



CloneTracker User's Guide

Fixstars Corporation
v1.0_en

Contents

[Contents](#)

[1 Run Analysis](#)

[2. Overview of analysis results](#)

[3. Duplicate code view](#)

[4. Settings for analysis execution](#)

[4-1. In the Project Settings](#)

[4-2. In the duplicate code view](#)

[5. Analysis log](#)

[6. How to share the analysis result](#)

[6-1. Export the analysis result](#)

[6-2. How to import the JSON file](#)

1 Run Analysis

After the project is registered, the initial analysis will be automatically performed.

The screenshot shows the CloneTracker interface with the project name 'sample'. At the top, it indicates '# of detected clones: 0' and shows risk levels: Low Risk: 0, Modified: 0, Mid Risk: 0, High Risk: 0. A red box highlights a status bar that says 'Analyzing the project. Please wait.' with a 'VIEW LOGS' button. Below this, a search bar is present, and a table with columns 'Judge', 'CT-ID', 'File', 'Line', and 'Comment' is shown. The table is empty, and a message states: 'Analyzing the project. The results will be displayed here soon. The analysis may take several hours if there are a large number of files to analyze.'

In the initial analysis, the git commit history and past source code are analyzed. The analysis traces the transition of duplicate code from the past to the present, detects duplicate code precisely, and evaluates the risk for each duplicate code. Therefore, it may take several hours to display a list of analysis results.

Once the analysis is completed, the results are displayed in a list, and the detected duplicate codes will be shown.

The screenshot shows the CloneTracker interface after analysis. It indicates '# of detected clones: 6' and shows risk levels: Low Risk: 2, Modified: 2, Mid Risk: 2, High Risk: 0. A status bar says 'Clones that have never been modified in the past'. Below this, a search bar is present, and a table with columns 'Judge', 'CT-ID', 'File', 'Line', and 'Comment' is shown. The table contains 6 rows of results, each with a star icon, a risk level, and a list of duplicate code blocks. The bottom right shows 'Rows per page: 100' and '1-6 of 6'.

Judge	CT-ID	File	Line	Comment
★	5-L	e1.cpp	L1-10	
★	5-L	e2.cpp	L1-10	
★	2-L	b1.cpp	L1-12	
★	2-L	b2.cpp	L1-12	
☆	4-L	d1.cpp	L1-21	
☆	4-L	d2.cpp	L1-21	
☆	1-L	a1.cc	L1-15	
☆	1-L	a2.cc	L1-15	
☆	6-L	f1.c	L1-29	
☆	6-L	f2.c	L1-29	
☆	3-L	c1.c	L1-12	
☆	3-L	c2.c	L1-12	

After the initial analysis, analysis is performed every 10 minutes on the latest commit.

2. Overview of analysis results

Once the analysis is done, check the duplicate code blocks.

CloneTracker
Duplicate Code Management Software

of detected clones: 6

Low Risk: 2 Modified: 2 Mid Risk: 2 High Risk: 0

Project Name: sample

ANALYZE PROJECT PROJECT SETTINGS ANALYSIS LOG

Clones that have never been modified in the past

6/ 6 items

Search by CT-ID, File

Judger	CT-ID	File	Line	Comment
★	5-L	e1.cpp	L1-10	
★	2-L	e2.cpp	L1-10	
★	2-L	b1.cpp	L1-12	
★	2-L	b2.cpp	L1-12	
☆	4-L	d1.cpp	L1-21	
☆	4-L	d2.cpp	L1-21	
☆	1-L	a1.cc	L1-15	
☆	1-L	a2.cc	L1-15	
☆	6-L	f1.c	L1-29	
☆	6-L	f2.c	L1-29	
☆	3-L	c1.c	L1-12	
☆	3-L	c2.c	L1-12	

Rows per page: 100 1-6 of 6 < >

First of all, please check the number of duplicate codes, and the breakdown of the duplicates classified as Low Risk, Modified, Mid Risk, and High Risk.

Low-Risk Duplicate :

This is a duplicate code that has never been modified in the past. Since the code has never been modified up to the time of the latest analysis, it is considered to be a “good” duplicate that does not require management.

Modified :

This is a copy-and-paste code that has been modified at least once from the time it was generated to the present. As long as the same modifications have been made, there is a possibility of inconsistencies between the duplicate codes, so we consider this to be required caution.

Mid Risk :

This is duplicate code in which inconsistencies between each duplicate code. Inconsistencies between duplicate codes may lead to bugs, so we consider them to be mid-risk duplicate codes.

High Risk :

This is a duplicate code that has been modified to align inconsistencies after inconsistencies have occurred between duplicate codes. This is a duplicate code that has been omitted modification and has a very high risk of generating bugs.

CloneTracker
Duplicate Code Management Software

of detected clones: 6

Low Risk: 2 Modified: 2 Mid Risk: 2 High Risk: 0

Project Name: sample

ANALYZE PROJECT PROJECT SETTINGS ANALYSIS LOG

Clones that have never been modified in the past

6/ 6 items

Search by CT-ID, File

Judger	CT-ID	File	Line	Comment
★	5-L	e1.cpp	L1-10	
★	2-L	e2.cpp	L1-10	
★	2-L	b1.cpp	L1-12	
★	2-L	b2.cpp	L1-12	
☆	4-L	d1.cpp	L1-21	
☆	4-L	d2.cpp	L1-21	
☆	1-L	a1.cc	L1-15	
☆	1-L	a2.cc	L1-15	
☆	6-L	f1.c	L1-29	
☆	6-L	f2.c	L1-29	
☆	3-L	c1.c	L1-12	
☆	3-L	c2.c	L1-12	

Rows per page: 100 1-6 of 6 < >

The overview list of duplicates has eight columns.

①Star:

This is a flag to be displayed at the top of the list.

Duplicates that are categorized above mid-risk will automatically be given a star.

②Risk:

Show Low-Risk, Modified, Mid-Risk, High-Risk.

③CT-ID:

This is a unique ID assigned to a duplicate code group.

④File:

Displays the file name having the duplicate code.

⑤Line:

Displays the line number in each file where the duplicate code exists.

⑥Comment:

Displays comments entered by users for a duplicate code group. It can be entered from each duplicate code view.

⑦Archive :

Click on the Archive button for the less important duplicates. The duplicate will be displayed at the bottom of the list.


⑧Trush :

Click on the Trash button for duplicates that need to be removed from the list.

Each column can be reordered in ascending or descending order, and it can be filtered.

3. Duplicate code view

The duplicate code is displayed by clicking any row on the overview of analysis list.



CT-ID: 4-L
 Importance: ★ Similar Statements: 17
 Similarity Score: 100%

MESSAGE:
 - New duplicate code has been detected at 2023-10-03 14:40:05.

COMMENT:

LEFT/RIGHT OLD/NEW Confirmed: Star

IGNORE SHARE

<input checked="" type="radio"/> src-1layer/node_modules/d1.cpp (L1-L21) View in VSCode <input type="radio"/> src-1layer/node_modules/d1.cpp (L1-L21) View in VSCode <input type="radio"/> src-1layer/node_modules/d1.cpp (L1-L21) View in VSCode <input type="radio"/> src-1layer/node_modules/d2.cpp (L1-L21) View in VSCode <input type="radio"/> src-1layer/node_modules/d2.cpp (L1-L21) View in VSCode <input type="radio"/> src-3layer/node_modules/d1.cpp (L1-L21) View in VSCode	<input type="radio"/> src-1layer/node_modules/d2.cpp (L1-L21) View in VSCode ③ <input checked="" type="radio"/> src-3layer/node_modules/d1.cpp (L1-L21) View in VSCode <input type="radio"/> src-3layer/node_modules/d2.cpp (L1-L21) View in VSCode <input type="radio"/> src-3layer/node_modules/d1.cpp (L1-L21) View in VSCode <input type="radio"/> src-3layer/node_modules/d2.cpp (L1-L21) View in VSCode <input type="radio"/> src-3layer/node_modules/d2.cpp (L1-L21) View in VSCode
<div> <div>src-1layer/node_modules/d1.cpp (L1-L21) ④</div> <pre> 1 struct ggml_tensor * get_tensor_for(llama_load_tensor & lt, ggml_backend backend) { 2 struct ggml_tensor * tensor; 3 if (backend != GGML_BACKEND_CPU) { 4 ggml_set_no_alloc(ggml_ctx, true); 5 } 6 if (lt.ne.size() == 2) { 7 tensor = ggml_new_tensor_2d(ggml_ctx, lt.type, lt.ne.at(0), lt.ne.at(1)); 8 } else { 9 LLAMA_ASSERT(lt.ne.size() == 1); 10 tensor = ggml_new_tensor_1d(ggml_ctx, lt.type, lt.ne.at(0)); 11 } 12 ggml_set_name(tensor, lt.name.c_str()); 13 LLAMA_ASSERT(!lt.ggml_tensor == NULL); // if this fails, we called get_tensor twice on the same tensor </pre> </div>	<div> <div>src-3layer/node_modules/d1.cpp (L1-L21) ⑤</div> <pre> 1 struct ggml_tensor * get_tensor_for(llama_load_tensor & lt, ggml_backend backend) { 2 struct ggml_tensor * tensor; 3 if (backend != GGML_BACKEND_CPU) { 4 ggml_set_no_alloc(ggml_ctx, true); 5 } 6 if (lt.ne.size() == 2) { 7 tensor = ggml_new_tensor_2d(ggml_ctx, lt.type, lt.ne.at(0), lt.ne.at(1)); 8 } else { 9 LLAMA_ASSERT(lt.ne.size() == 1); 10 tensor = ggml_new_tensor_1d(ggml_ctx, lt.type, lt.ne.at(0)); 11 } 12 ggml_set_name(tensor, lt.name.c_str()); 13 LLAMA_ASSERT(!lt.ggml_tensor == NULL); // if this fails, we called get_tensor twice on the same tensor </pre> </div>

The following seven items are displayed on the screen.

① Information about duplicate code

CT-ID, importance, number of similar statement lines, and similarity are displayed.

• CT-ID:

This is a unique ID assigned to a duplicate code group.

• Importance:

A higher importance score means greater technical debt. The score is calculated by a heuristic algorithm derived from the analysis of OSS results and user interviews. As it is still under development, its definition and the associated algorithm are subject to change.

• Number of similar statement lines:

Refers to the number of statements ($\hat{=}$ number of lines) in the duplicate code block.

• Similarity:

Indicates the percentage of all statements in the duplicate code block that are an exact match at the text level.

② COMMENT

This is an area to edit/display user comments about duplicate code.

③ Pair selection area for duplicate codes to be displayed on the code detail screens

This is the area for selecting the pair of duplicate codes to be displayed in ④ and ⑤. Select the radio button on the left/right corresponding to the pair of duplicate codes you wish to display.

④ Code view of the duplicate code on the left side

Displays the code for one of the duplicate code pairs.

The grayed-out area indicates the code that is an exact match to the code displayed on the right side. The red highlighted area indicates the code that differs from the code displayed on the right side.

⑤ **Code view of the duplicate code on the right side**

Displays the code for one of the duplicate code pairs.

The grayed-out area indicates the code that is an exact match to the code displayed on the right side. The green highlighted area indicates the code that differs from the code displayed on the right side.

⑥ **Mode change button(LEFT/RIGHT or OLD/NEW)**

In the LEFT/RIGHT code comparison mode, it displays the latest code of the duplicate code pair that you selected side by side.

In the OLD/NEW change history check mode, it displays the previous code of the duplicate code pair that you selected side by side. Note that it does not show the code when the inconsistency flag was set. In addition, if you switched branches before the analysis, the difference between the code from the current branch and the code from the previous branch is displayed as a change history. In order to avoid this miss comparison, it is necessary to manually run the analysis immediately after switching branches to initialize the history data.

⑦ **Archive button**

If you feel that the duplicate block does not need to be reviewed, click on this button. The duplicate data is not deleted, but will be displayed at the bottom of the list of remarks as low importance.

⑧ **Star**

If the star is on, the duplicate is displayed at the top of the analysis overview list. The duplicate with the “High risk” mark will automatically get this star mark.

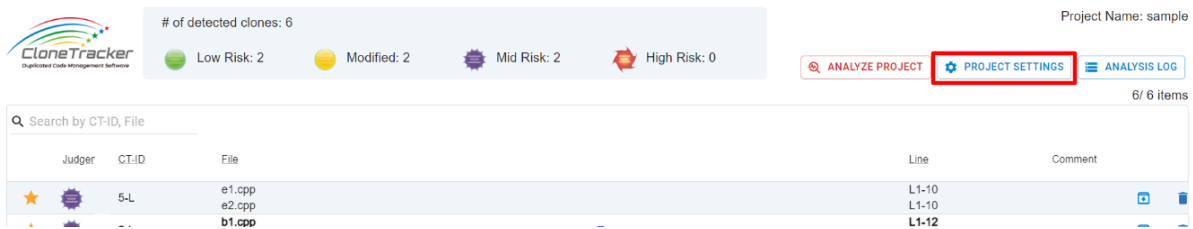
4. Settings for analysis execution

CloneTracker allows you to exclude specific files and folders from the analysis. Including files and folders that contain test code, sample code, or older versions of code in the analysis will result in the detection of more useless duplicate code. Please exclude files and folders from analysis properly to ensure the best analysis results.

There are two ways to exclude files/folders from the analysis.

4-1. In the Project Settings

Click “Project Settings”



This is the area to specify exclude files/folders.

Enter the file/folder name you want to exclude in the analysis. The separator is comma (,).

Example:

`test_code.cpp, readme.txt, log`

Spaces are not required.

You can also use wildcards. The format is the same as the ".gitignore" file, which is a file exclusion function in Git. Please refer to [the Git manual for details](#).

This exclusion setting is applied from the next analysis. Duplicate blocks in excluded files will disappear from the analysis result list after the next analysis.

Project Settings

Project Directory

The target path cannot be updated after project registration.

Languages

※ In a version, you can set Python/Java/JavaScript/PHP/VB.net/Rust as the target languages for analysis.

☒ C/C++ ☒ C#

α Ver.: ☐ Java ☐ Python ☐ Javascript ☐ PHP ☐ VB.NET ☐ Rust

Project Name ⓘ

Ignore Rules ⓘ

※ The paths that are set to be excluded from analysis by default are as follows.

test, *sample*, *proto*, *example*, *3rdparty*, dist, vendor, vendors, node_modules

Encoding

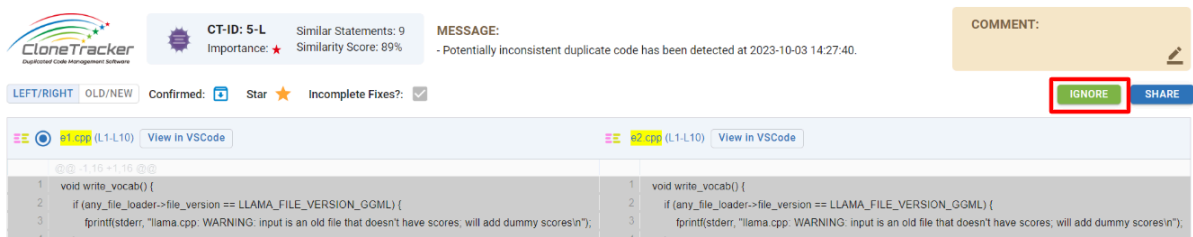
☒ UTF-8 ☐ Shift-JIS ☐ EUC-JP

License key

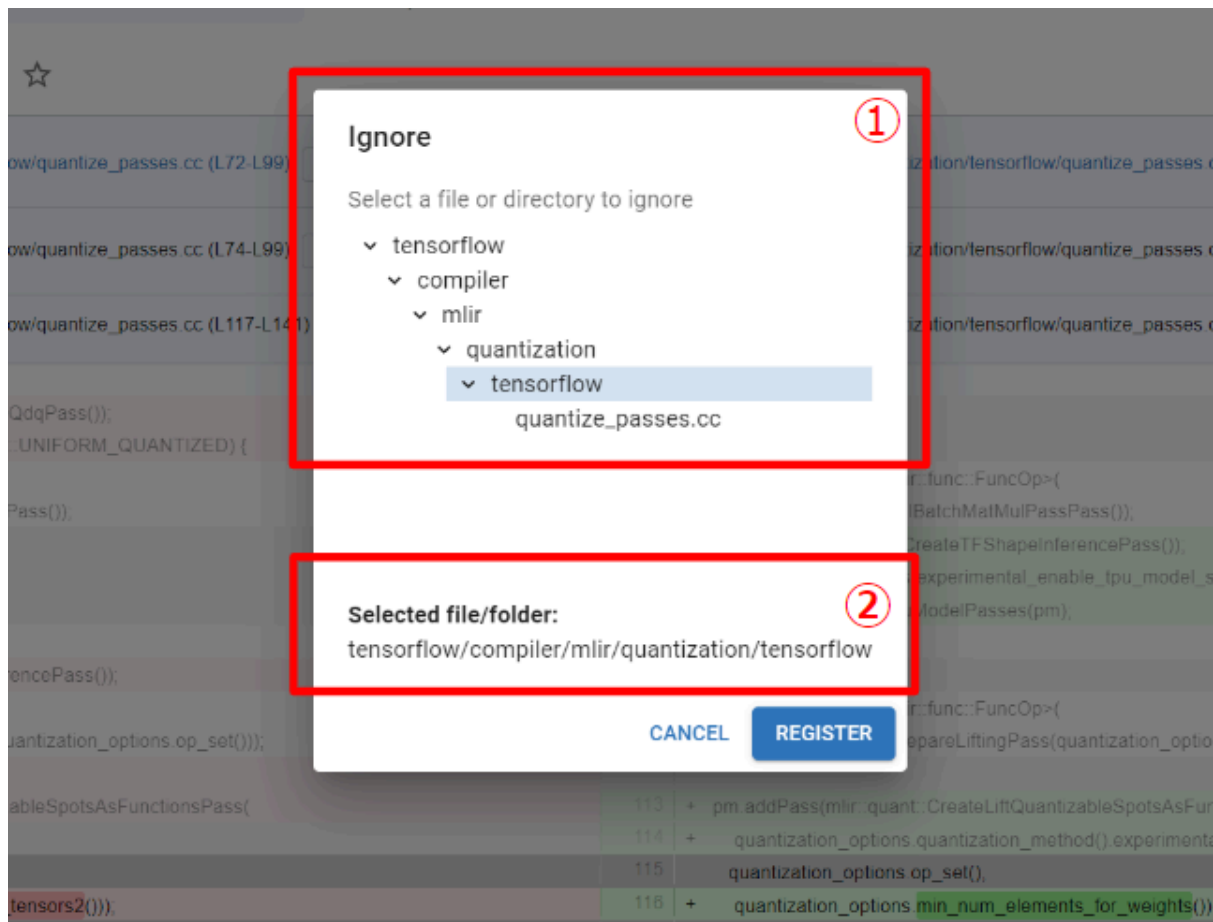
4-2. In the duplicate code view

You can exclude files in the duplicate code view.

Click the “IGNORE” button.



Register the files/folders in the modal window.



① Drill down to the file or folder to be excluded from analysis and select it.

② Confirm the selected file/folder, and click the “Register” button to complete.

This exclusion setting is applied from the next analysis. Duplicate blocks in excluded files will disappear from the analysis result list after the next analysis.

5. Analysis log

Analysis logs are available in the overview of analysis results.

CloneTracker
Duplication Code Management Software

of detected clones: 6

Low Risk: 2 Modified: 2 Mid Risk: 2 High Risk: 0

Project Name: sample

ANALYZE PROJECT PROJECT SETTINGS **ANALYSIS LOG**

Clones that have never been modified in the past

6/ 6 items

Search by CT-ID, File

Judger	CT-ID	File	Line	Comment
★	5-L	e1.cpp	L1-10	
★	5-L	e2.cpp	L1-10	
★	2-L	b1.cpp	L1-12	
★	2-L	b2.cpp	L1-12	
☆	4-L	d1.cpp	L1-21	
☆	4-L	d2.cpp	L1-21	
☆	1-L	a1.cc	L1-15	
☆	1-L	a2.cc	L1-15	
☆	6-L	f1.c	L1-29	
☆	6-L	f2.c	L1-29	
☆	3-L	c1.c	L1-12	
☆	3-L	c2.c	L1-12	

Rows per page: 100 1-6 of 6 < >

Click “ANALYSIS LOG” button.

CloneTracker
Duplication Code Management Software

Project Name: sample

← BACK TO DUPLICATE CODE LIST

2023/10/08 ①

★ 2023/10/08 (Sun) 15:09:09 Analyzed

2023/10/06

★ 2023/10/06 (Fri) 15:08:35 Analyzed ③

2023/10/06 (Fri) 15:07:53 Repository not modified ②

2023/10/03

2023/10/03 (Tue) 15:07:27 Repository not modified

★ 2023/10/03 (Tue) 15:06:28 Analyzed

VIEW LOGS DOWNLOAD

VIEW LOGS DOWNLOAD

VIEW LOGS DOWNLOAD

VIEW LOGS DOWNLOAD

VIEW LOGS DOWNLOAD

In this analysis log view, you can find following information.

- ① Analysis date
- ② Analysis time
- ③ Analysis results
- ④ Log view button
- ⑤ Log download button

Click “VIEW LOGS” button, and show the analysis log.



Analysis status: Analyzed
Analysis time: 2023-10-06 15:09:21

Project Name: sample

[← BACK TO HISTORY LIST](#)

[DOWNLOAD](#)

```
1: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][846] # ..... #
2: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][847] # Check Analyzing History ..... #
3: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][856] # ..... #
4: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][857] # Create Analyzing History ..... #
5: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][868] # ..... #
6: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][869] # Analyze Project : project 96 ..... #
7: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][871] # ..... #
8: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][872] # Collect Project Data ..... #
9: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][882] # ..... #
10: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][883] # Validate Project Data ..... #
11: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][892] # ..... #
12: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][893] # Put Feedback Files ..... #
13: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][927] # ..... #
14: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][928] # Build Analysis Args ..... #
15: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][942] # ..... #
16: [2023-10-08][15:09:09][INFO][clonetracker::project_analysis_handler][943] # Execute Analysis ..... #
17: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][426] start analyse_project ..... #
18: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][429] got app_handle ..... #
19: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][435] ..... #
20: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][436] SYSTEM LOG DIR :C:\Users\takuma\AppData\Roaming\com.clonetracker.dev\logs ..... #
21: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][437] ANALYSIS LOG DIR :C:\Users\takuma\AppData\Local\com.clonetracker.dev\projects\TestDate_6PairsOfCloneCode-1895161696979989622\20231008_150909_b7d351b-9e7d-42e6-ab33-371010d85933\log.txt ..... #
22: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][438] OUTPUT DIR :C:\Users\takuma\AppData\Local\com.clonetracker.dev\projects\TestDate_6PairsOfCloneCode-1895161696979989622\20231008_150909_b7d351b-9e7d-42e6-ab33-371010d85933\out ..... #
23: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][439] ..... #
24: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][441] try to get analysis_manager lock... ..... #
25: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][450] call analyse_project ..... #
26: [2023-10-08][15:09:09][INFO][clonetracker::analysis::analysis_manager][451] start analyse_project ..... #
```

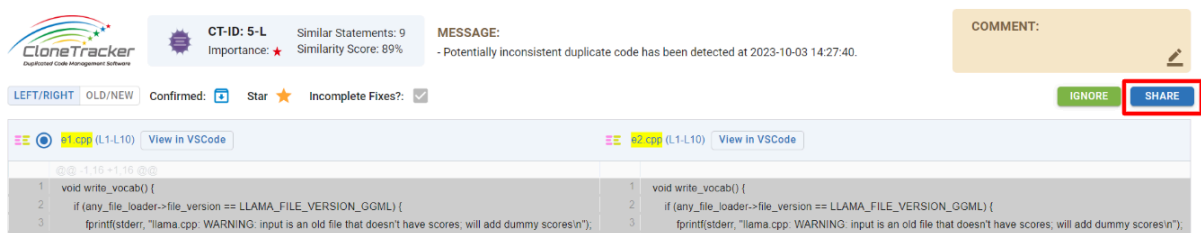
If an error occurs, please download the analysis log and contact customer support.

6. How to share the analysis result

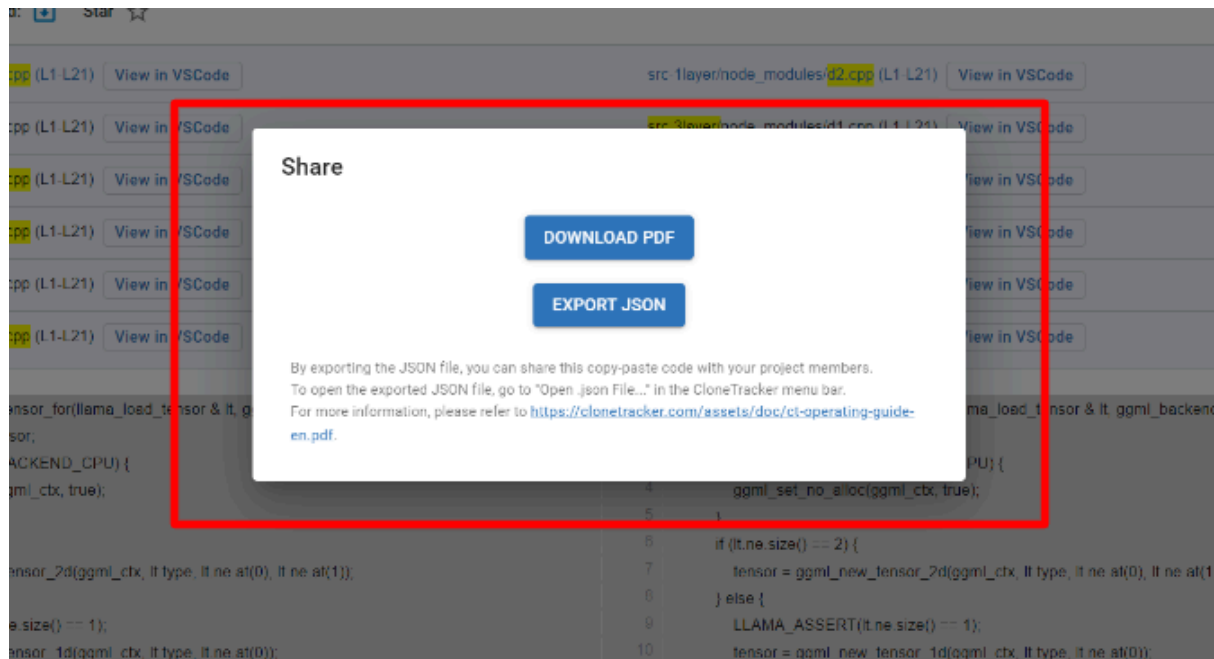
When you want to share the analysis result with development members, there are two methods: the first is to output the screenshot as a PDF file, and the second is to output a JSON file that restores the duplicate code view.

6-1. Export the analysis result

1. Show the duplicate code you want to share, click “SHARE” button.



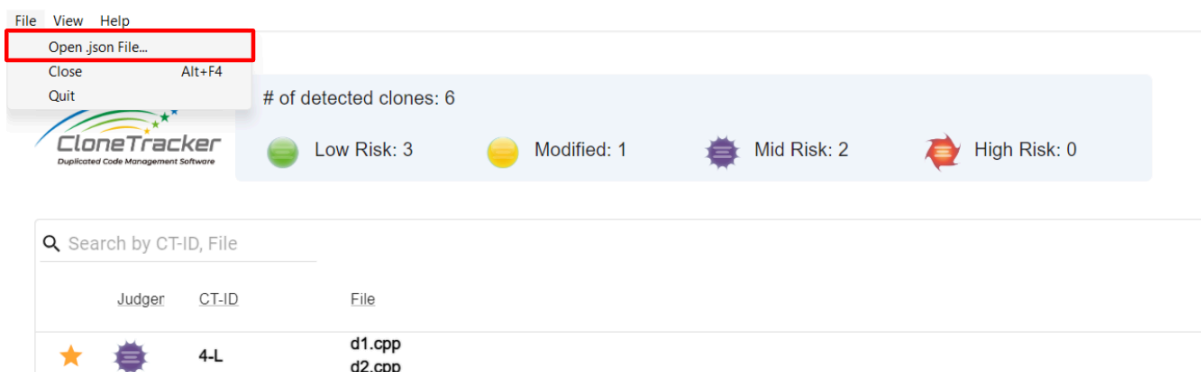
2. In the modal window, select a file format, and click “DOWNLOAD PDF” or “EXPORT JSON”.



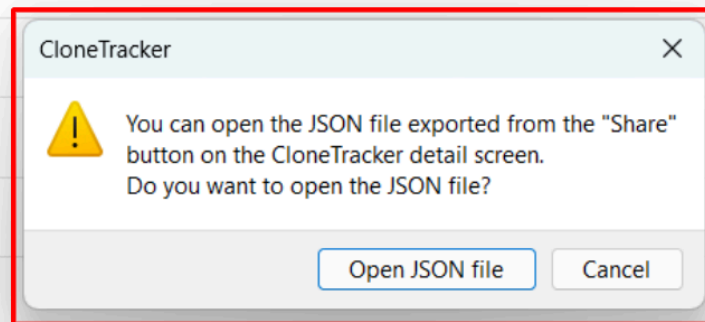
6-2. How to import the JSON file

To import the JSON file, click “File” -> “Open .json File...” on the menu bar in the overview of analysis results.

1. Click “File”



2. Click “Open .json File...”



A dialog screen will appear asking you to confirm opening the JSON file. Click the "Open JSON file" button.

A file explorer screen will be shown. Select the JSON file you want to import.