

demo

August 8, 2020

1 Demo for functional usage of CFG-explorer

Now, `cfg-explorer` can not only be used as a command line tool. We can also call it within a Python program.

1.1 Download Spec CPU Benchmark 2006

Save the suite outside our current repository:

```
$ cd ..  
$ git clone https://github.com/Multi2Sim/m2s-bench-spec2006
```

Every `.i386` file is a binary file for testing.

1.2 Import Libraries

If you have not installed `cfg_exploer` module, you can install it with

```
$ pip install cfg_explorer
```

Of course, If you just want to try it without installation, it's Okay:

First, to import `cfg_explore` in this subdirectory, you should include your `cfg-explorer` path into your `PATH` environment variable. You can do this by (suppose the whole `cfg-explorer` directory is located in `$HOME/cfg-explorer`):

```
$ export PATH=$HOME/cfg-explorer:$PATH
```

Or import it in this notebook by such an approach for that the target folder is actually the parent folder of this file:

```
[1]: import os  
import sys  
from pathlib import Path  
  
sys.path.insert(0, str(Path().resolve().parent))
```

1.3 Usages of `cfg_explore` Function

```
[2]: from cfgexplorer import cfg_explore
```

1.3.1 Launch an interactive web app

Now, call `cfg_explore` with the only argument `binary`, which is the path of the binary file we prepare to analysis. After running, it will host a website on `http://127.0.0.1:5050/` to show the *control flow graph* of the file. You can specify the port by `port` parameter.

```
[3]: cfg_explore(binary='../m2s-bench-spec2006/999.specrand/specrand_base.i386')
```

```
WARNING | 2020-08-08 00:30:05,082 | angr.analyses.cfg.cfg_fast |  
"collect_data_references" is deprecated and will be removed soon. Please  
use "data_references" instead  
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00  
  
* Serving Flask app "cfgexplorer.explorer" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production  
deployment.  
  Use a production WSGI server instead.  
* Debug mode: on  
  
INFO      | 2020-08-08 00:30:05,268 | werkzeug | * Running on  
http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Whenever you want to shut down the app, just interrupt the function. For example, in this notebook, click on interrupt the kernel button on the toolbar.

1.3.2 Generate multiple formats of outputs

Instead of launching a web app, you can also let `cfg_explore` generate output files by specifying `output` argument.

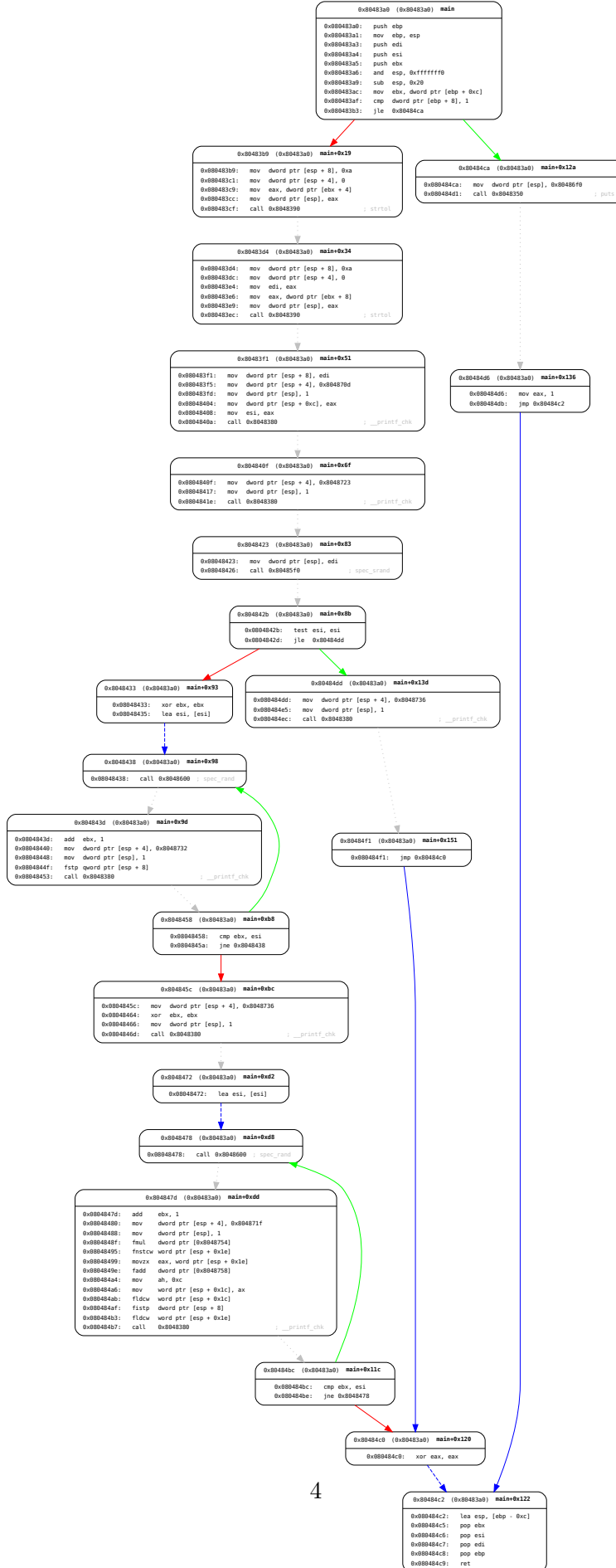
For instance, we can get the `svg` format CFG of `specrand_base.i386` using:

```
[4]: cfg_explore(binary='../m2s-bench-spec2006/999.specrand/specrand_base.  
    ↪i386',output='./test.svg')
```

```
WARNING | 2020-08-08 00:30:09,414 | angr.analyses.cfg.cfg_fast |  
"collect_data_references" is deprecated and will be removed soon. Please  
use "data_references" instead  
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
```

`.svg` files can be opened by web-browser directly, it can also be displayed in this notebook:

```
[5]: from IPython.core.display import SVG
display(SVG('test.svg'))
```



There are also a bunch of formats that `cfg_explore` can handle. The only thing you need to do is to specify proper suffix (i.e. extension name) in path which is passed as `output` argument.

Current supported formats are listed as below:

```
[6]: formats = [
    'canon', 'cmap', 'cmapx', 'cmapx_np', 'dot', 'fig', 'gd', 'gd2', 'gif',
    'imap', 'imap_np', 'imap', 'jpe', 'jpeg', 'jpg', 'mp', 'pdf', 'plain',
    'plain-ext', 'png', 'ps', 'ps2', 'svg', 'svgz', 'vml', 'vmlz', 'vrml',
    'wbmp', 'xdot', 'raw'
]
```

Try all formats by:

```
[7]: for format in formats:
    cfg_explore(binary='../m2s-bench-spec2006/999.specrand/specrand_base.
    ↪i386', output='./test.'+format)
```

```
WARNING | 2020-08-08 00:26:14,674 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:14,903 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:15,141 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:15,557 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:15,794 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:16,032 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:16,272 | angr.analyses.cfg.cfg_fast |
```

```

"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:16,684 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:17,083 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:17,568 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:17,822 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:18,082 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:18,346 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:18,666 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:19,106 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:19,419 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:19,702 | angr.analyses.cfg.cfg_fast |

```

```

"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:19,986 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:20,267 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:20,550 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:21,025 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:21,441 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:21,739 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:22,031 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:22,351 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:22,646 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:22,956 | angr.analyses.cfg.cfg_fast |

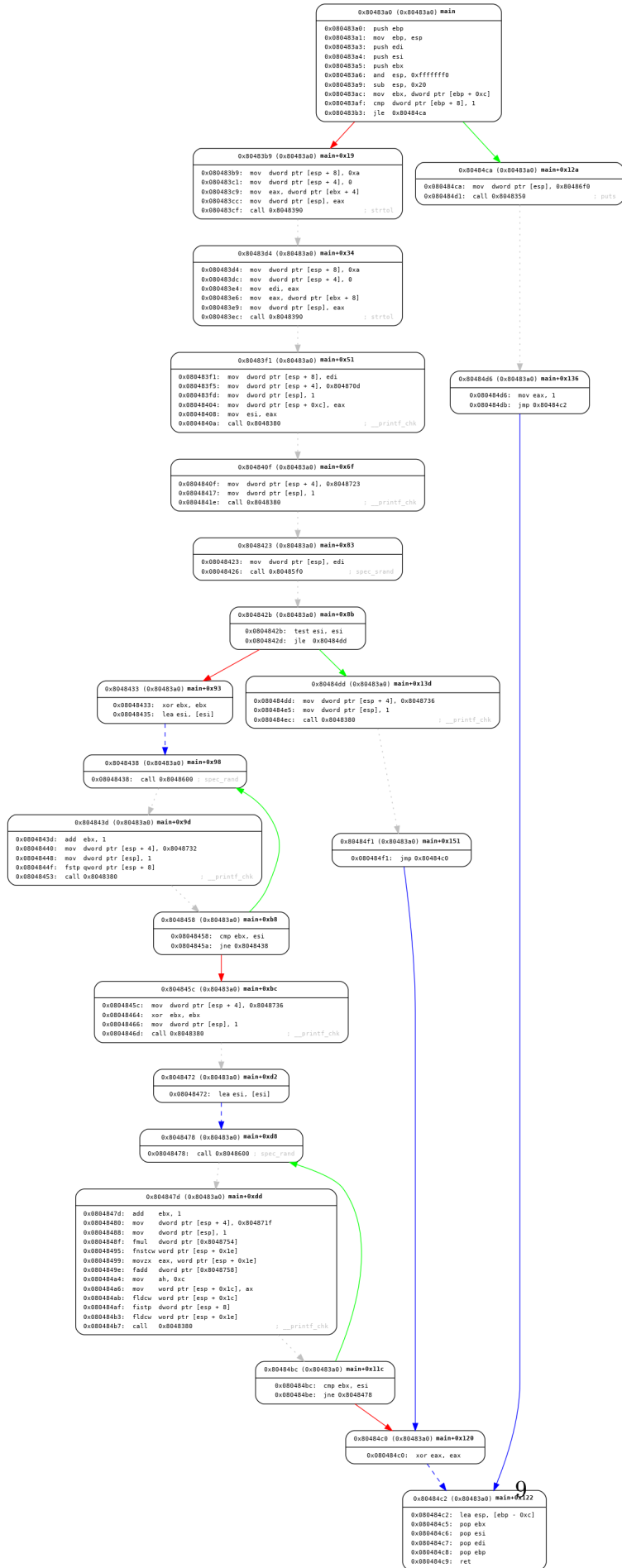
```

```

"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:23,514 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:23,898 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00
WARNING | 2020-08-08 00:26:24,209 | angr.analyses.cfg.cfg_fast |
"collect_data_references" is deprecated and will be removed soon. Please
use "data_references" instead
100% |#####| Elapsed Time: 0:00:00 Time: 0:00:00

```

View the output png:



1.3.3 Traversal a large folder to generate all CFGs

We still use `m2s-bench-spec2006` as an example.

Assume that we need to analyze all binary files in this folder. Wrapping `cfg-explorer` as a function makes the task more flexible inside a Python script.

First, get all potential binary files for analysis:

```
[8]: from glob import glob
progs = sorted(glob('../m2s-bench-spec2006/*/*.i386'))
progs
```

```
[8]: ['../m2s-bench-spec2006/401.bzip2/bzip2_base.i386',
      '../m2s-bench-spec2006/403.gcc/gcc_base.i386',
      '../m2s-bench-spec2006/410.bwaves/bwaves_base.i386',
      '../m2s-bench-spec2006/416.gamess/gamess_base.i386',
      '../m2s-bench-spec2006/429.mcf/mcf_base.i386',
      '../m2s-bench-spec2006/433.milc/milc_base.i386',
      '../m2s-bench-spec2006/434.zeusmp/zeusmp_base.i386',
      '../m2s-bench-spec2006/435.gromacs/gromacs_base.i386',
      '../m2s-bench-spec2006/436.cactusADM/cactusADM_base.i386',
      '../m2s-bench-spec2006/437.leslie3d/leslie3d_base.i386',
      '../m2s-bench-spec2006/444.namd/namd_base.i386',
      '../m2s-bench-spec2006/445.gobmk/gobmk_base.i386',
      '../m2s-bench-spec2006/447.dealII/dealII_base.i386',
      '../m2s-bench-spec2006/450.soplex/soplex_base.i386',
      '../m2s-bench-spec2006/453.povray/povray_base.i386',
      '../m2s-bench-spec2006/454.calculix/calculix_base.i386',
      '../m2s-bench-spec2006/456.hammer/hammer_base.i386',
      '../m2s-bench-spec2006/458.sjeng/sjeng_base.i386',
      '../m2s-bench-spec2006/459.GemsFDTD/GemsFDTD_base.i386',
      '../m2s-bench-spec2006/462.libquantum/libquantum_base.i386',
      '../m2s-bench-spec2006/464.h264ref/h264ref_base.i386',
      '../m2s-bench-spec2006/465.tonto/tonto_base.i386',
      '../m2s-bench-spec2006/470.lbm/lbm_base.i386',
      '../m2s-bench-spec2006/471.omnetpp/omnetpp_base.i386',
      '../m2s-bench-spec2006/473.astar/astar_base.i386',
      '../m2s-bench-spec2006/481.wrf/wrf_base.i386',
      '../m2s-bench-spec2006/482.sphinx3/sphinx3_base.i386',
      '../m2s-bench-spec2006/483.xalancbmk/xalancbmk_base.i386',
      '../m2s-bench-spec2006/998.specrand/specrand_base.i386',
      '../m2s-bench-spec2006/999.specrand/specrand_base.i386']
```

```
[9]: # create a directory to store the outputs
out_dir = './output'
if not os.path.exists(out_dir):
    os.mkdir(out_dir)
```

Then, we can simply call `cfg_explore` function inside loops. Keep it alone, we just need to wait for generating all '.svg' files.

Note: it might take a extremely long time. Be patient.

```
[10]: for p in progs:
      name = p.split('/')[3]
      print('start analysis of:',name)
      output_file = os.path.join(out_dir, name + '.svg')
      if not os.path.exists(output_file):
          cfg_explore(binary=p,output=output_file)
```

```
start analysis of: 401.bzip2
start analysis of: 403.gcc
start analysis of: 410.bwaves
start analysis of: 416.gamess
start analysis of: 429.mcf
start analysis of: 433.milc
start analysis of: 434.zeusmp
start analysis of: 435.gromacs
start analysis of: 436.cactusADM
start analysis of: 437.leslie3d
start analysis of: 444.namd
start analysis of: 445.gobmk
start analysis of: 447.dealII
start analysis of: 450.soplex
start analysis of: 453.povray
start analysis of: 454.calculix
start analysis of: 456.hmmmer
start analysis of: 458.sjeng
start analysis of: 459.GemsFDTD
start analysis of: 462.libquantum
start analysis of: 464.h264ref
start analysis of: 465.tonto
start analysis of: 470.lbm
start analysis of: 471.omnetpp
start analysis of: 473.astar
start analysis of: 481.wrf
start analysis of: 482.sphinx3
start analysis of: 483.xalancbmk
start analysis of: 998.specrand
start analysis of: 999.specrand
```

Now, you can view all outputs in `out_dir`.