



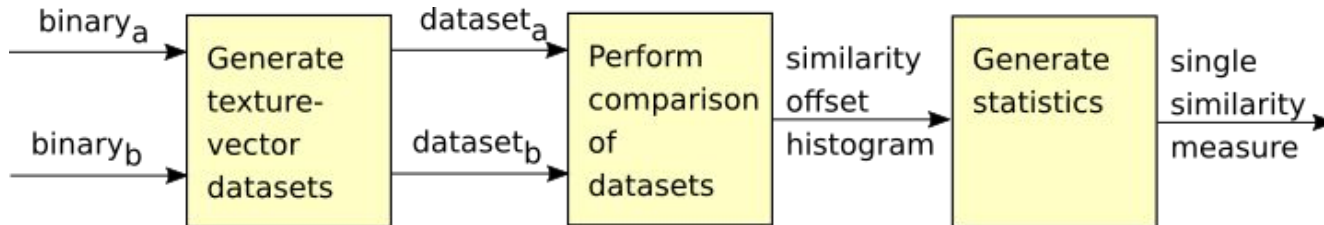
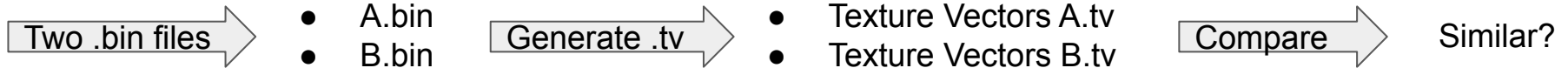
NAVAL  
POSTGRADUATE  
SCHOOL

# Using Texture Vector Analysis to identify File Similarity

Bruce Allen  
Neil Rowe

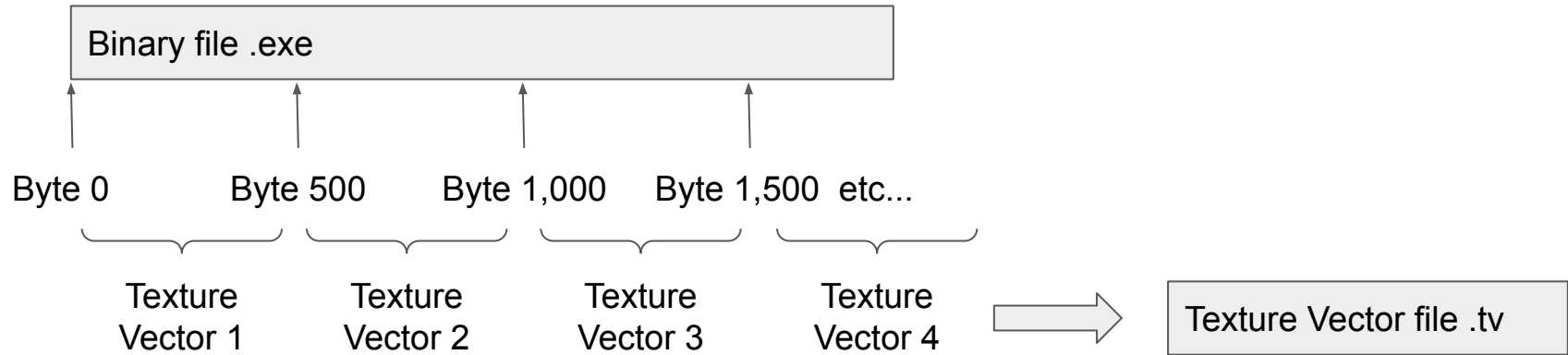
# Detecting Similarity using Texture Vectors

- No bytecode analysis
- No file structure decomposition
- Using Texture Vectors (.tv files)



# Texture Vector Generation

1. Break a file into even sections, for example 500-byte sections
2. For each section:
  - a. Calculate its Texture Vector

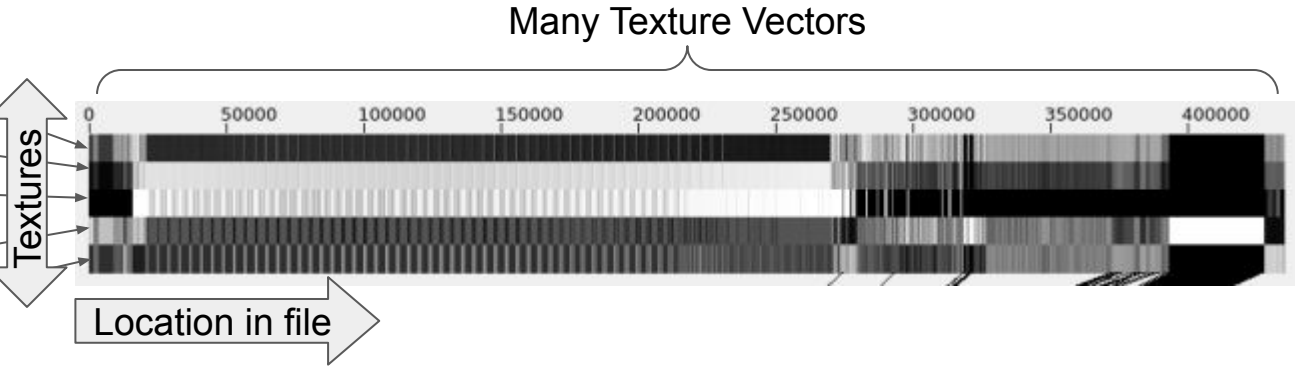


# Texture Vector Anatomy

Texture Vector: A vector of transforms calculated across a section of data

Five Transforms:

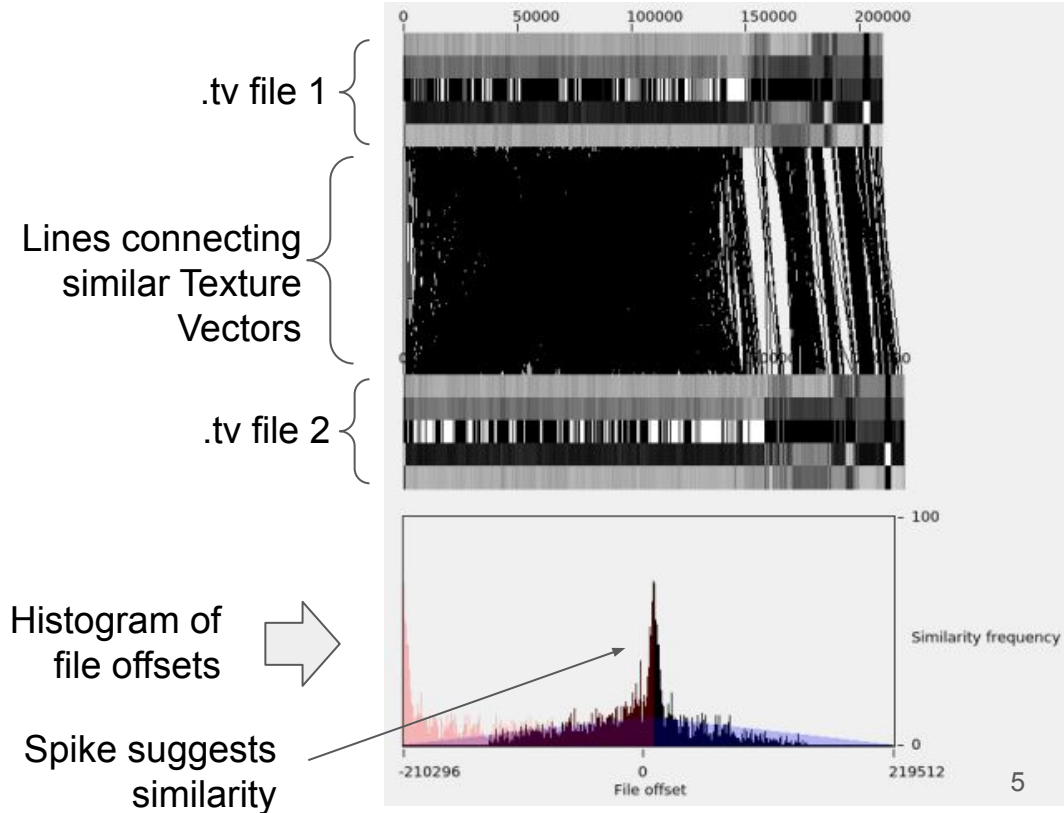
- Standard Deviation
- Mean
- Mode
- Mode Count
- Entropy



Collectively, Texture Vectors produce a file's "fingerprint" as a spectrogram

# Calculating Similarity Measures

- Similarity between two Texture Vectors
  - Euclidean distance:
$$w1(dv1)^2 + w2(dv2)^2 + w3(dv3)^2 + w4(dv4)^2 + w5(dv5)^2$$
- Similarity between two files
  - Histogram of file offsets between similar Texture Vectors
  - The similarity measure value is the standard deviation of the values of the histogram bars



# Our Dataset

- 1,134 files
- 23 groups of similar files
  - 22 groups of .exe and .dll files from Real Data Corpus
  - 1 group of Python scripts as a “dissimilar” data type
- 642,411 similarity comparisons calculated using Hamming

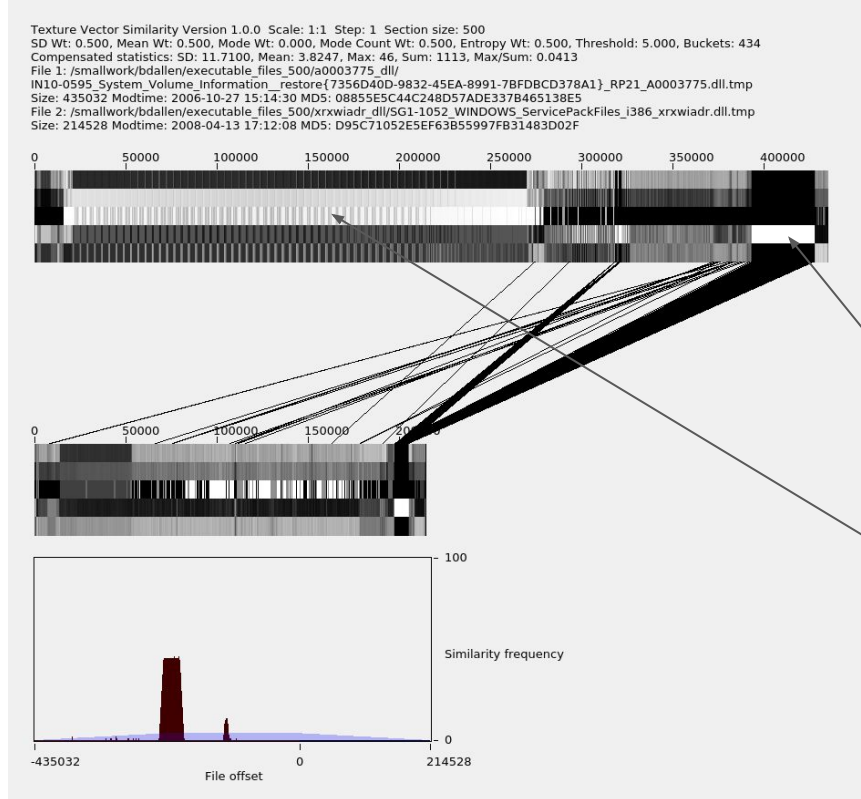
# Similarity by File Group

- Diagonal lines show that files within file groups are more similar than files across file groups
  - Usually

Family	n	1	2	3	4	5	6	7	8	9	10	11	12
a0003775_dll	1	4.5	1.2	5.4	2.1	3.8	3.0	3.6	2.8	2.2	3.3	5.1	2.3
bthserv_dll	2	1.2	3.7	1.2	0.7	0.7	0.7	0.5	0.7	0.6	0.3	0.9	0.6
ccalert_dll	3	5.4	1.2	11.4	2.5	3.9	3.2	2.2	2.9	3.6	4.3	4.8	2.2
cdfview_dll	4	2.1	0.7	2.5	10.0	1.3	1.1	1.6	2.2	1.8	0.9	1.7	0.8
dunzip32_dll	5	3.8	0.7	3.9	1.3	5.1	2.3	4.0	2.1	2.0	3.4	3.6	1.6
hotfix_exe	6	3.0	0.7	3.2	1.1	2.3	8.5	1.3	2.0	1.4	3.8	3.1	1.6
iexplore_exe	7	3.6	0.5	2.2	1.6	4.0	1.3	130.2	9.3	1.6	2.4	1.5	7.5
mobsync_exe	8	2.8	0.7	2.9	2.2	2.1	2.0	9.3	6.1	1.5	2.3	2.7	1.6
msrde_dll	9	2.2	0.6	3.6	1.8	2.0	1.4	1.6	1.5	4.5	1.4	2.0	0.9
nvrshu_dll	10	3.3	0.3	4.3	0.9	3.4	3.8	2.4	2.3	1.4	32.9	6.2	2.1
pacman_exe	11	5.1	0.9	4.8	1.7	3.6	3.1	1.5	2.7	2.0	6.2	1.5	2.2
policytool_exe	12	2.3	0.6	2.2	0.8	1.6	1.6	7.5	1.6	0.9	2.1	2.2	2.6
powerpnt_exe	13	3.5	0.4	2.2	1.1	3.1	1.5	41.2	5.8	1.4	2.6	2.2	4.6
rtinstaller32_exe	14	3.4	0.9	4.1	2.0	3.6	2.0	1.6	2.3	2.2	2.3	2.8	1.2
safrslv_dll	15	1.9	0.9	2.2	1.1	1.2	1.6	1.1	1.1	0.7	2.0	2.0	1.0
tabulate_drive_data_py	16	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.1	-	0.1	-	0.3
typeaheadfind_dll	17	0.9	0.6	1.3	0.7	0.4	0.3	0.4	0.5	0.7	0.1	0.7	0.5
udlaunch_exe	18	2.9	0.4	3.3	1.1	2.5	-	1.3	1.7	1.8	3.3	3.0	-
vsplugin_dll	19	3.0	0.6	3.4	1.0	2.0	2.5	4.0	1.8	1.2	3.2	3.0	1.6
webclnt_dll	20	3.3	1.0	3.6	1.1	2.3	1.3	1.8	1.8	1.5	2.2	2.8	1.0
winprint_dll	21	0.8	0.5	0.9	0.4	0.6	0.5	0.4	0.5	0.5	0.4	0.6	0.6
wmplayer_exe	22	3.1	0.4	3.1	0.9	2.4	2.0	21.7	3.6	1.3	3.0	2.8	2.4
xrxiadr_dll	23	11.5	0.8	12.1	2.5	9.2	4.1	3.2	4.6	3.3	12.9	13.0	3.8

Family	n	13	14	15	16	17	18	19	20	21	22	23
a0003775_dll	1	3.5	3.4	1.9	0.1	0.9	2.9	3.0	3.3	0.8	3.1	11.5
bthserv_dll	2	0.4	0.9	0.9	0.1	0.6	0.4	0.6	1.0	0.5	0.4	0.8
ccalert_dll	3	2.2	4.1	2.2	0.1	1.3	3.3	3.4	3.6	0.9	3.1	12.1
cdfview_dll	4	1.1	2.0	1.1	0.1	0.7	1.1	1.0	1.1	0.4	0.9	2.5
dunzip32_dll	5	3.1	3.6	1.2	0.1	0.4	2.5	2.0	2.3	0.6	2.4	9.2
hotfix_exe	6	1.5	2.0	1.6	0.1	0.3	-	2.5	1.3	0.5	2.0	4.1
iexplore_exe	7	41.2	1.6	1.1	0.2	0.4	1.3	4.0	1.8	0.4	21.7	3.2
mobsync_exe	8	5.8	2.3	1.1	0.1	0.5	1.7	1.8	1.8	0.5	3.6	4.6
msrde_dll	9	1.4	2.2	0.7	-	0.7	1.8	1.2	1.5	0.5	1.3	3.3
nvrshu_dll	10	2.6	2.3	2.0	0.1	0.1	3.3	3.2	2.2	0.4	3.0	12.9
pacman_exe	11	2.2	2.8	2.0	-	0.7	3.0	3.0	2.8	0.6	2.8	13.0
policytool_exe	12	4.6	1.2	1.0	0.3	0.5	-	1.6	1.0	0.6	2.4	3.8
powerpnt_exe	13	76.0	1.5	1.0	0.2	0.2	1.5	2.8	1.7	0.3	12.6	8.2
rtinstaller32_exe	14	1.5	13.4	1.1	0.1	0.4	3.1	1.9	2.0	0.6	1.7	6.3
safrslv_dll	15	1.0	1.1	3.3	0.1	0.8	-	1.4	1.2	0.6	1.1	2.6
tabulate_drive_data_py	16	0.2	0.1	0.1	2.8	0.1	-	0.2	0.2	-	0.1	0.3
typeaheadfind_dll	17	0.2	0.4	0.8	0.1	2.3	0.2	0.5	0.8	0.4	0.2	0.6
udlaunch_exe	18	1.5	3.1	-	-	0.2	-	2.1	0.9	0.5	2.2	3.6
vsplugin_dll	19	2.8	1.9	1.4	0.2	0.5	2.1	3.2	1.7	0.5	2.7	3.5
webclnt_dll	20	1.7	2.0	1.2	0.2	0.8	0.9	1.7	3.8	0.7	1.5	5.0
winprint_dll	21	0.3	0.6	0.6	-	0.4	0.5	0.5	0.7	1.1	0.4	0.6
wmplayer_exe	22	12.6	1.7	1.1	0.1	0.2	2.2	2.7	1.5	0.4	9.1	5.7
xrxiadr_dll	23	8.2	6.3	2.6	0.3	0.6	3.6	3.5	5.0	0.6	5.7	15.9

# False Positives



- Regions of uniform pattern
  - e.g. tables
- Regions of high entropy
  - e.g. compressed content

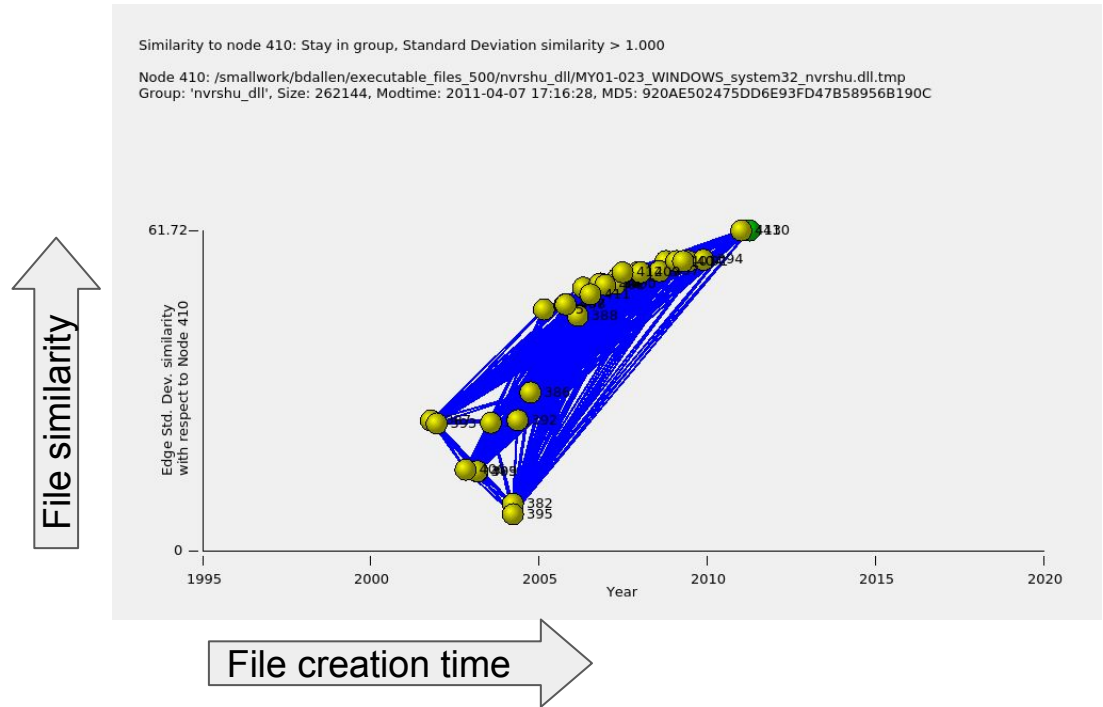


# Time-based Analysis

Infer change based on similarity and timestamp

- Version change
- Virus injection

Example: File family nvrshu\_dll



# Texture-Vector Dataset and Tools

- Dataset

- The 1,134 texture vector (.tv) files
- The node and edge data for creating similarity graphs

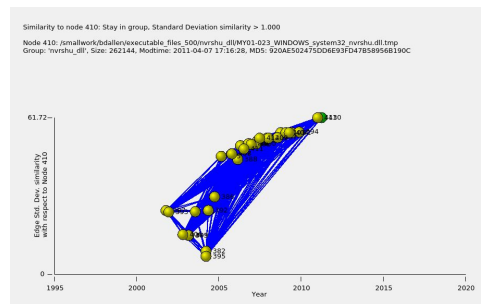
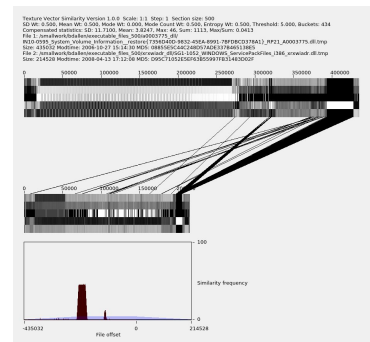
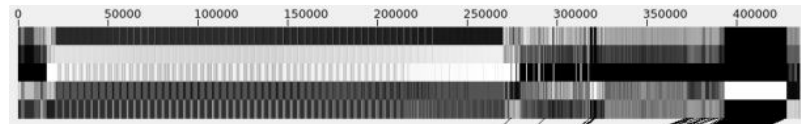
- Tools

- **calc\_tv.py** - calculate .tv files from binary files
- **tv.py** - plot similarity between pairs of files
- **tv\_browser.py** - graph similarities between groups of files

- Where

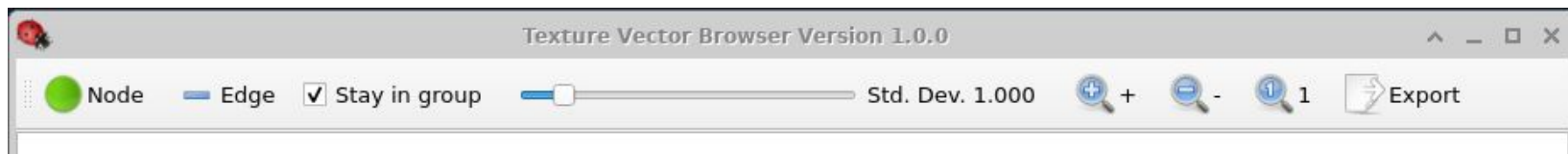
- GitHub:

[https://github.com/NPS-DEEP/tv\\_sim](https://github.com/NPS-DEEP/tv_sim)

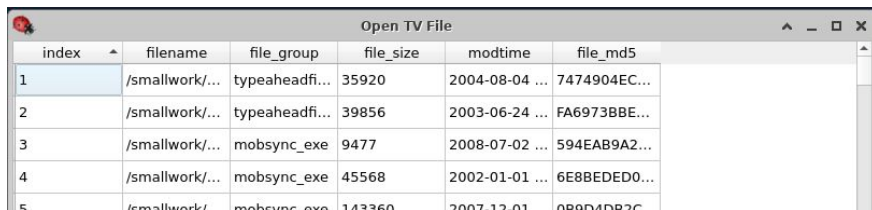


# Real time Analysis (1 of 2)

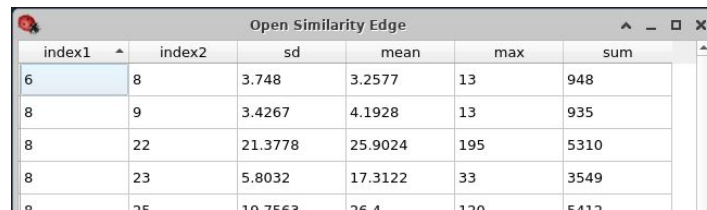
- Download: Clone from [https://github.com/NPS-DEEP/tv\\_sim](https://github.com/NPS-DEEP/tv_sim)
- Type: “cd tv\_sim/python; ./tv\_browser.py”:



- Then:
  - Click on “Node” to inspect files or on “Edge” to inspect similarity between files
  - Then hover or click on nodes or edges for analysis

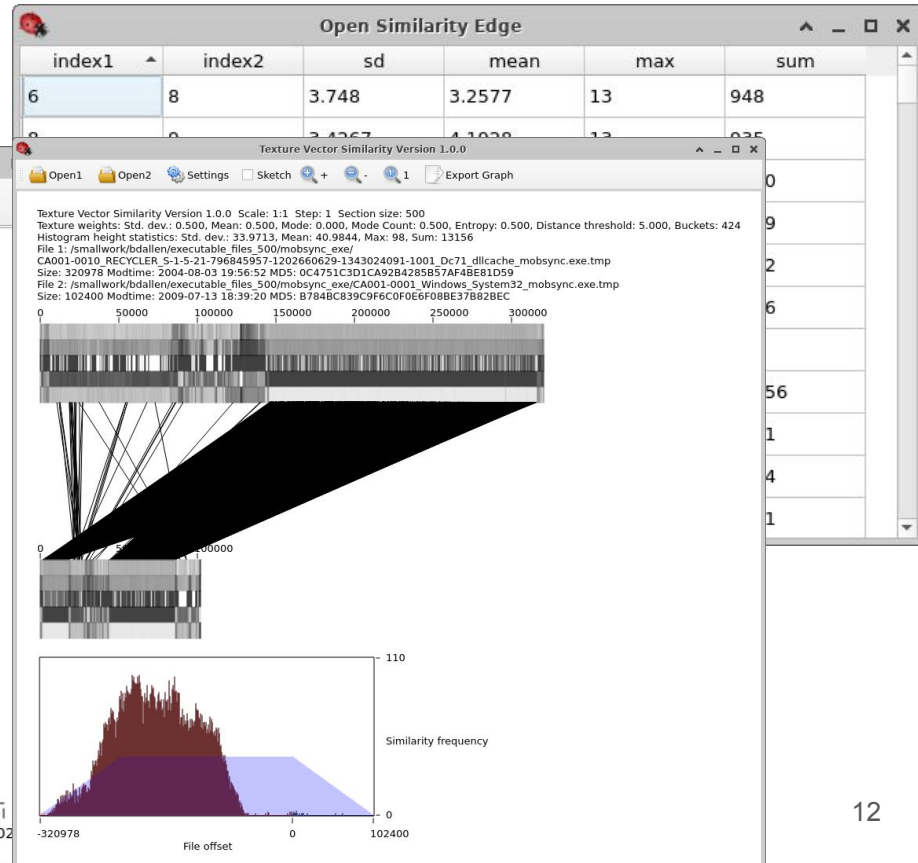
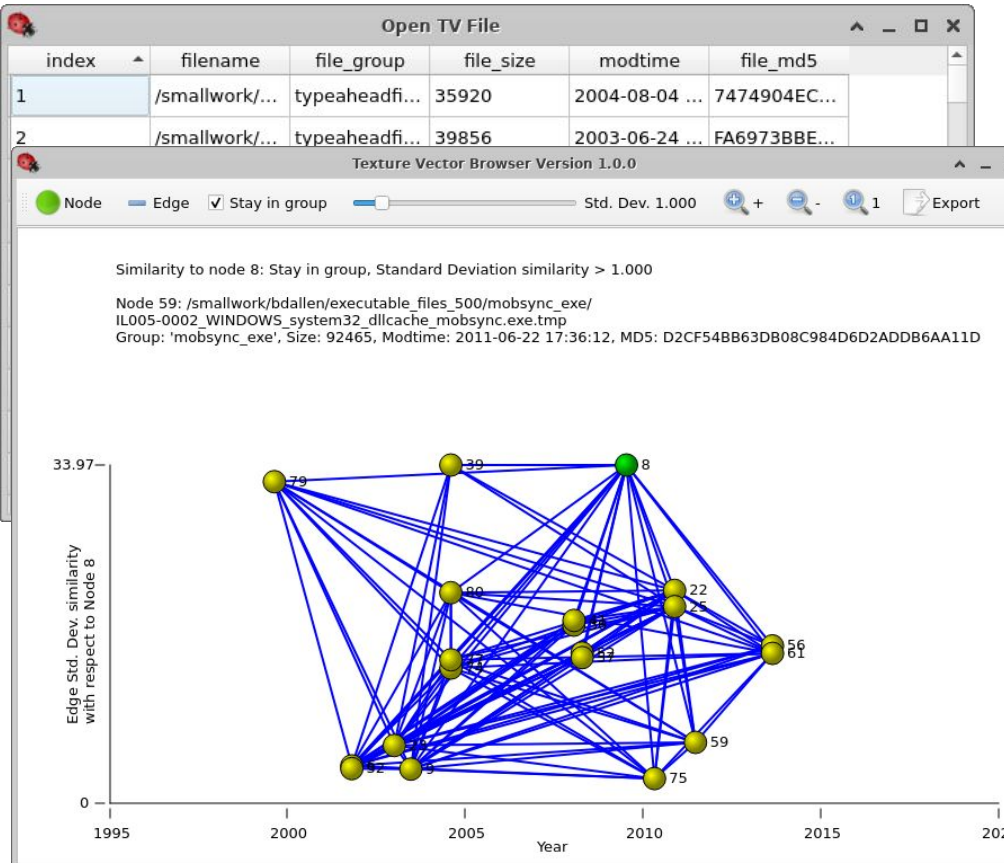


index	filename	file_group	file_size	modtime	file_md5
1	/smallwork/...	typeaheadfi...	35920	2004-08-04 ...	7474904EC...
2	/smallwork/...	typeaheadfi...	39856	2003-06-24 ...	FA6973BBE...
3	/smallwork/...	mobsync_exe	9477	2008-07-02 ...	594EAB9A2...
4	/smallwork/...	mobsync_exe	45568	2002-01-01 ...	6E8BEDED0...
5	/smallwork/...	mobsync_exe	143360	2007-12-01	0R9D4DR2C



index1	index2	sd	mean	max	sum
6	8	3.748	3.2577	13	948
8	9	3.4267	4.1928	13	935
8	22	21.3778	25.9024	195	5310
8	23	5.8032	17.3122	33	3549
8	25	10.7563	26.1	120	5412

# Real time Analysis (2 of 2)



# Future Work

- Classify file types by their texture: .exe, .wav, .jpg ...
- Evaluate similarity by evaluating runs of texture patterns
- Identify and remove false-positives from the similarity calculation

