



Runtime Verification of Scientific Software

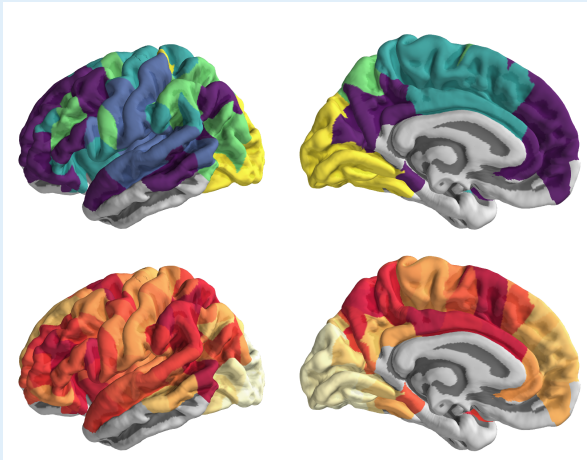
Maxwell Shinn, Clarence Lehman, and Ruzica Piskac

October 31, 2018

Motivation

```
def graph_measure(filename):  
    timeseries = load_from_csv(filename)  
    corr_matrix = corr_coef(timeseries)  
    normalized = fisher_transform(corr_matrix)  
    G = matrix_to_graph(normalized)  
    return graph_clustering(G)
```

Expected result



Actual result



Motivation

```
def graph_measure(filename):  
    timeseries = load_from_csv(filename)  
    corr_matrix = corr_coef(timeseries)  
    normalized = fisher_transform(corr_matrix)  
    G = matrix_to_graph(normalized)  
    return graph_clustering(G)
```

Motivation

```
def graph_measure(filename):  
    timeseries = load_from_csv(filename)  
    corr_matrix = corr_coef(timeseries) diag = 1+1e-10  
    normalized = fisher_transform(corr_matrix)  
    G = matrix_to_graph(normalized)  
    return graph_clustering(G)
```

Motivation

```
def graph_measure(filename):  
    timeseries = load_from_csv(filename)  
    corr_matrix = corr_coef(timeseries) diag = 1+1e-10  
    normalized = fisher_transform(corr_matrix) = NaN  
    G = matrix_to_graph(normalized)  
    return graph_clustering(G)
```

Motivation

```
def graph_measure(filename):  
    timeseries = load_from_csv(filename)  
    corr_matrix = corr_coef(timeseries) diag = 1+1e-10  
    normalized = fisher_transform(corr_matrix) = NaN  
    G = matrix_to_graph(normalized) NaN  $\rightarrow$  0  
    return graph_clustering(G)
```


Motivation

```
def graph_measure(filename):  
    timeseries = load_from_csv(filename)  
    corr_matrix = corr_coef(timeseries) diag = 1+1e-10  
    normalized = fisher_transform(corr_matrix) = NaN  
    G = matrix_to_graph(normalized) NaN  $\rightarrow$  0  
    return graph_clustering(G) ???
```

Our tool

PARANOID
SCIENTIST

The logo consists of the words "PARANOID" and "SCIENTIST" stacked vertically in a bold, black, sans-serif font. A magnifying glass icon is positioned over the letter "O" in "PARANOID", with its handle extending downwards and to the right, crossing over the letter "I" in "PARANOID" and the letter "T" in "SCIENTIST".

Our tool

PARANOID SCIENTIST



- In scientific software, we don't know what the output should be

Our tool

PARANOID SCIENTIST



- ▶ In scientific software, we don't know what the output should be
- ▶ Python library for scientific software

Our tool

PARANOID SCIENTIST

The logo for 'PARANOID SCIENTIST' features the word 'PARANOID' on the top line and 'SCIENTIST' on the bottom line. The letter 'O' in 'PARANOID' is replaced by a magnifying glass icon, with the handle of the magnifying glass extending downwards and to the right, pointing towards the letter 'I' in 'SCIENTIST'.

- ▶ In scientific software, we don't know what the output should be
- ▶ Python library for scientific software
- ▶ Checks entry and exit conditions

Our tool

PARANOID SCIENTIST



- ▶ In scientific software, we don't know what the output should be
- ▶ Python library for scientific software
- ▶ Checks entry and exit conditions
- ▶ Conditions specified with refinement types

Our tool

PARANOID SCIENTIST

- ▶ In scientific software, we don't know what the output should be
- ▶ Python library for scientific software
- ▶ Checks entry and exit conditions
- ▶ Conditions specified with refinement types
- ▶ Specification serves as documentation

Our tool

PARANOID SCIENTIST

The logo for 'PARANOID SCIENTIST' features the words in a bold, black, sans-serif font. The letter 'O' in 'PARANOID' is replaced by a magnifying glass icon, with the handle of the magnifying glass extending downwards and to the right, pointing towards the word 'SCIENTIST'.

- ▶ In scientific software, we don't know what the output should be
- ▶ Python library for scientific software
- ▶ Checks entry and exit conditions
- ▶ Conditions specified with refinement types
- ▶ Specification serves as documentation
- ▶ Automated offline testing