Please pull the latest TSAT source code because changes were made in the java in order to create the python version.

https://github.com/dwicke/TSAT

The source is here:

https://github.com/dwicke/TSAT/blob/master/python/tsail.py

You will need to have run

mvn package -Psingle

to compile TSAT and generate the jar file in:

```
/target/tsat-0.0.1-SNAPSHOT-jar-with-dependencies.jar
```

This is because the python library uses the java command line interface to TSAT to produce the output.  Within tsail.py you can modify the location of your jar file if you move it or rename it.

In the python directory in TSAT you can run python and then type:

import tsail
Then you can call the functions like

tsail.buildMotifs
tsail.RRA
tsail.RPMTrainTest
tsail.RPMTrain
tsail.RPMTest

For motifs, anomaly detection and representative pattern matching respectively.

TSAT CLI produces a json file that is the serialized version of the result object for the particular function.  I then can import that file as a dictionary in python and return that to you.  Below I refer to the java classes (that have documentation) of the contents of the dictionaries.

For buildMotifs it returns a dictionary where you can index the motifs (or rules generated by Sequitur).

<returned_dict> is a GrammarRules object that has a rules map indexed by the integer rule number starting at ‘0’.
https://github.com/jMotif/GI/blob/master/src/main/java/net/seninp/gi/logic/GrammarRules.java

<returned_dict>['rules']['rule_number'] will give you a GrammarRuleRecord for a particular rule_number:
https://github.com/jMotif/GI/blob/master/src/main/java/net/seninp/gi/logic/GrammarRuleRecord.java

For example, <returned_dict>['rules']['1'] will give you the grammarRuleRecord for rule 1.

The values within the GrammarruleRecord can then be accessed as you do a dict.

Also, note that `ruleIntervals` is and array of:
https://github.com/jMotif/GI/blob/32f58578f5a0b184fc836f9d397aa0bfc8e68ee6/src/main/java/net/seninp/gi/logic/RuleInterval.java

To access this you just do:
<returned_dict>['rules']['1']['ruleIntervals'][<rule_interval_index>]
For example,
<returned_dict>['rules']['1']['ruleIntervals'][0]

# RRA

When calling RRA you are returned a dictionary representation of DiscordRecords:

https://github.com/jMotif/SAX/blob/27607baa823df21a10d10e80827ffdd15090cbd9/src/main/java/net/seninp/jmotif/sax/discord/DiscordRecords.java

Which is a list of DiscordRecord based on:

https://github.com/jMotif/SAX/blob/660e837edf1c8058eac6ef05185c7f83e12f3689/src/main/java/net/seninp/jmotif/sax/discord/DiscordRecord.java

The same description for accessing the values as above is used.

# RPM

There are three functions:

RPMTrainTest, RPMTrain, and RPMTest.

For RPMTrainTest both training and testing will be output.

The training output is an array of:

https://github.com/dwicke/TSAT/blob/72498ab66795e221eb6e26fbc65b0f156169ca66/src/main/java/edu/gmu/grammar/patterns/TSPattern.java

The test output is a 2D array of strings for each instance we have the value corresponds to [["inst#", "actual class", "predicted class","timeSeries"]]

The test output when the data input is unlabeled is a 2D array of strings where for each instance we have:

[["inst#", "comma separated list of the probabilities of being in the particular class", "predicted class", "timeSeries"]]

When running RPMTrainTest you will generate three files as output instead of 1.

<outputfileName>.train
<outputfileName>.test
<outputfileName>

Train and test are the json of the python dicts as talked about previously. The last file can be imported into TSAT GUI as it is the same as the saved model in TSAT.