BinDiffFilter

1. Requirments

IDA Pro with installed PySide. The best option is to have IDA Pro 6.6 because PySide is installed by default.

2. How does it work ?

BinDiffFilter finds BinDiff “Matched Functions” window which is QTableView and make further operation on it and it’s model and proxymodel.

3. Functionality

a. addFilterFunction

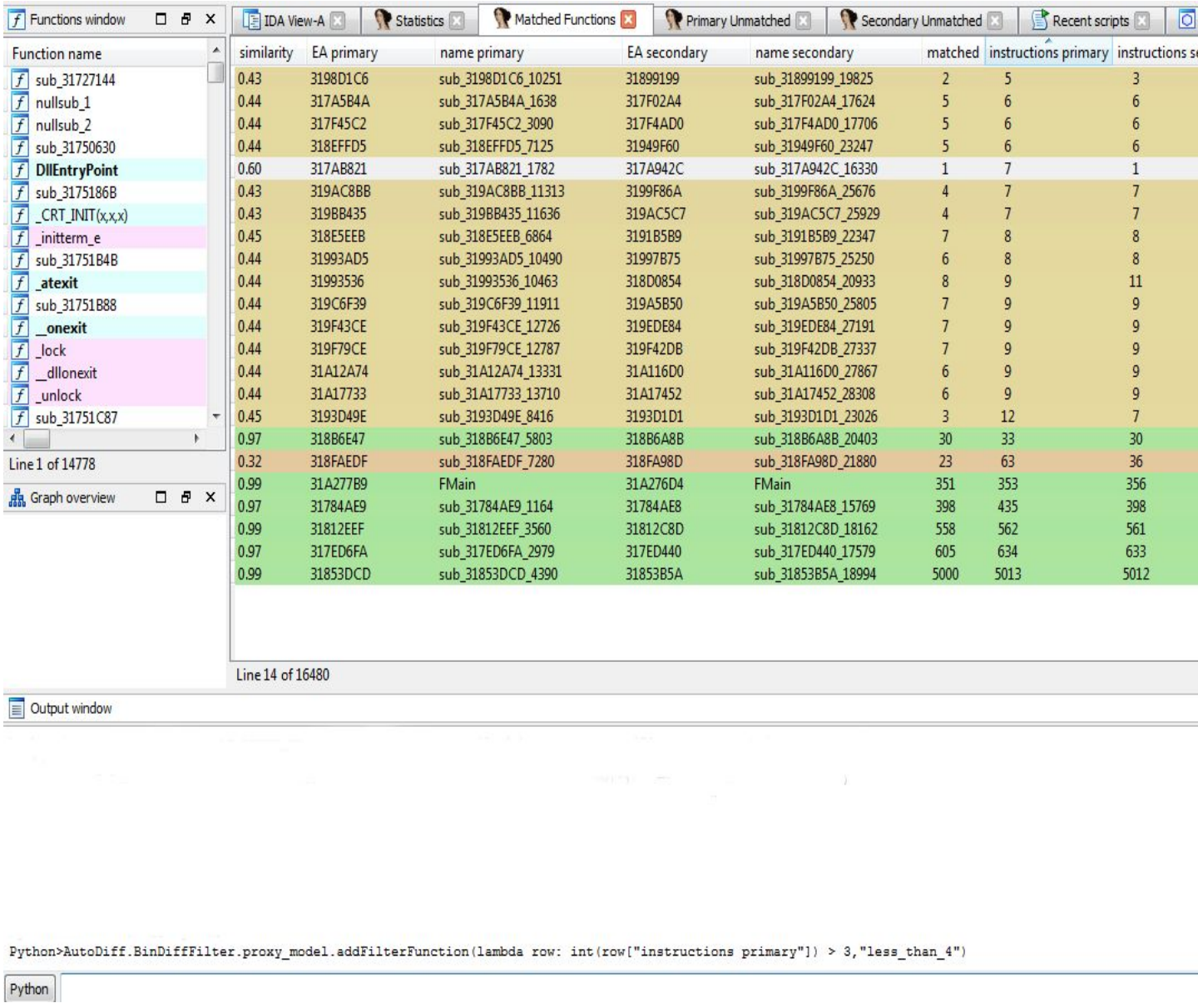
Add filter method gives u possibility to add custom lambda function which as parameter receives QTableView row in python form which is dictionary.
Example of usage:
Let’s create filter function which will hide all functions with instructions number less than 4.
Before filtering

similarity	EA primary	name primary	EA secondary	name secondary	matched	instructions primary	instructions secondary
0.43	318C00D9	sub_318C00D9_5999	318D237A	sub_318D237A_20957	1	2	2
0.43	3191EF7C	sub_3191EF7C_7793	3191EC1A	sub_3191EC1A_22400	1	2	2
0.43	3191F61E	sub_3191F61E_7823	31924D1F	sub_31924D1F_22562	1	2	2
0.43	31925A34	sub_31925A34_7970	31930345	sub_31930345_22840	1	2	2
0.43	3194AC5B	sub_3194AC5B_8657	3195351E	sub_3195351E_23425	1	2	2
0.43	319977E3	sub_319977E3_10598	3199786D	sub_3199786D_25231	1	2	2
0.43	3199C506	sub_3199C506_10788	3199CD15	sub_3199CD15_25467	1	2	2
0.43	31A0D231	sub_31A0D231_13188	31A0D306	sub_31A0D306_27809	1	2	2
0.43	31A2CAC4	sub_31A2CAC4_14205	31A2CA9F	sub_31A2CA9F_28816	1	2	2
0.43	3198D1C6	sub_3198D1C6_10251	31899199	sub_31899199_19825	2	5	3
0.44	317A5B4A	sub_317A5B4A_1638	317F02A4	sub_317F02A4_17624	5	6	6
0.44	317F45C2	sub_317F45C2_3090	317F4AD0	sub_317F4AD0_17706	5	6	6
0.44	318EFFD5	sub_318EFFD5_7125	31949F60	sub_31949F60_23247	5	6	6
0.60	317AB821	sub_317AB821_1782	317A942C	sub_317A942C_16330	1	7	1
0.43	319AC8BB	sub_319AC8BB_11313	3199F86A	sub_3199F86A_25676	4	7	7
0.43	319BB435	sub_319BB435_11636	319AC5C7	sub_319AC5C7_25929	4	7	7
0.45	318E5EEB	sub_318E5EEB_6864	3191B5B9	sub_3191B5B9_22347	7	8	8
0.44	31993AD5	sub_31993AD5_10490	31997B75	sub_31997B75_25250	6	8	8
0.44	31993536	sub_31993536_10463	318D0854	sub_318D0854_20933	8	9	11
0.44	319C6F39	sub_319C6F39_11911	319A5B50	sub_319A5B50_25805	7	9	9
0.44	319F43CE	sub_319F43CE_12726	319EDE84	sub_319EDE84_27191	7	9	9
0.44	319F79CE	sub_319F79CE_12787	319F42DB	sub_319F42DB_27337	7	9	9
0.44	31A12A74	sub_31A12A74_13331	31A116D0	sub_31A116D0_27867	6	9	9
0.44	31A17733	sub_31A17733_13710	31A17452	sub_31A17452_28308	6	9	9
0.45	3193D49E	sub_3193D49E_8416	3193D1D1	sub_3193D1D1_23026	3	12	7
0.97	318B6E47	sub_318B6E47_5803	318B6A8B	sub_318B6A8B_20403	30	33	30

Line 14 of 16480

After:

AutoDiff.BinDiffFilter.proxy_model.addFilterFunction(lambda row: int(row[“instructions primary”])) > 3,”less_than_4”)



notice that second parameter is optional name of filter. Is good to add name which says something specific about filter , that info can be usefull during filter removal. If non name is passed filter gets global name which is filter counter.

b. Display in use filters

Python>"\n".join(AutoDiff.BinDiffFilter.proxy_model.getFilterFunctions().keys())

hideFunction
less_than_4
matched

c. Remove filter

Python>AutoDiff.BinDiffFilter.proxy_model.removeFilterFunction("less_than_4")

d. hideFunction

You can in easy way hide particular function. I found this functionality especially usefull during work with patch analysis. When u analized function and decided that it's not interesting u can easly hide it. This filter is also heavly used by AutoDiff to hide functions pointed by Sanitizer and Rematcher.

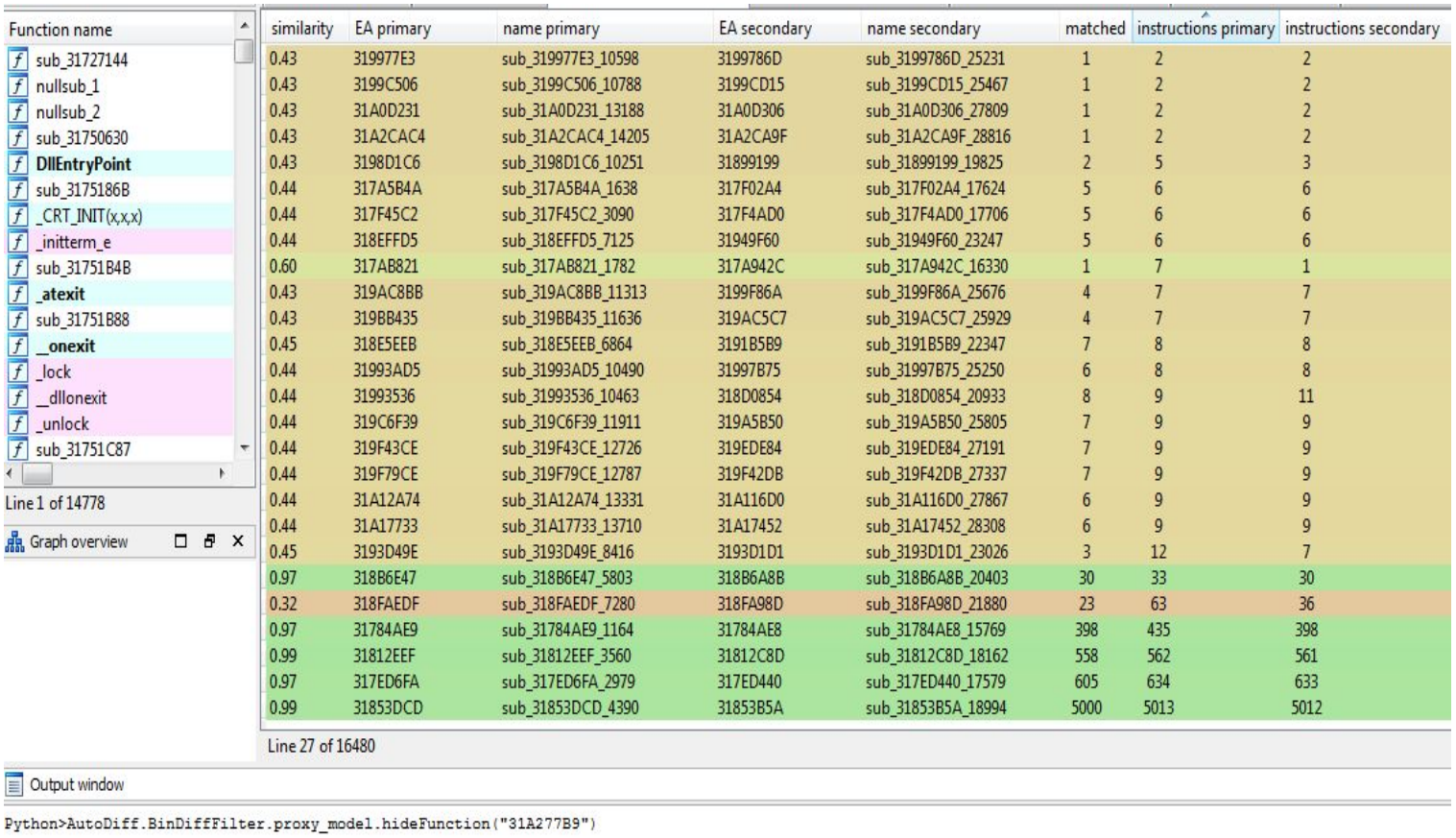
Example:

Let's we hide function with name **Fmain**:

Before:

similarity	EA primary	name primary	EA secondary	name secondary	matched	instructions primary	instructions secondary
0.43	3199C506	sub_3199C506_10788	3199CD15	sub_3199CD15_25467	1	2	2
0.43	31A0D231	sub_31A0D231_13188	31A0D306	sub_31A0D306_27809	1	2	2
0.43	31A2CAC4	sub_31A2CAC4_14205	31A2CA9F	sub_31A2CA9F_28816	1	2	2
0.43	3198D1C6	sub_3198D1C6_10251	31899199	sub_31899199_19825	2	5	3
0.44	317A5B4A	sub_317A5B4A_1638	317F02A4	sub_317F02A4_17624	5	6	6
0.44	317F45C2	sub_317F45C2_3090	317F4AD0	sub_317F4AD0_17706	5	6	6
0.44	318EFFD5	sub_318EFFD5_7125	31949F60	sub_31949F60_23247	5	6	6
0.60	317AB821	sub_317AB821_1782	317A942C	sub_317A942C_16330	1	7	1
0.43	319AC8BB	sub_319AC8BB_11313	3199F86A	sub_3199F86A_25676	4	7	7
0.43	319BB435	sub_319BB435_11636	319AC5C7	sub_319AC5C7_25929	4	7	7
0.45	318E5EEB	sub_318E5EEB_6864	3191B5B9	sub_3191B5B9_22347	7	8	8
0.44	31993AD5	sub_31993AD5_10490	31997B75	sub_31997B75_25250	6	8	8
0.44	31993536	sub_31993536_10463	318D0854	sub_318D0854_20933	8	9	11
0.44	319C6F39	sub_319C6F39_11911	319A5B50	sub_319A5B50_25805	7	9	9
0.44	319F43CE	sub_319F43CE_12726	319EDE84	sub_319EDE84_27191	7	9	9
0.44	319F79CE	sub_319F79CE_12787	319F42DB	sub_319F42DB_27337	7	9	9
0.44	31A12A74	sub_31A12A74_13331	31A116D0	sub_31A116D0_27867	6	9	9
0.44	31A17733	sub_31A17733_13710	31A17452	sub_31A17452_28308	6	9	9
0.45	3193D49E	sub_3193D49E_8416	3193D1D1	sub_3193D1D1_23026	3	12	7
0.97	318B6E47	sub_318B6E47_5803	318B6A8B	sub_318B6A8B_20403	30	33	30
0.32	318FAEDF	sub_318FAEDF_7280	318FA98D	sub_318FA98D_21880	23	63	36
0.99	31A277B9	FMain	31A276D4	FMain	351	353	356
0.97	31784AE9	sub_31784AE9_1164	31784AE8	sub_31784AE8_15769	398	435	398
0.99	31812EEF	sub_31812EEF_3560	31812C8D	sub_31812C8D_18162	558	562	561
0.97	317ED6FA	sub_317ED6FA_2979	317ED440	sub_317ED440_17579	605	634	633
0.99	31853DCD	sub_31853DCD_4390	31853B5A	sub_31853B5A_18994	5000	5013	5012

After:



You can bring back hidden function by calling showFunction method.
Python>AutoDiff.BinDiffFilter.proxy_model.showFunction(“31A277B9”)

e. Hide matched functions

During patch analysis we are not interested in matched functions with similarity 1.0. BinDiffFilter gives u possibility to filter out all these functions in an easy way:
Example:
Python>AutoDiff.BinDiffFilter.proxy_model.hideMatchedFunctions()

this filter can be also removed and it’s name is “matched”:
Python>AutoDiff.BinDiffFilter.proxy_model.removeFilterFunction("matched")
the same effect:
Python>AutoDiff.BinDiffFilter.proxy_model.showMatchedFunctions()

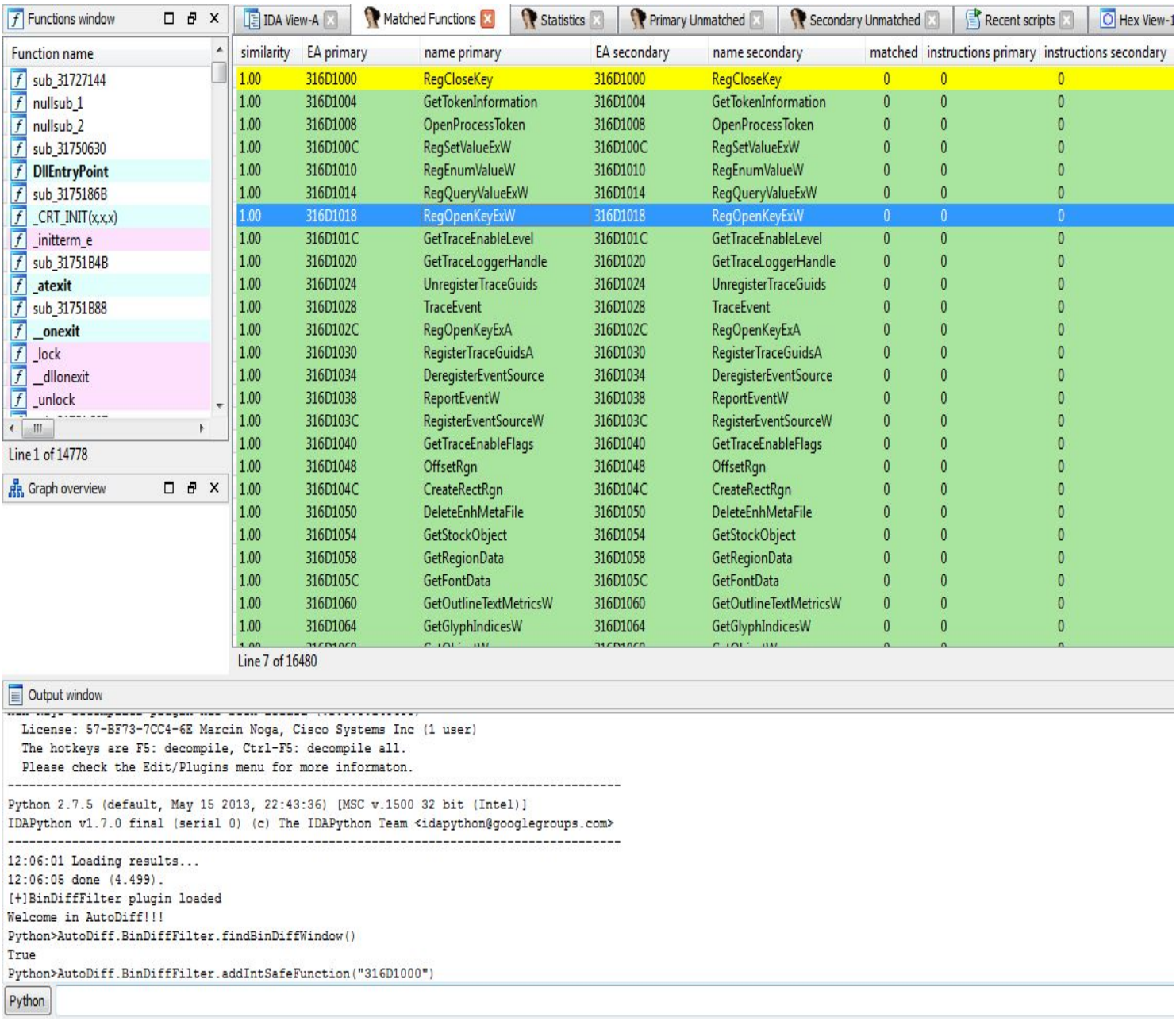
f. Highlight row

Option used with AutoDiff results from SignificantFunctions module which points “positives functions”, what means functions which should be in some how marked instead of hidden. That’s way BinDiffFilter gives u possibility to highlight specific row based on “EA primary” value.

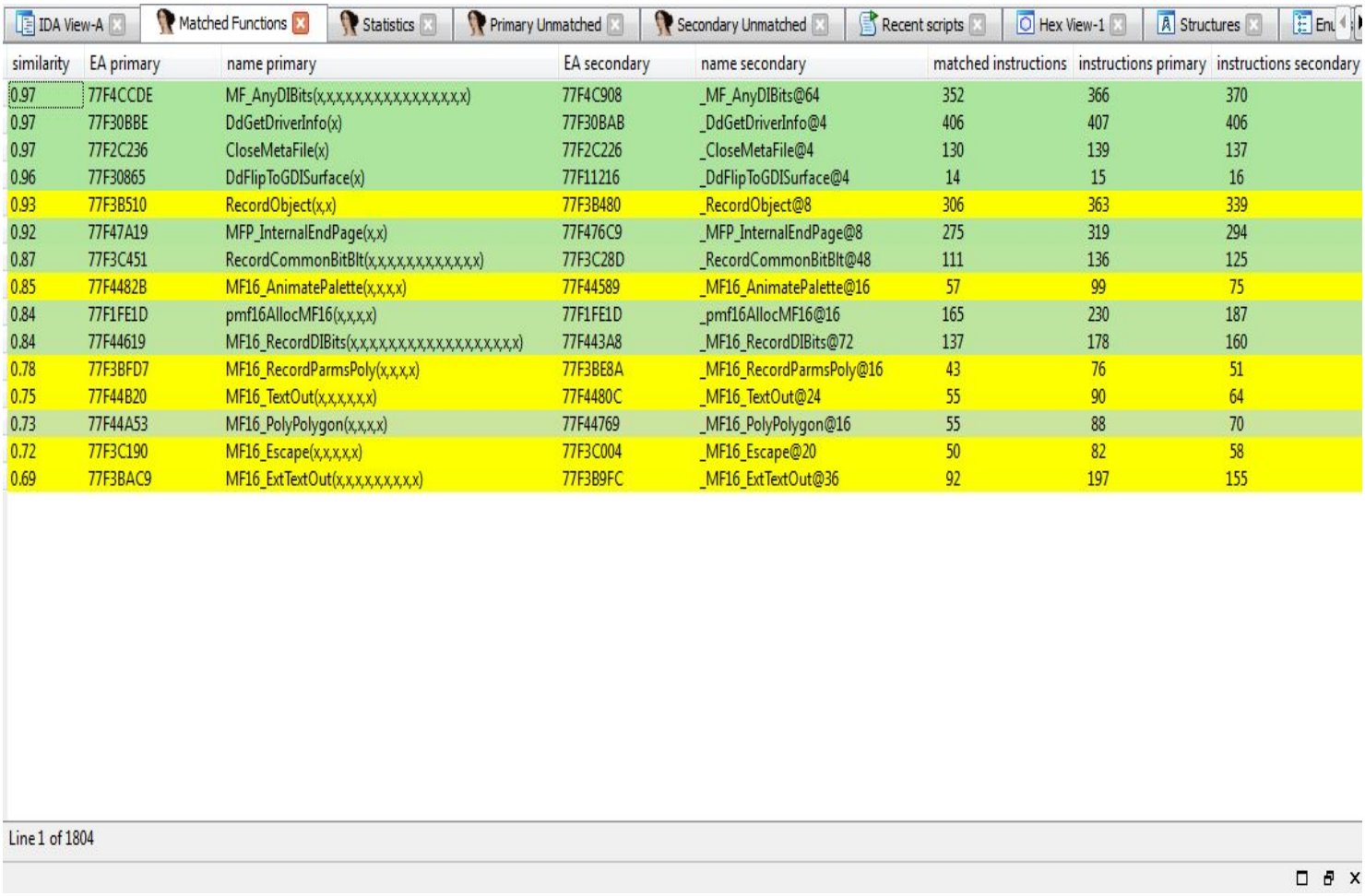
Example:
Let’s we highlight function row : **RegCloseKey**
Before:

similarity	EA primary	name primary	EA secondary	name secondary	matched	instructions primary	instructions secondary
1.00	316D1000	RegCloseKey	316D1000	RegCloseKey	0	0	0
1.00	316D1004	GetTokenInformation	316D1004	GetTokenInformation	0	0	0
1.00	316D1008	OpenProcessToken	316D1008	OpenProcessToken	0	0	0
1.00	316D100C	RegSetValueExW	316D100C	RegSetValueExW	0	0	0
1.00	316D1010	RegEnumValueW	316D1010	RegEnumValueW	0	0	0
1.00	316D1014	RegQueryValueExW	316D1014	RegQueryValueExW	0	0	0
1.00	316D1018	RegOpenKeyExW	316D1018	RegOpenKeyExW	0	0	0
1.00	316D101C	GetTraceEnableLevel	316D101C	GetTraceEnableLevel	0	0	0
1.00	316D1020	GetTraceLoggerHandle	316D1020	GetTraceLoggerHandle	0	0	0
1.00	316D1024	UnregisterTraceGuids	316D1024	UnregisterTraceGuids	0	0	0
1.00	316D1028	TraceEvent	316D1028	TraceEvent	0	0	0
1.00	316D102C	RegOpenKeyExA	316D102C	RegOpenKeyExA	0	0	0
1.00	316D1030	RegisterTraceGuidsA	316D1030	RegisterTraceGuidsA	0	0	0
1.00	316D1034	DeregisterEventSource	316D1034	DeregisterEventSource	0	0	0
1.00	316D1038	ReportEventW	316D1038	ReportEventW	0	0	0
1.00	316D103C	RegisterEventSourceW	316D103C	RegisterEventSourceW	0	0	0
1.00	316D1040	GetTraceEnableFlags	316D1040	GetTraceEnableFlags	0	0	0
1.00	316D1048	OffsetRgn	316D1048	OffsetRgn	0	0	0
1.00	316D104C	CreateRectRgn	316D104C	CreateRectRgn	0	0	0
1.00	316D1050	DeleteEnhMetaFile	316D1050	DeleteEnhMetaFile	0	0	0
1.00	316D1054	GetStockObject	316D1054	GetStockObject	0	0	0
1.00	316D1058	GetRegionData	316D1058	GetRegionData	0	0	0
1.00	316D105C	GetFontData	316D105C	GetFontData	0	0	0
1.00	316D1060	GetOutlineTextMetricsW	316D1060	GetOutlineTextMetricsW	0	0	0
1.00	316D1064	GetGlyphIndicesW	316D1064	GetGlyphIndicesW	0	0	0

After:



U can achieve similar effect automaticly when u use AutoDiff “Summarize 2.0” function.
Let’s we see effect for GDI.dll



IDA View-A | Matched Functions | Statistics | Primary Unmatched | Secondary Unmatched | Recent scripts | Hex View-1 | Structures | Enu

similarity	EA primary	name primary	EA secondary	name secondary	matched instructions	instructions primary	instructions secondary
0.97	77F4CCDE	MF_AnyDIBits(x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x)	77F4C908	_MF_AnyDIBits@64	352	366	370
0.97	77F30BBE	DdGetDriverInfo(x)	77F30BAB	_DdGetDriverInfo@4	406	407	406
0.97	77F2C236	CloseMetaFile(x)	77F2C226	_CloseMetaFile@4	130	139	137
0.96	77F30865	DdFlipToGDISurface(x)	77F11216	_DdFlipToGDISurface@4	14	15	16
0.93	77F3B510	RecordObject(x,x)	77F3B480	_RecordObject@8	306	363	339
0.92	77F47A19	MFP_InternalEndPage(x,x)	77F476C9	_MFP_InternalEndPage@8	275	319	294
0.87	77F3C451	RecordCommonBitBlt(x,x,x,x,x,x,x,x,x,x,x,x)	77F3C28D	_RecordCommonBitBlt@48	111	136	125
0.85	77F4482B	MF16_AnimatePalette(x,x,x,x)	77F44589	_MF16_AnimatePalette@16	57	99	75
0.84	77F1FE1D	pmf16AllocMF16(x,x,x,x)	77F1FE1D	_pmf16AllocMF16@16	165	230	187
0.84	77F44619	MF16_RecordDIBits(x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x)	77F443A8	_MF16_RecordDIBits@72	137	178	160
0.78	77F3BFD7	MF16_RecordParmsPoly(x,x,x,x)	77F3BE8A	_MF16_RecordParmsPoly@16	43	76	51
0.75	77F44B20	MF16_TextOut(x,x,x,x,x,x)	77F4480C	_MF16_TextOut@24	55	90	64
0.73	77F44A53	MF16_PolyPolygon(x,x,x,x)	77F44769	_MF16_PolyPolygon@16	55	88	70
0.72	77F3C190	MF16_Escape(x,x,x,x,x)	77F3C004	_MF16_Escape@20	50	82	58
0.69	77F3BAC9	MF16_ExtTextOut(x,x,x,x,x,x,x,x,x)	77F3B9FC	_MF16_ExtTextOut@36	92	197	155

Line 1 of 1804

g. Hide Column(s)

BinDiff presents a lot of columns with data which not always can be usefull and bring not necesary mess in result window. Now u can easly hide particular column calling :
Python>AutoDiff.BinDiffFilter.hideColumn("similarity")
reverse action:
Python>AutoDiff.BinDiffFilter.showColumn("similarity")

When u choose “Summarize 2.0” AutoDiff option , automaticly call method:

AutoDiff.BinDiffFilter.hideSomeStandardColumns()
these standards collumns are

```
self.__uselessStandardColumns = [
                                "confidence",
                                "change",
                                "comments ported",
                                "algorithm",
                                "matched basicblocks",
                                "basicblocks primary",
                                "basicblocks secondary",
                                "matched edges",
                                "edges primary",
                                "edges secondary"
                                ]
```

Effect with hidden above defined columns u could observe on previous screenshots.
Reverse method is :
showSomeStandardColumns

h. Filters serialization (save/restore)

When u work with patch and filter out some functions during the analysis sometimes u want to save current state of filters close IDA and return to work later. BinDiffFilter provides that possibility via loadFilters and saveFilters.
It uses python pickling module to save current stat of filters and to restore it.

4. Problems

a. BinDiffFilter can not find BinDiff window

Somtimes in somehow BinDiffFilter is not able to find reference to QTableView object and in this case u should:

1. Close “Matched Functions” window.
2. Re-open it via View->Open subviews->BinDiff Matched Functions
3. Execute AutoDiff option “Summarize 2.0” or just call in console:
   AutoDiff.BinDiffFilter.findBinDiffWindow()
4. Repeat following steps untill u see message like this one presented below

[+]BinDiffFilter plugin loaded
Welcome in AutoDiff!!!
Python>AutoDiff.BinDiffFilter.findBinDiffWindow()
True