Entropia 1.5

User's Guide

Hanan Alkhammash

The University of Melbourne
halkhammash@student.unimelb.edu.au

Entropia is an open-source command-line tool that implements a family of conformance checking measures. This guide is intended to explain through a series of simple steps how to use the *Entropia* tool to measure precision and recall between a model and an event log using classical non-deterministic and stochastic conformance checking approaches. This guide covers the following topics:

1. Getting Started: find out how to get *Entropia* and prerequisites required to run the tool.
2. Classical Non-deterministic Conformance Checking Measures: The *Entropia* commands are listed and each command line instruction is illustrated by an example, making it easy to find the information you are looking for. The section covers the following measures:
    1. Exact Matching Precision and Recall [5],
    2. Partial Matching Precision and Recall [3]; and
    3. Controlled Partial Matching Precision and Recall [1],
3. The *Entropia* commands for stochastic conformance checking measures. Here, you can find a detailed description of different stochastic conformance checking approaches. The section includes the following measures:
    1. Stochastic Precision and Recall [2]; and
    2. Entropic Relevance [4].

Getting Started with Entropia

This section gets you started using the *Entropia* tool.

a) Checking for Prerequisite

Prior to running the *Entropia* commands, you will need to ensure that the JDK (Java Development Kit) is in place. Note that the tool was built using JDK 1.8.

b) Downloading and Running Entropia

1. Clone or download the JPBT library to your local machine: https://github.com/jbpt/codebase/.
2. Navigate to the **jbpt-pm/entropia/** folder in your terminal.
3. Issue the following command to verify that the *Entropia* tool is properly downloaded and display its version number as shown in the output screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -v
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -v
1.5
```

Use the option (-h) to display the help message that shows information of the core and specific options of the *Entropia* tool.

```
>java -jar jbpt-pm-entropia-1.5.jar -h
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -h
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

usage: java -jar jbpt-pm.jar <options>
 -cpmp,--controlled-partial-precision          compute entropy-based precision (controlled
                                               partial trace matching with fixed number of skips)
 -cpmpr,--controlled-partial-precision-recall  compute entropy-based precision and recall
                                               (controlled partial trace matching with fixed
                                               number of skips)
 -cpmr,--controlled-partial-recall             compute entropy-based recall (controlled partial
                                               trace matching with fixed number of skips)
 -dent,--diluted-entropy                       compute entropy measure (for "diluted" traces)
 -emp,--precision                              compute entropy-based precision (exact trace
                                               matching)
 -empr,--precision-recall                      compute entropy-based precision and recall (exact
                                               trace matching)
 -emr,--recall                                 compute entropy-based recall (exact trace
                                               matching)
 -ent,--entropy                                compute entropy measure (for exact traces)
 -h,--help                                     print help message
 -pmp,--partial-precision                      compute entropy-based precision (partial trace
                                               matching)
 -pmpr,--partial-precision-recall              compute entropy-based precision and recall
                                               (partial trace matching)
 -pmr,--partial-recall                         compute entropy-based recall (partial trace
                                               matching)
 -r,--entropic-relevance                       compute entropic relevance for the given relevant
                                               (XES, CSV) and retrieved (DFG, SDFA) traces
 -rel,--relevant <file path>                   model that describes relevant traces
 -ret,--retrieved <file path>                  model that describes retrieved traces
 -s,--silent                                   print the results only
 -sp,--stochastic-precision                    compute stochastic precision for the given
                                               relevant (XES) and retrieved (sPNML) traces
 -spr,--stochastic-precision-recall            compute stochastic precision for the given
                                               relevant (XES) and retrieved (sPNML) traces
 -sr,--stochastic-recall                       compute stochastic recall for the given relevant
                                               (XES) and retrieved (sPNML) traces
 -srel,--srel <number of skips>                add specified amount of skips to relevant traces
 -sret,--sret <number of skips>                add specified amount of skips to retrieved traces
 -v,--version                                  get version of this tool
================================================================================
```

Models and Event Log

This tutorial uses example files provided along with the tool, which are:

- The event logs (log1.xes), presented in Table 1, and (log2.xes), presented in Table 1 in paper [2].
- The Petri net (model1.pnml), presented in Figure 1, and stochastic Petri net (model2.pnml), presented in Fig.1(b) in paper [2].
- The stochastic deterministic finite automaton (automaton.json) presented in Figure 2.

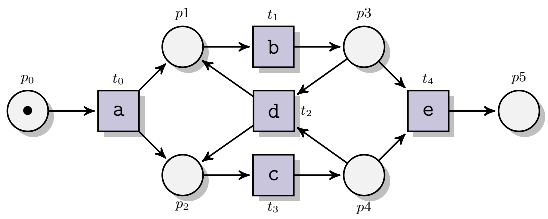

Figure 1: Petri Net.

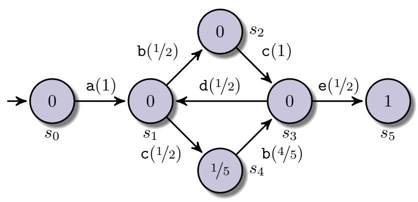

Figure 2: SDFA.

Trace
⟨a, b, c, e⟩
⟨a, b, c, d, c, b, e⟩
⟨a, b, d, c, b, e⟩
⟨a, c, e⟩
⟨b, c, e⟩
⟨b, c, e⟩
⟨a, a, a, c, b, e⟩

Table 1: Event Log

All the files are located in the folder **jbpt-pm/entropia/examples/**. If the folder does not contain the files, you can download them from `https://github.com/jbpt/codebase/tree/master/jbpt-pm/entropia/examples/`. It is recommended to use the event log and the models provided to understand how to run the *Entropia* commands. Then, you can try it with your own data. Refer to Table III in the demo paper for more details about file types and data formats supported by the tool.

Classical Non-Deterministic Conformance Checking Measures

a) Matching Precision and Recall Measures

To compute the exact matching precision value between the event log (log.xes) and process model (model.pnml), use the option (`-emp`) as follows.

```
>java -jar jbpt-pm-entropia-1.5.jar -emp -rel=log1.xes -ret=model1.pnml
```

Note that in the command the paths to the event log and process model files are assigned to the relevant (`-rel=<path>`) and retrieved (`-ret=<path>`) models respectively.

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -emp -rel=log1.xes -ret=model1.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

Computing eigenvalue-based precision/recall based on exact matching of traces.
The technique is described in:
Artem Polyvyanyy, Andreas Solti, Matthias Weidlich, Claudio Di Ciccio,
Jan Mendling. Monotone Precision and Recall for Comparing Executions and
Specifications of Dynamic Systems.
ACM Transactions on Software Engineering and Methodology (TOSEM) (2020)

Loading the retrieved model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\model1.pnml.
The retrieved modeln loaded in                            93 ms.
Loading the relevant model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\log1.xes.
The relevant model loaded in                              44 ms.
The boundedness of the retrieved model checked in         297 ms.
The boundedness of the relevant model checked in          0 ms.
Constructing automaton RET that encodes the retrieved model.
Automaton RET constructed in                              7 ms.
Automaton RET has 6 states and 7 transitions.
Constructing automaton REL that encodes the relevant model.
Automaton REL constructed in                              1 ms.
Automaton REL has 10 states and 14 transitions.

===================Calculating precision and recall=============================

The intersection INT of RET and REL constructed in        3 ms.
Automaton INT has 8 states and 8 transitions.

A largest eigenvalue of the adjacency matrix of RET is 1.4371560431001351.
A largest eigenvalue of the adjacency matrix of RET computed in     76 ms.

A largest eigenvalue of the adjacency matrix of INT is 1.1147978039377693.
A largest eigenvalue of the adjacency matrix of INT computed in     8 ms.

Precision computed in                                               84 ms.
Precision: 0.7756971202187645.
```

When the option (-s) is added to commands, the tool runs in a silent mode. The following command is an example of using the (-s) option:

```
>java -jar jbpt-pm-entropia-1.5.jar -emp -s -rel=log1.xes -ret=model1.pnml
```

The tool, in the silent mode, only prints out the result, the exact matching precision in this case, omitting the debug information and execution data. The expected output of the command will be as follows:

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -emp -s -rel=log1.xes -ret=model1.pnml
0.7756971226454726
```

By replacing the option (-emp) with (-emr), the tool computes the exact matching recall value between the event log and process model.

```
>java -jar jbpt-pm-entropia-1.5.jar -emr -rel=log1.xes -ret=model1.pnml
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -emr -rel=log1.xes -ret=model1.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
```

```
PNML format:     http://www.pnml.org/
XES format:      https://xes-standard.org/
================================================================================

Computing eigenvalue-based precision/recall based on exact matching of traces.
The technique is described in:
Artem Polyvyanyy, Andreas Solti, Matthias Weidlich, Claudio Di Ciccio,
Jan Mendling. Monotone Precision and Recall for Comparing Executions and
Specifications of Dynamic Systems.
ACM Transactions on Software Engineering and Methodology (TOSEM) (2020)

Loading the retrieved model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\model1.pnml.
The retrieved modeln loaded in                              80 ms.
Loading the relevant model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\log1.xes.
The relevant model loaded in                                40 ms.
The boundedness of the retrieved model checked in           252 ms.
The boundedness of the relevant model checked in            0 ms.
Constructing automaton RET that encodes the retrieved model.
Automaton RET constructed in                                7 ms.
Automaton RET has 6 states and 7 transitions.
Constructing automaton REL that encodes the relevant model.
Automaton REL constructed in                                1 ms.
Automaton REL has 10 states and 14 transitions.

===================Calculating precision and recall=============================

The intersection INT of RET and REL constructed in          3 ms.
Automaton INT has 8 states and 8 transitions.

A largest eigenvalue of the adjacency matrix of REL is 1.389926793626072.
A largest eigenvalue of the adjacency matrix of REL computed in     101 ms.

A largest eigenvalue of the adjacency matrix of INT is 1.1147977995154013.
A largest eigenvalue of the adjacency matrix of INT computed in     2 ms.

Recall computed in                                                  103 ms.
Recall: 0.8020550468036464.
```

b) Partial Matching Precision and Recall Measures

To measure the partial matching precision value between the event log and model, use the option (-pmp) in the command line followed by the paths to log (-rel=<path>) and model files (-ret=<path>), as follows:

```
>java -jar jbpt-pm-entropia-1.5.jar -pmp -s -rel=log1.xes -ret=model1.pnml
```

Output Screen: (in the silent mode)

```
>java -jar jbpt-pm-entropia-1.5.jar -pmp -s -rel=log1.xes -ret=model1.pnml
0.867587367599503
```

When you replace the option (-pmp) with (-pmr), the tool measures the partial matching recall value.

```
>java -jar jbpt-pm-entropia-1.5.jar -pmr -rel=log1.xes -ret=model1.pnml
```

Note that the option (-s) is removed as the debug information and execution data are shown on the output screen.

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -pmr -rel=log1.xes -ret=model1.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:     http://www.pnml.org/
XES format:      https://xes-standard.org/
================================================================================
```

```
Computing eigenvalue-based precision/recall based on partial matching of traces.
The technique is described in:
Artem Polyvyanyy, Anna Kalenkova. Monotone Conformance Checking for Partially
Matching Designed and Observed Processes. ICPM 2019: 81-88.
https://doi.org/10.1109/ICPM.2019.00022

Loading the retrieved model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\model1.pnml.
The retrieved modeln loaded in                              79 ms.
Loading the relevant model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\log1.xes.
The relevant model loaded in                                39 ms.
The boundedness of the relevant model checked in            0 ms.

The boundedness of the retrieved model checked in           245 ms.

Constructing automaton RET that encodes the retrieved model.
Automaton RET constructed in                                6 ms.
Automaton RET has 6 states and 7 transitions.
Constructing automaton REL that encodes the relevant model.
Automaton REL constructed in                                1 ms.
Automaton REL has 10 states and 14 transitions.
Minimizing automaton REL.
Automaton REL was minimized in                              1 ms.
The minimized version of REL has 10 states and 14 transitions.
Determinizing the minimized version of REL.
The minimized version of REL was determinized in            1 ms.
The determinized version of the minimized version of REL has 11 states and 36 transitions.
Minimizing deterministic version of automaton REL.
The deterministic version of REL was minimized in           0 ms.
The minimized deterministic version of automaton REL has 9 states and 28 transitions.
Minimizing automaton RET.
Automaton RET was minimized in                              0 ms.
The minimized version of RET has 6 states and 7 transitions.
Determinizing the minimized version of RET.
The minimized version of RET was determinized in            1 ms.
The determinized version of the minimized version of RET has 5 states and 17 transitions.
Minimizing deterministic version of automaton RET.
The deterministic version of RET was minimized in           0 ms.
The minimized deterministic version of automaton RET has 3 states and 9 transitions.
The intersection INT of RET and REL constructed in          3 ms.
Automaton INT has 8 states and 23 transitions.

A largest eigenvalue of the adjacency matrix of REL is 3.9664313615514604.
A largest eigenvalue of the adjacency matrix of REL computed in     94 ms.

A largest eigenvalue of the adjacency matrix of INT is 3.899365208677346.
A largest eigenvalue of the adjacency matrix of INT computed in     2 ms.

Recall computed in                                                  96 ms.
Recall: 0.9830915634834327.
```

c) Controlled Partial Matching Precision and Recall Measures

In order to measure controlled partial matching precision and recall values, the options (`-cpmp`) and (`-cpmr`) are used, respectively. Both options should be followed by the paths to log (`-rel<path>`) and model files (`-ret<path>`) and (`-srel=<num>`) and (`-sret=<num>`) options to specify the number of allowed skips in relevant and retrieved traces.
The following command computes the controlled partial matching precision value between the event log and the model, where (`-cpmp`) is applied with a maximum of 3 allowed skipped actions in a trace described by each of the event log and model.

```
>java -jar jbpt-pm-entropia-1.5.jar -cpmp -srel=3 -sret=3 -rel=log1.xes -ret=model1.pnml
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -cpmp -srel=3 -sret=3 -rel=log1.xes -ret=model1.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
```

```
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

Computing eigenvalue-based precision/recall based on exact matching of traces
with the fixed number of event skips.
The technique is described in:
Anna Kalenkova, Artem Polyvyanyy. A Spectrum of Entropy-Based Precision and
Recall Measurements Between Partially Matching Designed and Observed Processes
International Conference on Service Oriented Computing (ICSOC) (2020)

Loading the retrieved model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\model1.pnml.
The retrieved modeln loaded in                            86 ms.
Loading the relevant model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\log1.xes.
The relevant model loaded in                              39 ms.
The boundedness of the retrieved model checked in         261 ms.

The boundedness of the relevant model checked in          0 ms.

Constructing automaton RET that encodes the retrieved model.
Automaton RET constructed in                              7 ms.
Automaton RET has 6 states and 7 transitions.
Constructing automaton REL that encodes the relevant model.
Automaton REL constructed in                              1 ms.
Automaton REL has 10 states and 14 transitions.

===================Calculating precision and recall=============================

Number of states in : REL is 17
Construction time of : REL is 4 ms.
Number of states in : RET is 18
Construction time of : RET is 2 ms.
The intersection INT of RET and REL constructed in        5 ms.
Automaton INT has 18 states and 48 transitions.

A largest eigenvalue of the adjacency matrix of RET is 3.089129894294137.
A largest eigenvalue of the adjacency matrix of RET computed in     8550 ms.

A largest eigenvalue of the adjacency matrix of INT is 2.895675670596542.
A largest eigenvalue of the adjacency matrix of INT computed in     16 ms.

Precision computed in                                               8566 ms.
Precision: 0.9373758209213151.
```

Similarly, in the following command, (`-cpmr`) is used to measure the controlled partial matching recall value, where the maximal number of allowed skipped actions in traces in the event log (`-srel`) and the model (`-sret=<num>`) are 2 and 3, respectively.

```
>java -jar jbpt-pm-entropia-1.5.jar -cpmr -srel=2 -sret=3 -rel=log1.xes -ret=model1.pnml
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -cpmr -srel=2 -sret=3 -rel=log1.xes -ret=model1.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

Computing eigenvalue-based precision/recall based on exact matching of traces
with the fixed number of event skips.
The technique is described in:
Anna Kalenkova, Artem Polyvyanyy. A Spectrum of Entropy-Based Precision and
Recall Measurements Between Partially Matching Designed and Observed Processes
International Conference on Service Oriented Computing (ICSOC) (2020)

Loading the retrieved model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\model1.pnml.
The retrieved modeln loaded in                            81 ms.
Loading the relevant model from C:\Users\hfk\git\codebase\jbpt-pm\entropia\examples\log1.xes.
```

```
The relevant model loaded in                              42 ms.
The boundedness of the retrieved model checked in         261 ms.

The boundedness of the relevant model checked in          0 ms.

Constructing automaton RET that encodes the retrieved model.
Automaton RET constructed in                              7 ms.
Automaton RET has 6 states and 7 transitions.
Constructing automaton REL that encodes the relevant model.
Automaton REL constructed in                              1 ms.
Automaton REL has 10 states and 14 transitions.

===================Calculating precision and recall=============================

Number of states in : REL is 17
Construction time of : REL is 4 ms.
Number of states in : RET is 18
Construction time of : RET is 2 ms.
The intersection INT of RET and REL constructed in        5 ms.
Automaton INT has 16 states and 39 transitions.

A largest eigenvalue of the adjacency matrix of REL is 2.824903783369232.
A largest eigenvalue of the adjacency matrix of REL computed in     82 ms.

A largest eigenvalue of the adjacency matrix of INT is 2.7830889874013653.
A largest eigenvalue of the adjacency matrix of INT computed in     2 ms.

Recall computed in                                                  84 ms.
Recall: 0.985197798164299.
```

Stochastic Conformance Checking Measures

a) Stochastic Precision and Recall Measures

To compute the stochastic precision value between the event log and the process model, use the option (-sp) as follows:

```
>java -jar jbpt-pm-entropia-1.5.jar -sp -rel=log2.xes -ret=model2.pnml
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -sp -rel=log2.xes -ret=model2.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

Computing precision based on stochastic approach.
The technique is described in:
Sander Leemans, Artem Polyvyanyy. Stochastic-aware conformance checking:
An entropy-based approach. CAISE 2020: 217-233.
https://doi.org/10.1007/978-3-030-49435-3_14

Stochastic precision: 0.9160724510254685.
```

Use the option (-sr) instead of (-sp) to get the stochastic recall value between the event log and the process model.

```
>java -jar jbpt-pm-entropia-1.5.jar -sr -rel=log2.xes -ret=model2.pnml
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -sr -rel=log2.xes -ret=model2.pnml
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
```

```
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

Computing precision based on stochastic approach.
The technique is described in:
Sander Leemans, Artem Polyvyanyy. Stochastic-aware conformance checking:
An entropy-based approach. CAISE 2020: 217-233.
https://doi.org/10.1007/978-3-030-49435-3_14

Stochastic recall: 1.0.
```

b) Entropic Relevance Measure

You can measure relevance of a stochastic process model to an event log using the option (`-r`), as the following command shows. Note that the retrieved model is specified to the stochastic process model, i.e. the retrieved model is a stochastic deterministic automaton in JSON format.

```
>java -jar jbpt-pm-entropia-1.5.jar -r -rel=log1.xes -ret=automaton.json
```

Output Screen:

```
>java -jar jbpt-pm-entropia-1.5.jar -r -rel=log1.xes -ret=automaton.json
================================================================================
Tool to compute quality measures for Process Mining and Process Querying ver. 1.5.
For support, please contact us at jbpt.project@gmail.com.
================================================================================
PNML format:    http://www.pnml.org/
XES format:     https://xes-standard.org/
================================================================================

Computing entropic relevance.
The technique is described in:
Artem Polyvyanyy, Alistair Moffat, Luciano Garcia-Bonuelos. An Entropic Relevance Measure for
Stochastic Conformance Checking in Process Mining. ICPM 2020.
Relevance: 11.653256727657693.
```

References

[1] Anna Kalenkova and Artem Polyvyanyy. A spectrum of entropy-based precision and recall measurements between partially matching designed and observed processes. In *ICSOC*, LNCS. Springer, 2020. In Press.

[2] Sander J. J. Leemans and Artem Polyvyanyy. Stochastic-aware conformance checking: An entropy-based approach. In *Advanced Information Systems Engineering*, volume 12127 of *LNCS*, pages 217–233. Springer, 2020.

[3] Artem Polyvyanyy and Anna Kalenkova. Monotone conformance checking for partially matching designed and observed processes. In *International Conference on Process Mining, ICPM 2019, Aachen, Germany, June 24-26, 2019*, pages 81–88. IEEE, jun 2019.

[4] Artem Polyvyanyy, Alistair Moffat, and Luciano García-Bañuelos. An entropic relevance measure for stochastic conformance checking in process mining. *CoRR*, abs/2007.09310, 2020.

[5] Artem Polyvyanyy, Andreas Solti, Matthias Weidlich, Claudio Di Ciccio, and Jan Mendling. Monotone precision and recall measures for comparing executions and specifications of dynamic systems. *ACM Trans. Softw. Eng. Methodol.*, 29(3), June 2020.