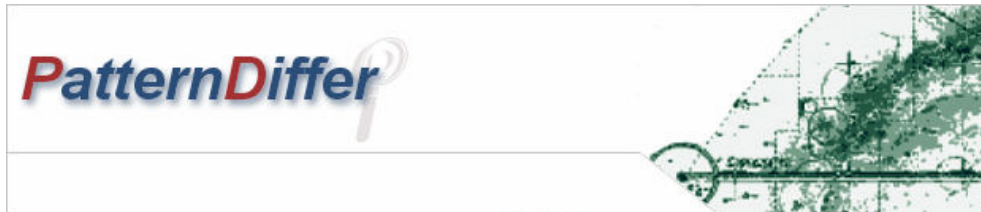


Manual PatternDiffer v1.1



Program type : Application for analysis of data structures
Operating system : Windows (all) / Linux / Unix / MacOS / Solaris
Requirements : **Java Virtual Machine v1.4.2**, PHP 5, MySQL 4.1
Author : Dipl.-Inf. (FH) Fenn Stefan

Contents

1 Description	1
2 Steps of the analysis	1
3 Options	2
4 Example of use	2
4.1 Simple text analysis	2
4.2 Analysis of different files	5
4.3 Search for Icon in exe file	6
4.4 Finding largest sequence in PI	6
5 Analysis algorithm	7
6 Additional notes	7

1 Description

Using this tool you have two possibilities to find in two different files equal data sequences also known as patterns. If PatternDiffer detects identic patterns they are displayed in a tabular or in a hexadecimal view. Additionally the data is show in a graphic. Moreover you have the possibility to save the results in XML file format in order to process the result later automatically by any other program. Since the data is analysed at a binary level its stucture is not relevance. Simplified you can consider PatternDiffer to analyse the biggest coherent parts in two files.

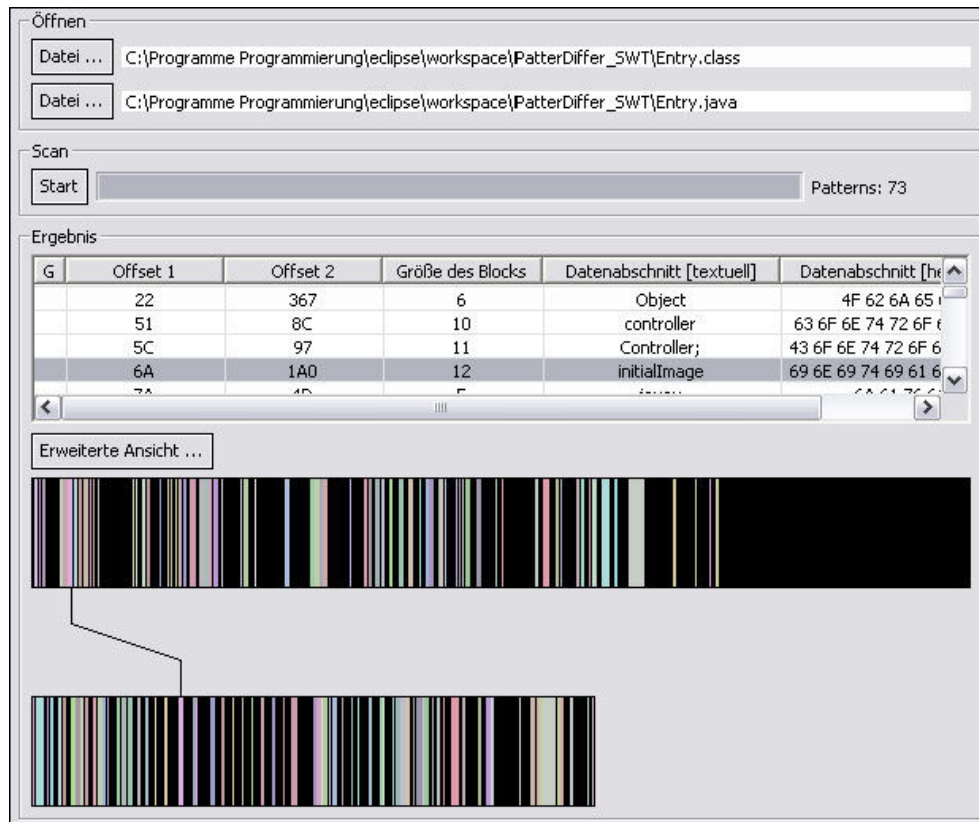


Figure 1: Screenshot PatternDiffer

Additional information in chapter "Handling" .

2 Steps of the analysis

After you selected the files to be analysed, the analysis starts by pressing the "start" -buttons. During the analysis process a fall back of the progress bar can happen. Don't worry about that because this depends on the used analysis algorithm in PatternDiffer

The action of analysis can be canceled anytime but the result PatternDiffern found will still stay and can be exported to xml too.

After the analysis is finished the found pattern blocks will be shown in a table. Below you can view the

result in a visual view to be able to find important places quickly. If you need more detailed information concerning a special pattern found you can get it by simply pressing the "Extended Analysis" -button.

If the retrieved results are designated for automatic processing with another program for example you can export the results through the export function in the menu.

3 Options

In menu "Options" and in submenu "Preferences" you can choose the analysis mode

There are two different modi called "Analysis inside a file" and "Analysis in different files" . If you start the analysis inside a file so PatternDiffer will skip sequences at the same file position. This is for detection of identic areas in a file. For example you can find in the pattern "abbbabbbcdddccabbccddaaadb" the longest equal sequence. In this case it's "abbbab" . You can follow this example in chapter **Example of application/Simple text analysis**. It doesn't matter what kind of data you input to pattern differ in form of files you simply analyse this data for the existence of equal structures and patterns

Using the "Analysis in different files" those patterns are detected which can be found in both source files and spanning the biggest possible interval.

Lets assume that you have two files with unknown file formats you can use PatternDiffer to retrieve Information about the similarities inside the format itself. In future there is no need to know the file format you will simple analyse it with PatternDiffer. You can use it to detect file headers, special areas, attributes, signatures and much more...

Look at chapter **Example of applications/Reading a word signature** You will find there detaild description of how a data analysis can take place.

If you don't want binary '

0' to be detected as part of a pattern you can simple enable the option "Ignore binary 0" . This is especially necessary for files containing large amounts of binary '0'-s.

If you set the option to "shortest sequence" only the smallest sequences are found. The main advantage of this option is that is very fast.

Attention: If you use the analysis on huge files (1MB or above) it can take much time. This is simply a side effect of the complexity of the search algorithm. The described preferences should be used to fasten the search engine.

4 Example of use

4.1 Simple text analysis

Let's assume that we do a simple text analysis which should show the internal similarities inside a text.

- Start the PatternDiffer.
- Press the "File ..." -button and the file selection is opening. Search the file **examples/text01/a.txt**.

- Press the second "File ..." -button and choose again **examples/text01/a.txt**. This file has following test sequence in it "abbbabbccdddccabbccddaaadb".
- Press the "Start" -button and the search begins.
- As result you get a pattern which is spanning over the whole file. This means the the largest sequence in the both files is the file itself. The result is displayes in the graph below.

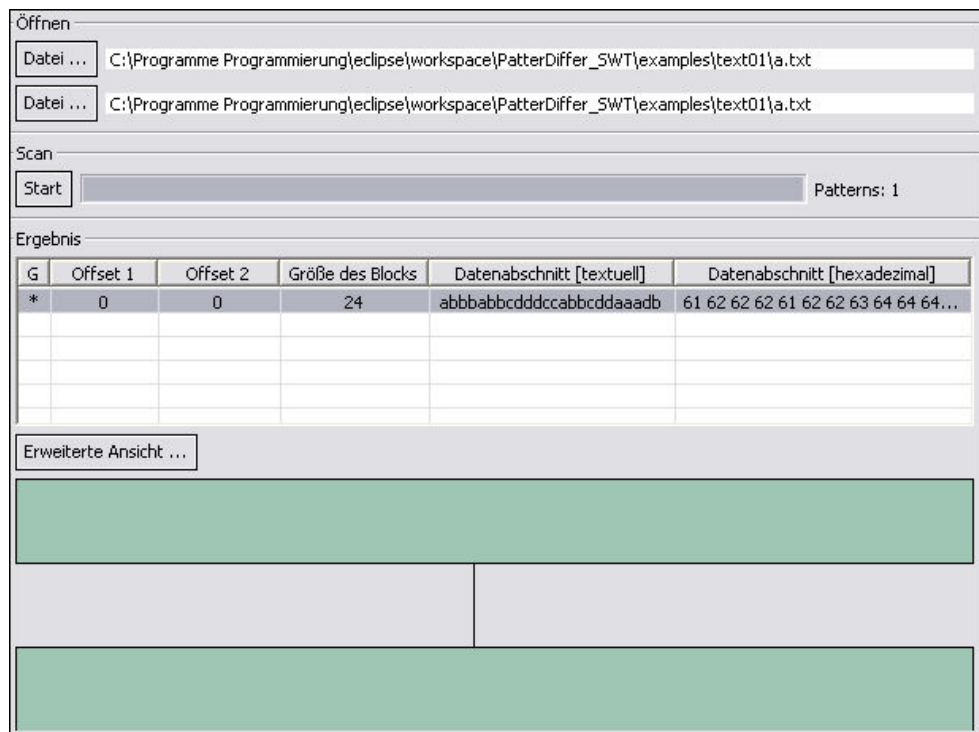


Figure 2: Screenshot PatternDiffer - the whole text of the file is the result

- But we wanted to find identic sequences in the file itself, this means that the solution "whole file" is not valid for this approach. So we have to alter the options in "Options/Preferences/Analysis mode/inside a file" . If you start the search again by pressing STRG + S two sequences are found. Since we chosen "search inside a file"the two sequences are equal. So the largest sequence inside the file is abbcdd. You can see the result below.

- Normally we would say this is a quite good result but maybe there are other results too ? Select "Options/Preferences/Analysis demand/Userdefined" **1**. This activates the option that each hex value is examined. Normally you search for sequences which are longer than one byte. If you start the search again you can see each position changing and the before found sequence.

Note: Even a position change is possible. PatternDiffer searches for variants with biggest connected sequences. The shorter the sequences are the more probable the this sequences are equal.

You can see the result in the graph below.

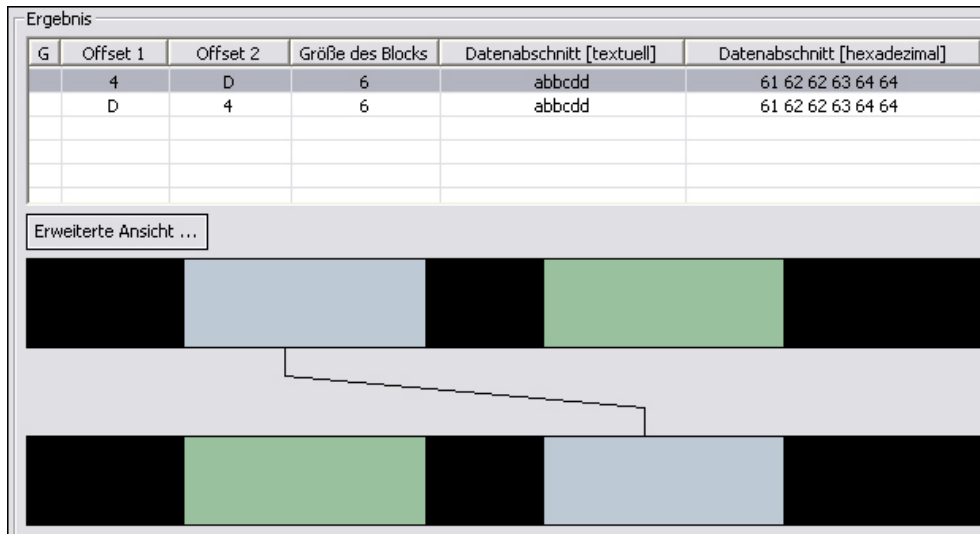


Figure 3: Search inside a file, sequence abbcdd found

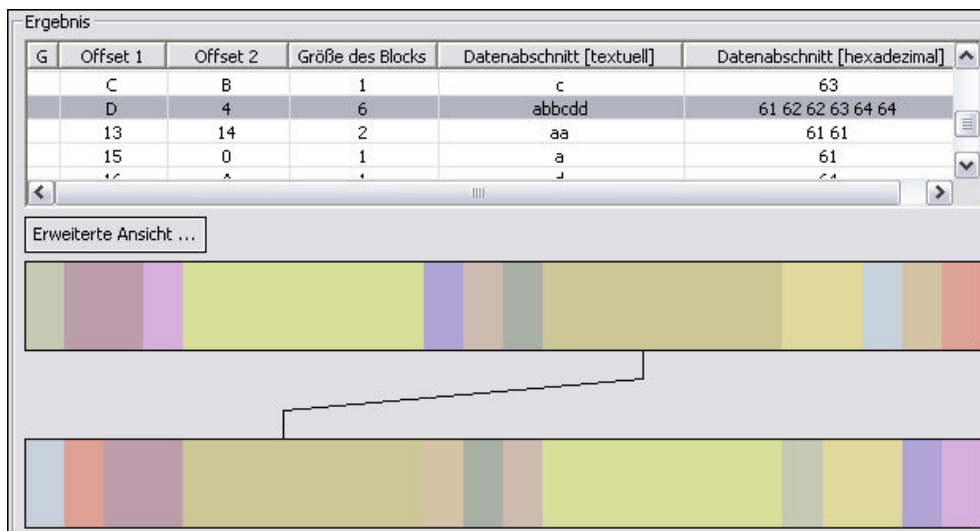


Figure 4: Result of a search, which differs each byte

Just experiment with the analyse by changing the text file or just select other files.

4.2 Text analysis in two different files

Now we examine which text parts can be found in the first,second, third or fourth strophe in "Das Lied von der Glocke" by Friedrich Schiller

- Start the search again.
- Choose "File/Open..." and select the both files **examples/text02/a.txt** and **examples/text02/b.txt**. In the first file you can see the first and the second strophe of the poem and in the second file the third and the fourth strophes are included. Click on "Open" .
- Press "Start" -button and the search begins again.
- 41 patterns are found. This means that we have about 41 sequences with a minimal length of four bytes which are include in the first file and in the second file too. If you search for the largest sequence you will find a sequence with a length of 10 bytes which is equal to "Menschen" .
- If you select the sequence you will be shown a graphical view of the position of the sequence in the file. If you press "Extended view ..." you will see in which coherence the word "Menschen" was shown. Look at the window below.



Figure 5: Extended view of the sequence

Hint: If you search inside big files the search progress can last long. In the directory "examples/statistic" you can use the both files "randomNumber01.txt" and "randomNumber02.txt" to analyse and check the coherence between different **minimal sequence length**. If you cancel the search all sequences until now are shown.

4.3 Search for Icon in exe file

We just want to analyse how and where an icon image can be found in a compiled file. Unlike other tools PatternDiffer is able to find fragments of sequences. This is useful because in much files header it is present but in the second file it's changed or deleted. Aim of this example is to find out whether the hex-code of the image is present in the exe file in its original sequence.

- Start the PatternDiffer.
- Choose "File/Open..." and select the both files **examples/exe_icon/HelloWorld.ico** and **examples/exe_icon/HelloWorld.exe**. The first file is the icon file itself and the second one is the executable with the compiled image in it. Now press "Open".
- Since we are only interested in the image information and we assume that this informations are connected we alter in "Optionen/Einstellungen" the smallest sequence to 100 byte. This fasten the search a lot. Now press the "Start" -button and search starts.
- A pattern with a size of 6.786 bytes is found. The icon file itself is about 6.838 bytes large. This show that almost every byte of the original file was compiled in the binary. If you search for smaller sequences you will be able to find also some sparse header informations but this is not interesting for us. **Offset 1** means that the search sequence starts at an offset of 0x34 (decimal 52) and **Offset 2** means that the sequence in the second file starts at 0xCB02 (decimal 51.970). In the view below you see the result.

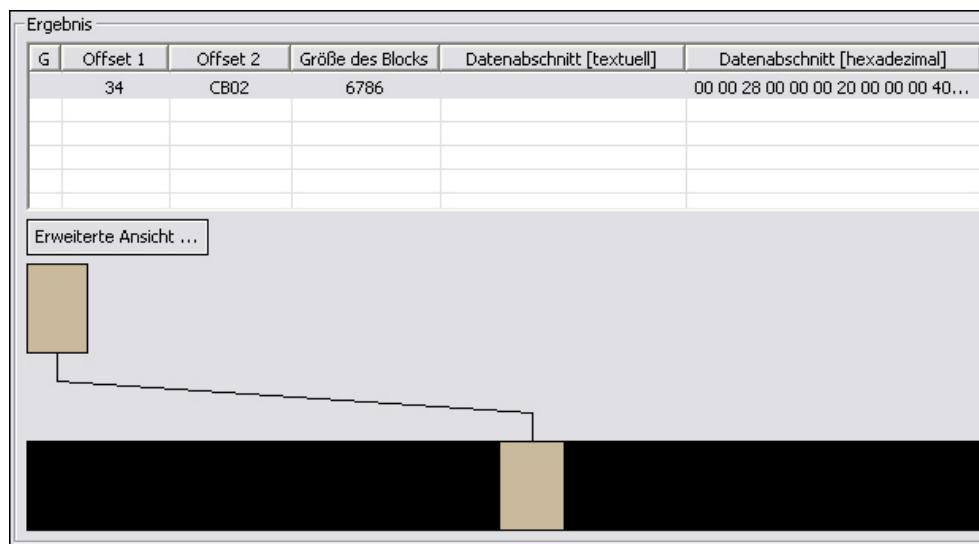


Figure 6: Searching for an icon image in an executable file

4.4 Finding largest sequence in circle π

In this lection we will show you how to find the largest sequence in π which exist more than one time in the decimal place with an accuracy of 100.000 digits. Also we want to get familiar with the export possibilities of PatternDiffer.

- Start PatternDiffer.
- Choose the file "examples/statistics/pi.txt" in both directory text fields. Set the minimal sequence length to 8 and choose analys mode "inside a file" . Now start the sequence process. The search will take about 5-20 minutes.
- If you take a look at the search result you will see that the longest sequence is in π with 9 digits. Now we will try to save the results in order to process it later by any other program.
- First we export the result but pressing "File/Export..." to the file "examples/statistics/pi_result.xml" . You will be prompted to override the old XML-File. The result is now save to XML-File-Format.
- Now you can transform all possible data exploits form this xml-file. In this file all information about the found sequences and analysis settings can be found. Now we want to create a simple statistic from sequeze length and occurence
Start the batch-file makeStatistik.bat. The file "pi_result.xml" will now bei transformed with the transformation "statistik.xslt" (XSLT is a transformation-language). The result will now be written to the file "pi_statistik.xml" . Open the file and you see that sequences of length eight 80 times and sequences of length nine 6 times can be found. Shorter sequences have been ignored in the analysis.

5 Details concerning the analysis algorithm

The algorithm searches in both data structure for equal sequences. The number of different Sequence possibilities is rather large. The runtime of the analysis is $O(n) = n^3$. Therefore the first optimal solution will be shown. Other possible solutions are neglected. Restriction like "Ignore binary ' 0'" or "smallest sequence length" fasten the search progress. So we advise to analyse little parts first.

First both files are completly read into memory after that they are indexed. The aquirement of all possible sequence intervals takes place by complete recombination of all constructable possibilities. Now depending on the selected options the best fitting sequences are selected.

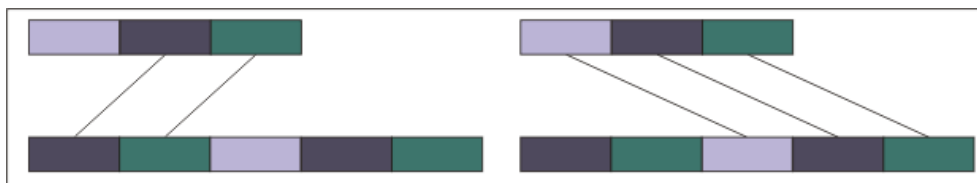


Figure 7: To the left short sequence, to the right long sequence. PatternDiffer choose the right solution.

6 Additional notes

If you find bugs or if you have ideas of how to improve PatternDiffer we would be glad to get a mail from you. [Email](#). or for other help [Support-Board](#)

Your metholution team

