to open the popup menu and select *Analyze Overlapping* menu item.

Overlapping Clone pairs list



```
            bounds = getOwner().get(TRANSFORM).createTransformedShape(bounds);
        }
        bounds = view.getDrawingToViewTransform().createTransformedShape(bounds);
        Stroke stroke1;
        Color strokeColor1;
        Stroke stroke2;

            stroke1 = (Stroke) getEditor().getHandleAttribute(stroke1Enabled);
            strokeColor1 = (Color) getEditor().getHandleAttribute(strokeColor1Enabled);
            stroke2 = (Stroke) getEditor().getHandleAttribute(stroke2Enabled);
            strokeC                                              olor2Enabled);
        } else {
            stroke                                                       ed);
            stroke                                                      lor1Disabled);
            stroke2 = (Stroke) getEditor().getHandleAttribute(stroke2Disabled);
```

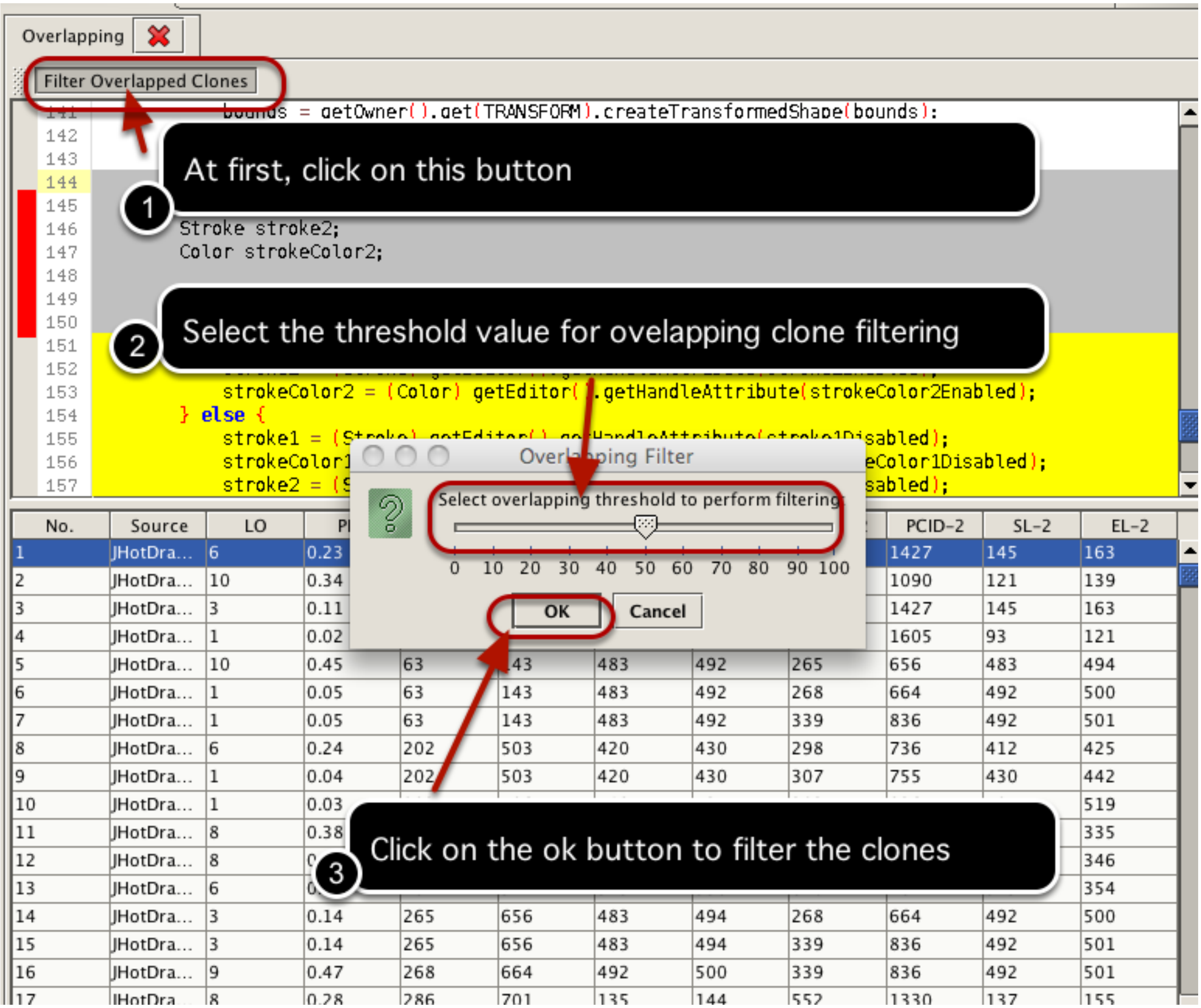
The length of the red bar indicates the level of overlapping

List of overlapping clone pairs

No.	Source	LO	PLO	CCID-1	PCID-1	SL-1	EL-1	CCID-2	PCID-2	SL-2	EL-2
1	JHotDra...	6	0.23	28	64	144	150	594	1427	145	163
2	JHotDra...	10	0.34	124	314	126	135	448	1090	121	139
3	JHotDra...	3	0.11	289	712	161	169	594	1427	145	163
4	JHotDra...	1	0.02	448	1090	121	139	674	1605	93	121
5	JHotDra...	10	0.45	63	143	483	492	265	656	483	494
6	JHotDra...	1	0.05	63	143	483	492	268	664	492	500
7	JHotDra...	1	0.05	63	143	483	492	339	836	492	501
8	JHotDra...	6	0.24	202	503	420	430	298	736	412	425
9	JHotDra...	1	0.04	202	503	420	430	307	755	430	442
10	JHotDra...	1	0.03	203	506	519	535	383	936	504	519
11	JHotDra...	8	0.38	207	514	323	330	495	1200	323	335
12	JHotDra...	8	0.25	207	514	323	330	658	1571	323	346
13	JHotDra...	6	0.16	207	514	323	330	678	1614	325	354
14	JHotDra...	3	0.14	265	656	483	494	268	664	492	500
15	JHotDra...	3	0.14	265	656	483	494	339	836	492	501
16	JHotDra...	9	0.47	268	664	492	500	339	836	492	501
17	JHotDra...	8	0.28	286	701	135	144	552	1330	137	155

VisCad analyzes and shows the list of overlapping clone pairs.

Set threshold value for overlapping clone filtering



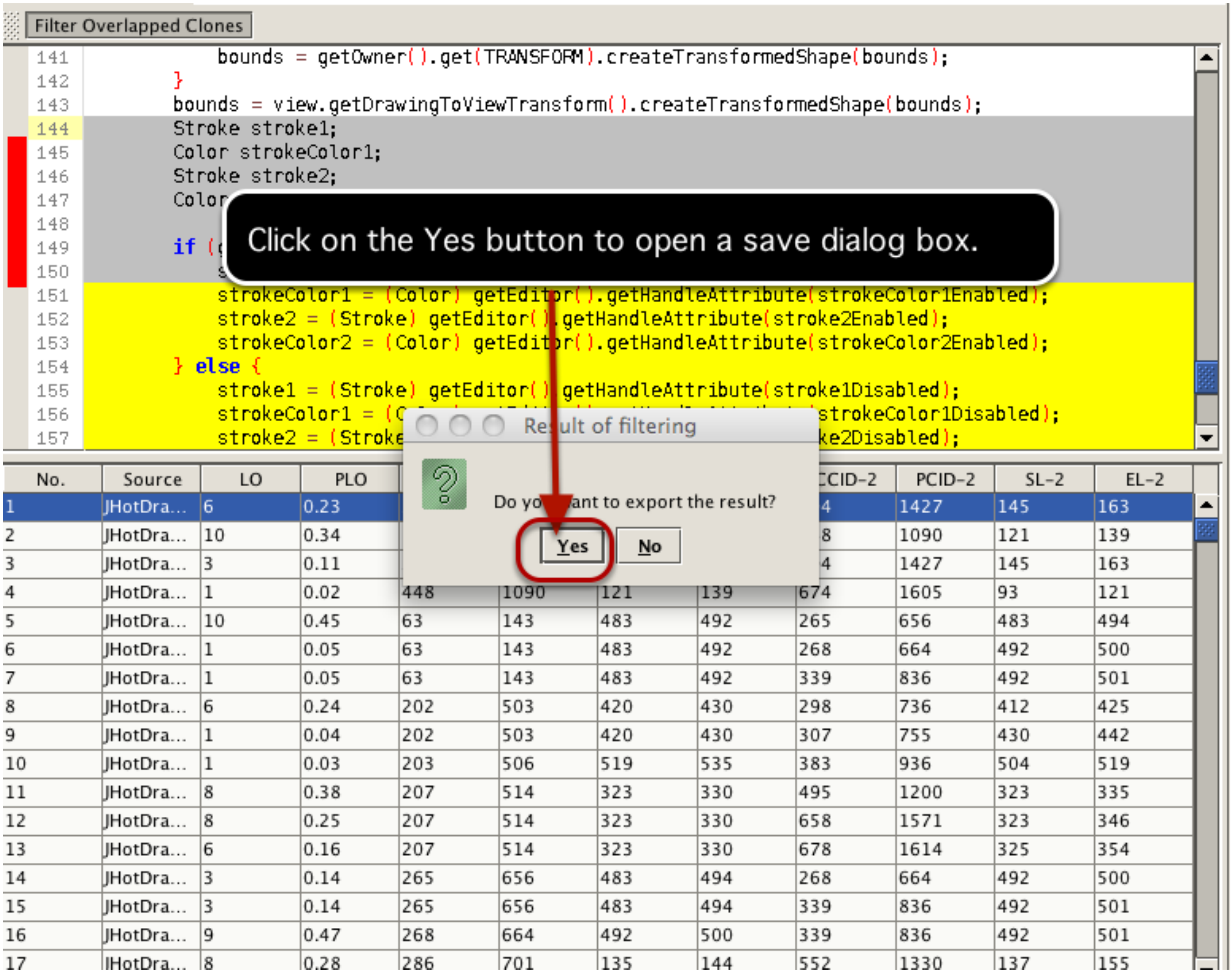
1 At first, click on this button

2 Select the threshold value for ovelapping clone filtering

Select overlapping threshold to perform filtering

3 Click on the ok button to filter the clones

Saving the result



Click on the Yes button to open a save dialog box.

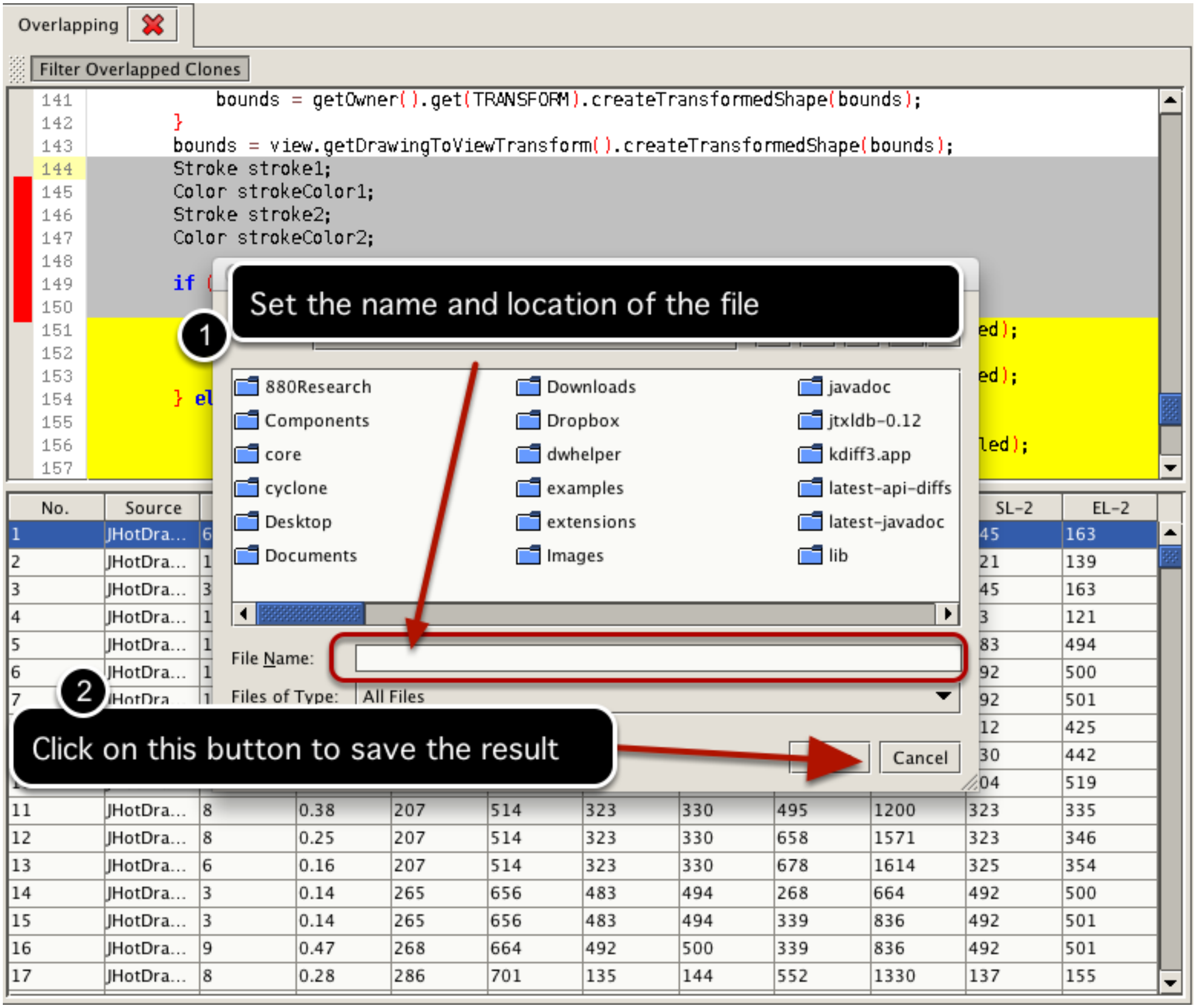
```
            bounds = getOwner().get(TRANSFORM).createTransformedShape(bounds);
        }
        bounds = view.getDrawingToViewTransform().createTransformedShape(bounds);
        Stroke stroke1;
        Color strokeColor1;
        Stroke stroke2;
        Color

        if (
            s
            strokeColor1 = (Color) getEditor().getHandleAttribute(strokeColor1Enabled);
            stroke2 = (Stroke) getEditor().getHandleAttribute(stroke2Enabled);
            strokeColor2 = (Color) getEditor().getHandleAttribute(strokeColor2Enabled);
        } else {
            stroke1 = (Stroke) getEditor().getHandleAttribute(stroke1Disabled);
            strokeColor1 = (C                              strokeColor1Disabled);
            stroke2 = (Stroke                              ke2Disabled);
```

Result of filtering

Do yo ant to export the result?

No.	Source	LO	PLO					CCID-2	PCID-2	SL-2	EL-2
1	JHotDra...	6	0.23					4	1427	145	163
2	JHotDra...	10	0.34					8	1090	121	139
3	JHotDra...	3	0.11					4	1427	145	163
4	JHotDra...	1	0.02	448	1090	121	139	674	1605	93	121
5	JHotDra...	10	0.45	63	143	483	492	265	656	483	494
6	JHotDra...	1	0.05	63	143	483	492	268	664	492	500
7	JHotDra...	1	0.05	63	143	483	492	339	836	492	501
8	JHotDra...	6	0.24	202	503	420	430	298	736	412	425
9	JHotDra...	1	0.04	202	503	420	430	307	755	430	442
10	JHotDra...	1	0.03	203	506	519	535	383	936	504	519
11	JHotDra...	8	0.38	207	514	323	330	495	1200	323	335
12	JHotDra...	8	0.25	207	514	323	330	658	1571	323	346
13	JHotDra...	6	0.16	207	514	323	330	678	1614	325	354
14	JHotDra...	3	0.14	265	656	483	494	268	664	492	500
15	JHotDra...	3	0.14	265	656	483	494	339	836	492	501
16	JHotDra...	9	0.47	268	664	492	500	339	836	492	501
17	JHotDra...	8	0.28	286	701	135	144	552	1330	137	155

Saving the result(Continued)



1 Set the name and location of the file

2 Click on this button to save the result

```
            bounds = getOwner().get(TRANSFORM).createTransformedShape(bounds);
        }
        bounds = view.getDrawingToViewTransform().createTransformedShape(bounds);
        Stroke stroke1;
        Color strokeColor1;
        Stroke stroke2;
        Color strokeColor2;

        if (
                                                                        ed);

                                                                        ed);
        } el

                                                                        led);
```

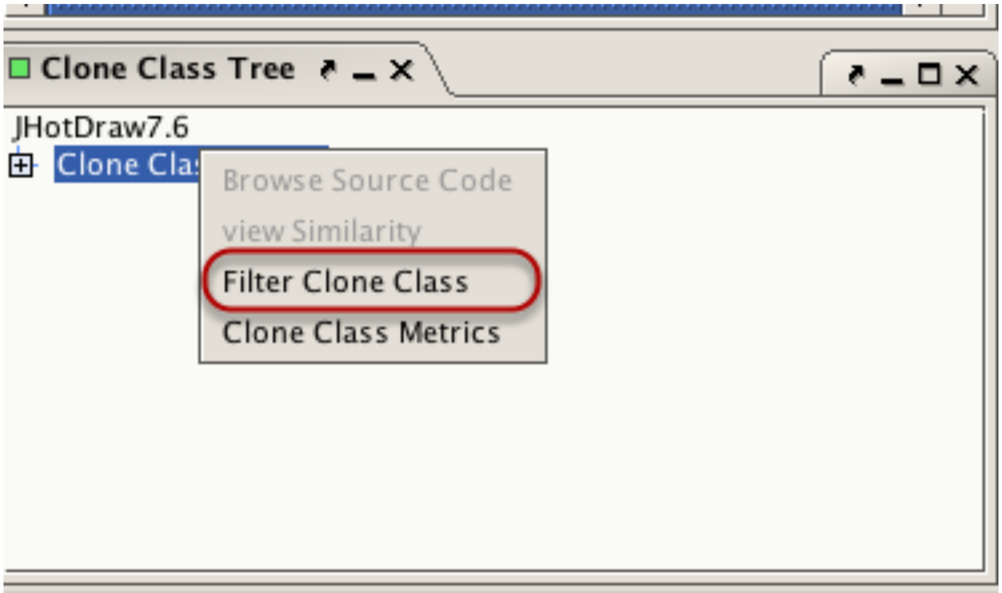
No.	Source									SL-2	EL-2
1	JHotDra...	6									163
2	JHotDra...	1									139
3	JHotDra...	3									163
4	JHotDra...	1									121
5	JHotDra...	1									494
6	JHotDra...	1									500
7		1									501
											425
											442
											519
11	JHotDra...	8	0.38	207	514	323	330	495	1200	323	335
12	JHotDra...	8	0.25	207	514	323	330	658	1571	323	346
13	JHotDra...	6	0.16	207	514	323	330	678	1614	325	354
14	JHotDra...	3	0.14	265	656	483	494	268	664	492	500
15	JHotDra...	3	0.14	265	656	483	494	339	836	492	501
16	JHotDra...	9	0.47	268	664	492	500	339	836	492	501
17	JHotDra...	8	0.28	286	701	135	144	552	1330	137	155

VisCad saves the filtered clone detection result as a *VisCad input file format*.

Textual filtering

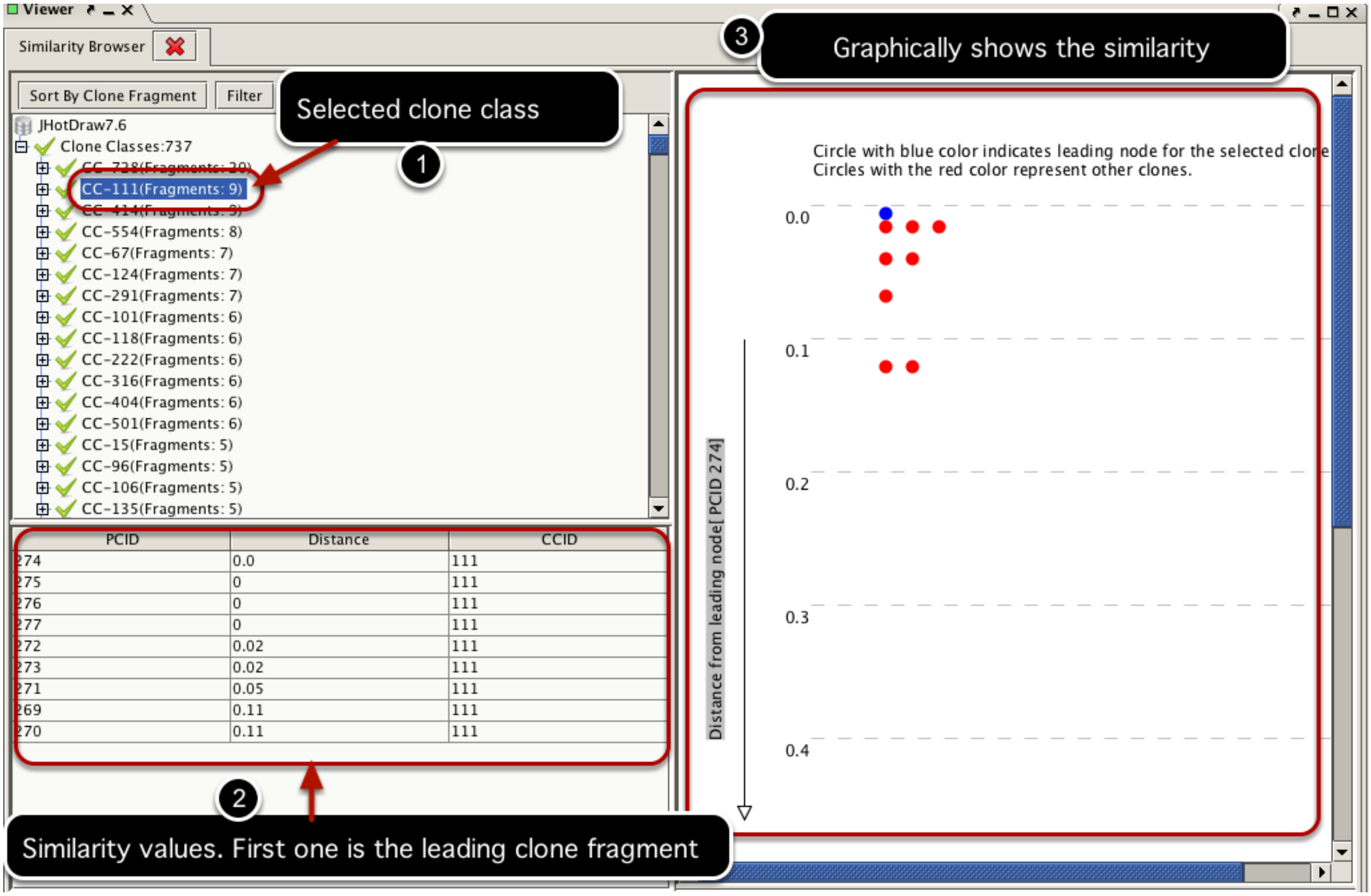
Textual filtering allows to remove clones that are only structurally similar without having any semantic similariy. For each clone class, VisCad determines the clone fragment that maximizes the sum of the textual similarity to all other fragments of that class. We call this fragment as the ‘leading clone fragment’ for that class. If the textual similarity between the ‘leading clone fragment’ and any other clone fragment in the clone class falls below a given threshold value, we remove the fragment from the analysis. We discard an entire clone class from the analysis when the textual similarities between the leading clone fragment and all other non-leading clone fragments of that clone class fall below the threshold value.

Make the selection



Select the *Clone Classes* node from the *Clone Class Tree.* Right click on it to bring the popup menu and select the *Filter Clone Class* option. This open the *similarity browser* view.

Similarity Browser View



PCID	Distance	CCID
274	0.0	111
275	0	111
276	0	111
277	0	111
272	0.02	111
273	0.02	111
271	0.05	111
269	0.11	111
270	0.11	111

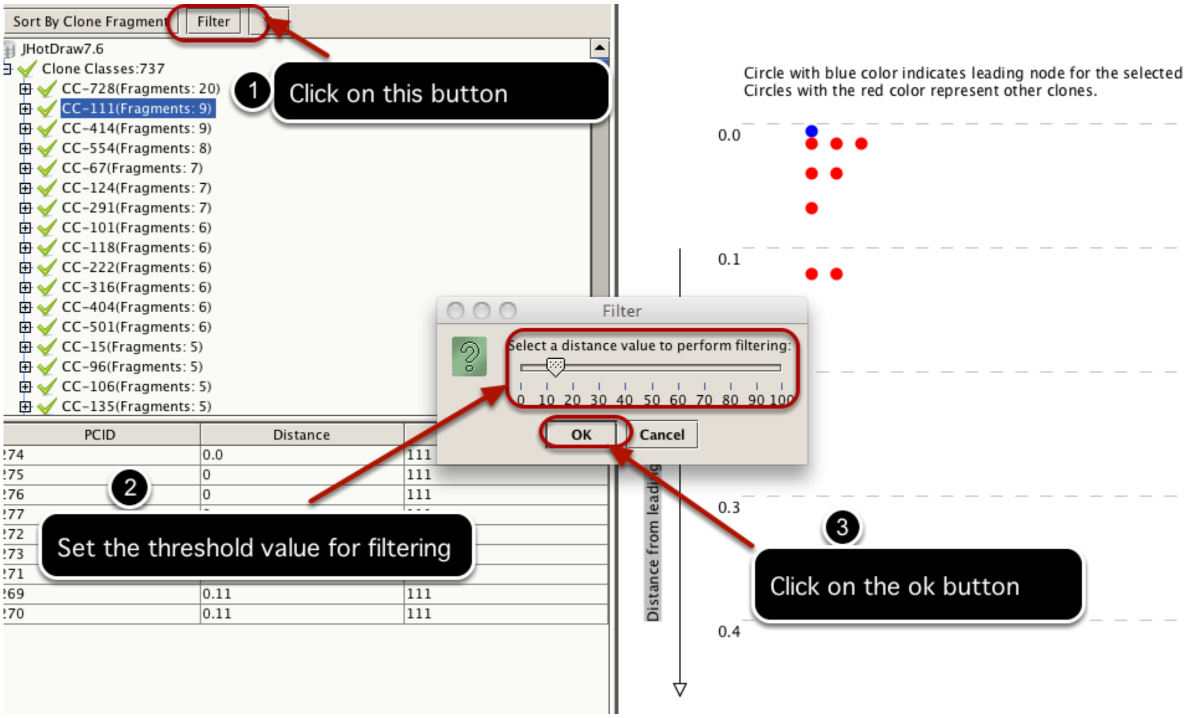
Circle with blue color indicates leading node for the selected clone
Circles with the red color represent other clones.

1 Selected clone class

2 Similarity values. First one is the leading clone fragment

3 Graphically shows the similarity

Filter the result



Circle with blue color indicates leading node for the selected
Circles with the red color represent other clones.

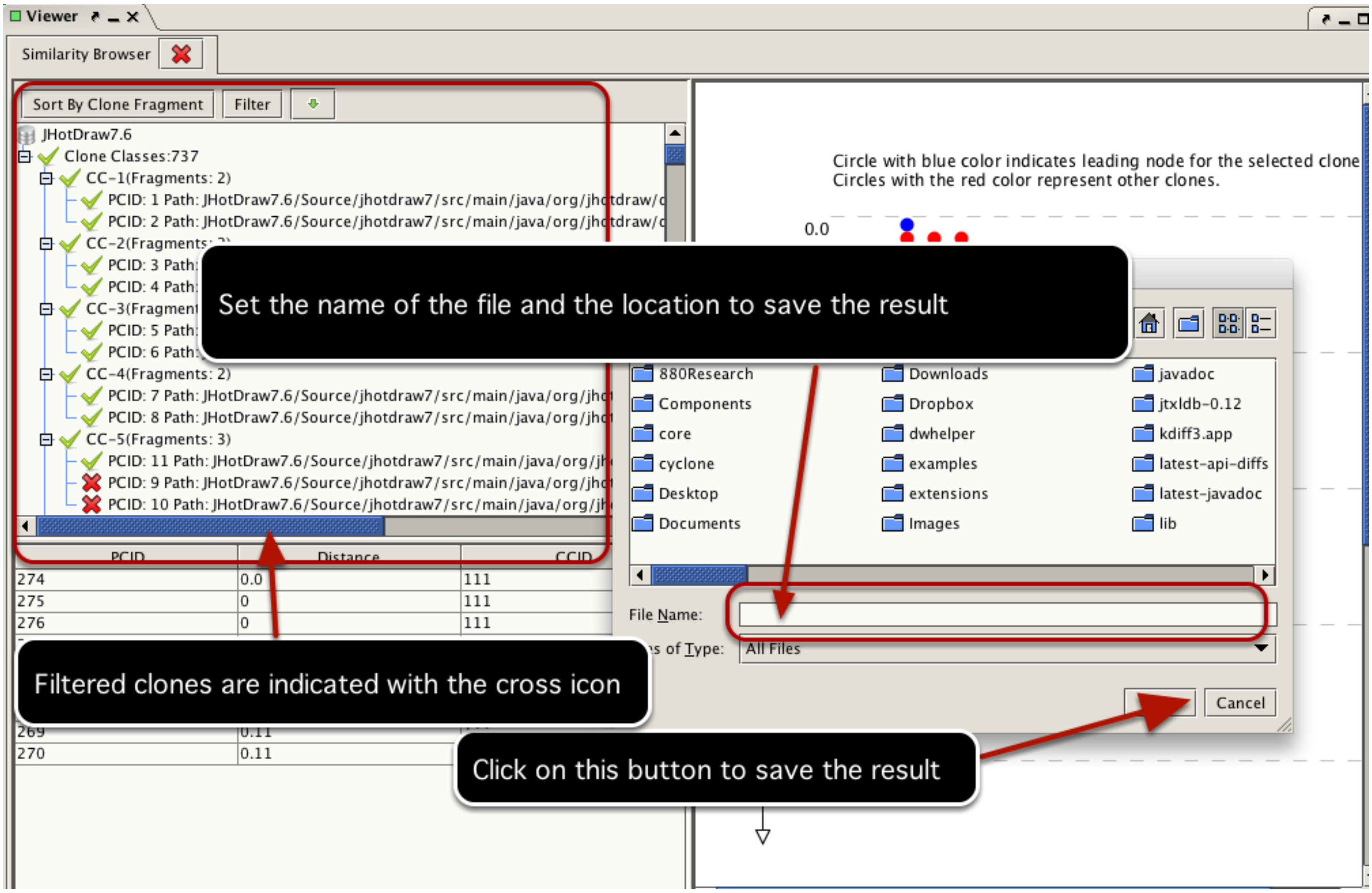
Select a distance value to perform filtering:

1 Click on this button

2 Set the threshold value for filtering

3 Click on the ok button

Filter the result(Continued)



Set the name of the file and the location to save the result

Filtered clones are indicated with the cross icon

Click on this button to save the result

VisCad saves the filtered clone detection result in *VisCad input file format*.