

# Pattern-level edit distance: Reproducibility & implementation notes

Maxime Raynal (LIG/MRIM & Nokia Bell Labs)

Marc-Olivier Buob (Nokia Bell Labs)

Georges Quénot (LIG/MRIM)

**NOKIA** Bell Labs



# Plan

- 1- Pattern clustering overview
- 2- Experimental pipeline
- 3- Reproducibility of experiments
- 4- Conclusion

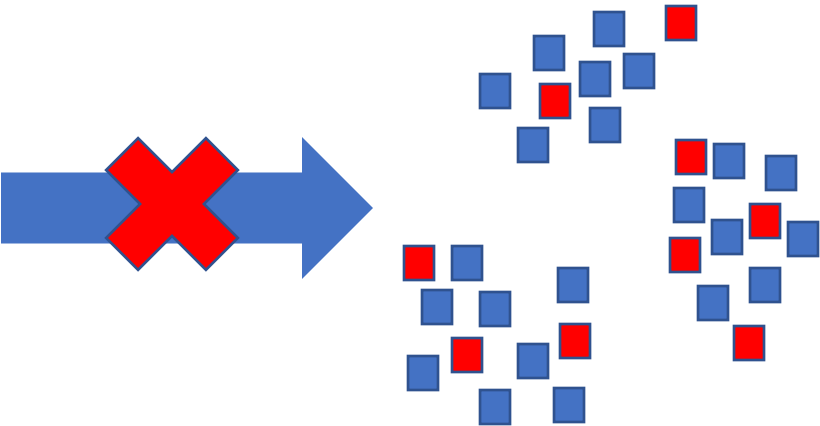
# Part 1: Pattern clustering overview

# Goal: partition a set of strings into homogeneous groups

- Use case: log clustering for automatic parsing
- Novelty: generalizes existing edit distances to operate at the pattern level

```
192.168.0.2 (EE) XYZ  
1.2.3.4 (II) ABC  
12/3/2004 10:12 ABCD  
12/6/2018 10:12 XYZ  
12 34 ABCD  
92 168 EEXYZ
```

Input file



```
1 2 3 4      ABCD  
1.2.3.4 (II) ABC  
12/3/2004 10:12 ABCD
```

```
192.168.0.2 (EE) XYZ  
92 168 EEXYZ  
12/6/2018 10:12 XYZ
```

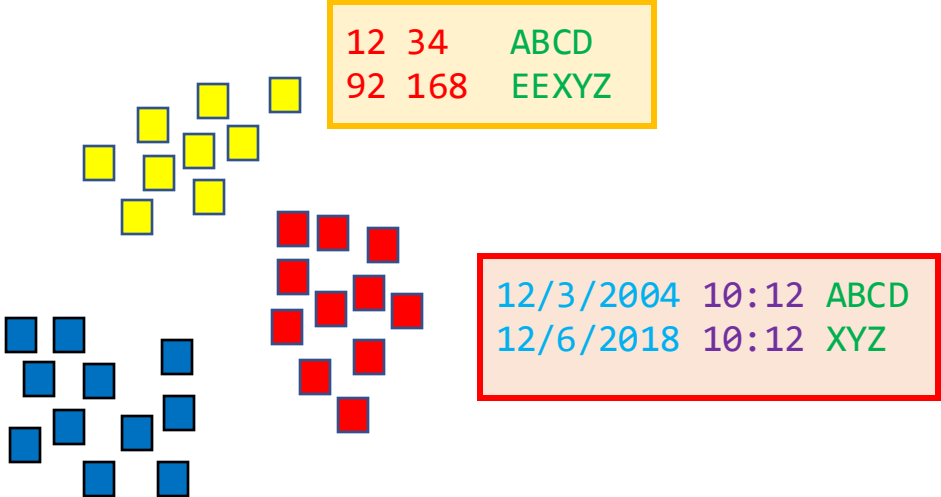
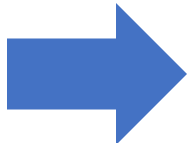
Output clusters

# Goal: partition a set of strings into homogeneous groups

- Use case: log clustering for automatic parsing
- Novelty: generalizes existing edit distances to operate at the **pattern level**

```
192.168.0.2 (EE) XYZ  
1.2.3.4 (II) ABC  
12/3/2004 10:12 ABCD  
12/6/2018 10:12 XYZ  
12 34 ABCD  
92 168 EEXYZ
```

Input file



```
12 34 ABCD  
92 168 EEXYZ
```

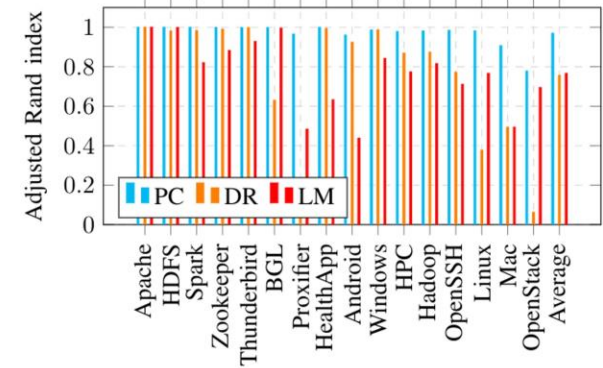
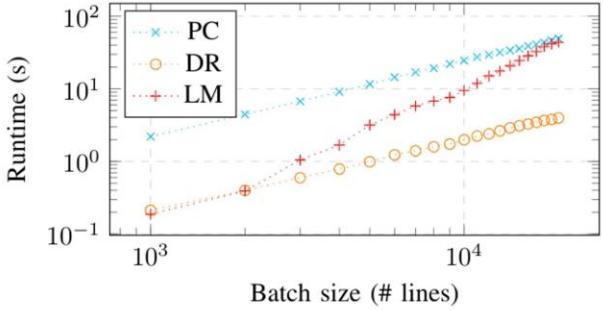
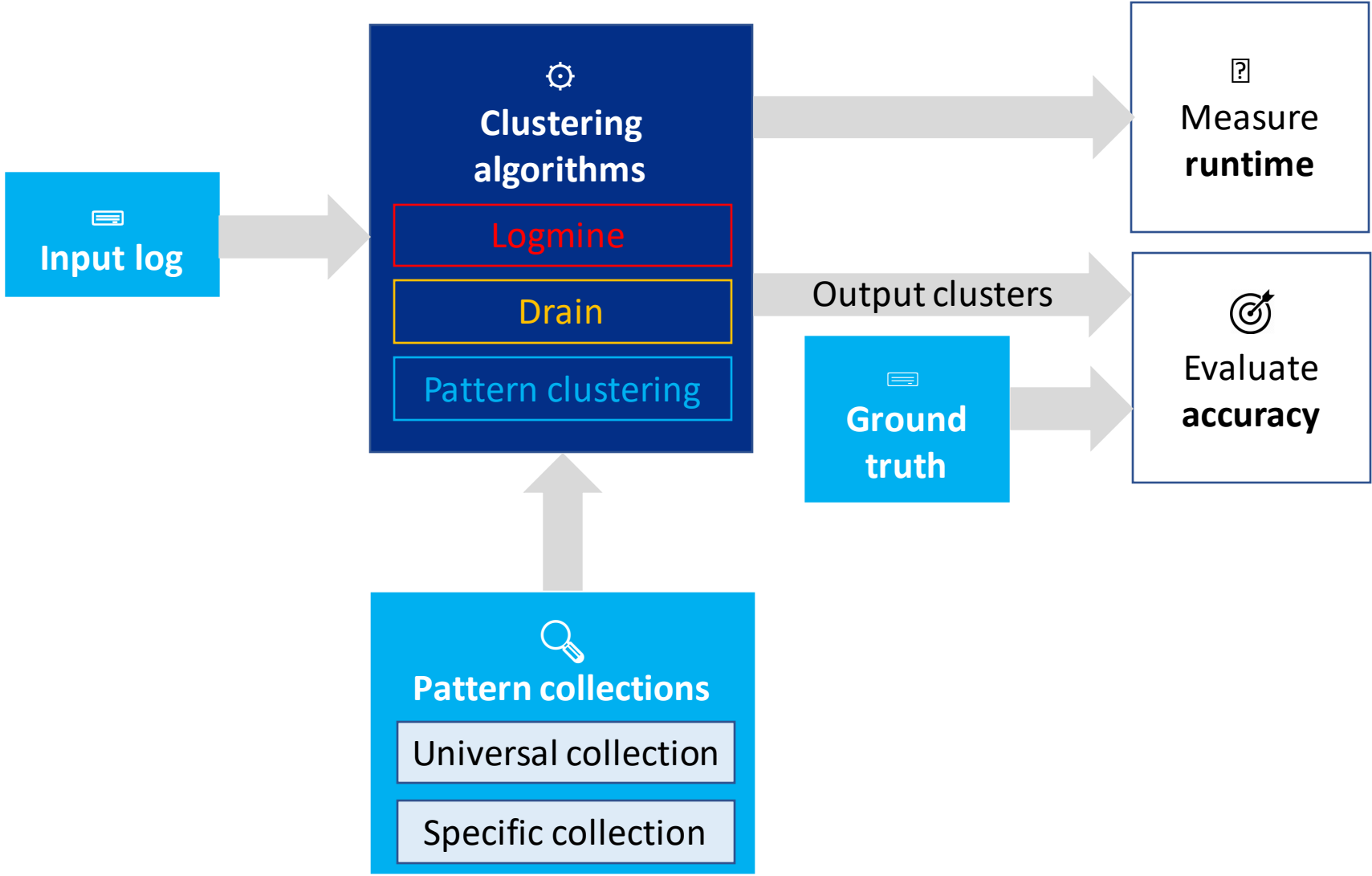
```
12/3/2004 10:12 ABCD  
12/6/2018 10:12 XYZ
```

```
1.2.3.4 (II) ABC  
192.168.0.2 (EE) XYZ
```

Output clusters

# Part 2: Experimental pipeline

# Our experiment pipeline in a nutshell



# Input logs

- We use the public **Loghub dataset** [1].
  - Zhu et al [2] used this dataset to benchmark several log clustering tools.
- It gathers **16 log files**, including:
  - Specific software (openSSH, proxifier, Apache)
  - Super-computers (HPC)
  - Distributed systems (Hadoop, HDFS)
  - Operating systems (Windows, Linux, Mac, Android)

[1] He, Shilin, et al. "Loghub: a large collection of system log datasets towards automated log analytics." *arXiv preprint arXiv:2008.06448* (2020). <https://github.com/logpai/loghub>

[2] Zhu, Jieming, et al. "Tools and benchmarks for automated log parsing." *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 2019.



# Ground truth

- The **original** ground truth is provided by the LogHub dataset
  - Each line identifies a cluster and the corresponding template
  - But it contains weird clusters. In the example below, E18, E19, E20 should be gathered:

```

E17,addNotification key=<*>|<*>|<*>|null|<*>
E18,"animateCollapsePanels:flags=<*>, force=false, delayed=false, mExpandedVisible=false"
E19,"animateCollapsePanels:flags=<*>, force=false, delayed=false, mExpandedVisible=true"
E20,"animateCollapsePanels:flags=<*>, force=true, delayed=true, mExpandedVisible=true"
E21,"Animating brightness: target=<*>, rate=<*>"

```

Original

- We corrected these inconsistencies to obtain a **fixed ground truth**

```

E17,addNotification key=<*>|<*>|<*>|null|<*>
E18,"animateCollapsePanels:flags=<*>, force=<*>, delayed=<*>, mExpandedVisible=<*>"
E21,"Animating brightness: target=<*>, rate=<*>"

```

Fixed

- The **original** and the **fixed** ground truths are publicly available on our repository and may be easily compared using **diff**.



# Clustering algorithms

- We compared 3 clustering algorithms :
  - **Pattern clustering (PC)** ⇒ our proposal
  - **Drain (DR)** ⇒ the **most accurate** clustering algorithm in literature [3]
  - **Logmine (LM)** ⇒ a **widely used** log clustering tool [4]
- *Remark: we slightly* modified **DR** and **LM** to get the needed information to plot our results.
  - **Minor** and **verifiable** patches (see our Github forks)

[3] He, Pinjia, et al. "Drain: An online log parsing approach with fixed depth tree." *2017 IEEE international conference on web services (ICWS)*. IEEE, 2017.

[4] Hamooni, Hossein, et al. "Logmine: Fast pattern recognition for log analytics." *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 2016.



# Pattern collections

- **Specific collection:** crafted by Zhu et al [2], one **dedicated** pattern collection for each log file.
  - Some patterns are **not intuitive**.
  - **Requires human intervention**  $\Rightarrow$  not suited for automatic parsing.

```
'Andriod': {  
  'log_file': 'Andriod/Andriod_2k.log',  
  'log_format': '<Date> <Time> <Pid> <Tid> <Level> <Component>: <Content>',  
  'regex': [r'(/[\\w-]+)+', r'([\\w-]+\\.){2,}[\\w-]+', r'\\b(\\-?\\+?\\d+)\\b|\\b0[Xx][a-fA-F\\d]+\\b|\\b[a-fA-F\\d]{4,}\\b'],  
  'st': 0.2,  
  'depth': 6  
},
```

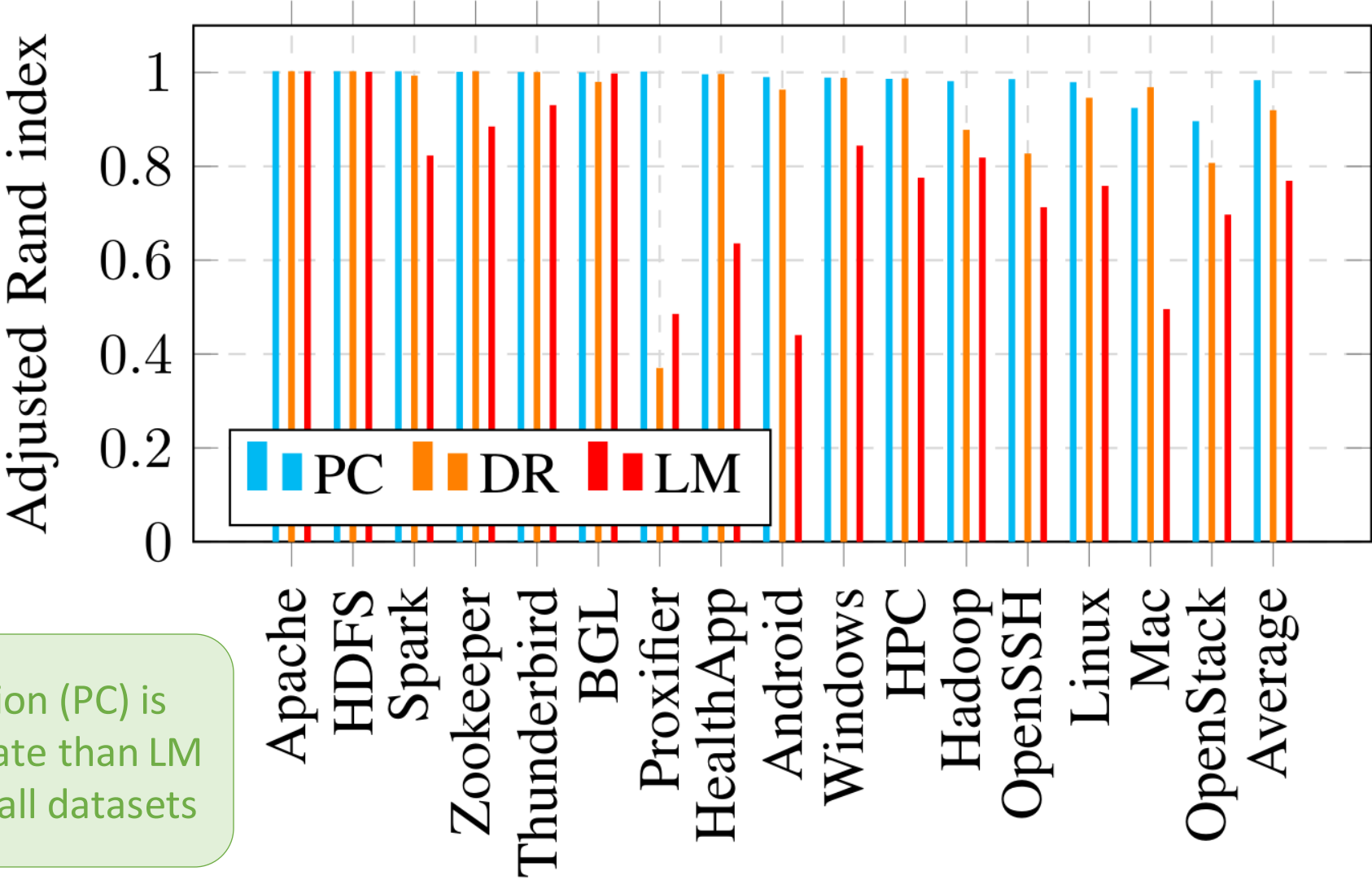
- **Universal collection:** we defined a **universal** pattern collection corresponding to a dozen standard data types (dates, numeric values, network addresses, paths) used for **every** log file.
  - **No human intervention**
  - Use to evaluate how **generalizes** each tool.



## Performance metric (accuracy)

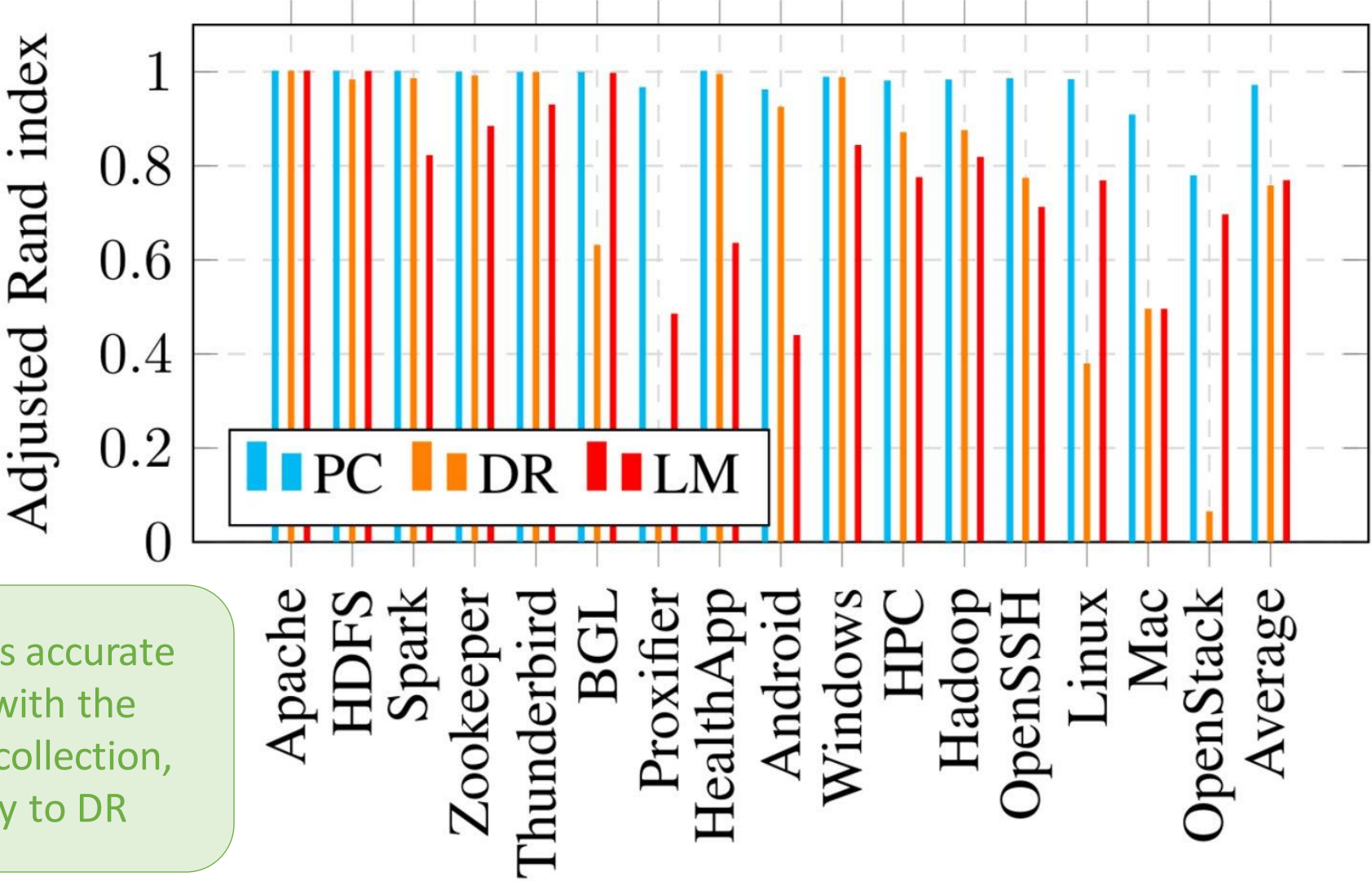
- Zhu et al survey [2] defines **parsing accuracy**.
  - Cluster  $C$  reward:  $|C|$  if correct,  $0$  otherwise.
  - **Highly sensitive** to:
    - Small changes in ground truth
    - Minor errors from the tool.
- We also considered the **adjusted Rand index**.
  - Compares pairwise assignments.
  - **Well-known** metric.
  - **Reflects better the clustering quality.**

# Accuracy: specific collection



Our solution (PC) is more accurate than LM and DR for all datasets

# Accuracy: universal collection



PC outputs accurate results with the universal collection, contrary to DR

# Part 3: Reproducibility of our experiments

# Overview of the code base

- **Open-source architecture:**
  - C++ and python code base.
  - Boost.python helps to translate python objects to C++ objects (e.g., python list to std::vector) and vice versa.
- **Optimized code:**
  - C++ core to accelerate the processing
  - Parallelization
- **User friendly:**
  - Python wrappers
  - Helpers to have fancy HTML displays in Jupyter notebook.

Front end



Wrappers,  
preprocessing



Core  
algorithms





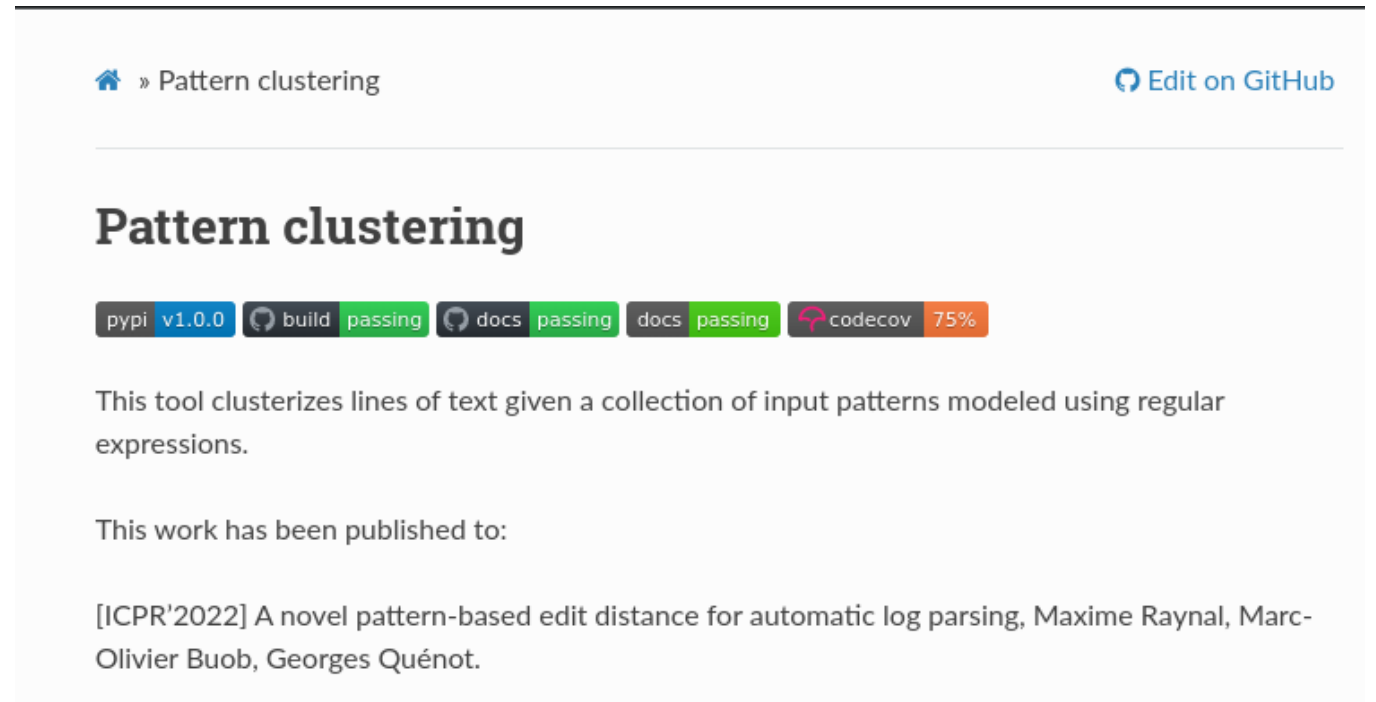
# Reproducibility

- **Goal:** make our experiments reproducible by anyone.
- The code is **public** (BSD license)
  - <https://github.com/nokia/pattern-clustering/>
- This repository provides the **complete** material to run our experiments:
  - The installation steps (including to install competitor algorithms)
  - The datasets
  - The experimental parameters



# Making sure others can run and reuse our code

- Users:
  - Installation tutorial.
  - **Open-source** and **standard** dependencies
- Developers: **continuous integration**
  - [Github repository](#)
  - [Pypi deployment \(src\)](#)
  - [Tests and coverage \(pytest, codecov\)](#)
  - [API documentation \(readthedocs\)](#)



The screenshot shows the PyPI page for the 'Pattern clustering' package. At the top, there is a breadcrumb trail '» Pattern clustering' and a link to 'Edit on GitHub'. The main heading is 'Pattern clustering'. Below the heading, there are several status badges: 'pypi v1.0.0', 'build passing', 'docs passing', 'docs passing', and 'codecov 75%'. The description states: 'This tool clusterizes lines of text given a collection of input patterns modeled using regular expressions.' Below the description, it says 'This work has been published to:' followed by a citation: '[ICPR'2022] A novel pattern-based edit distance for automatic log parsing. Maxime Raynal, Marc-Olivier Buob, Georges Quénot.'

# Conclusion

- A new log clustering tool
  - Code base **publicly available** on GitHub
  - For more details, see our RRPR & ICPR'2022 papers :-)
- **Experimental validation** of our proposal against logmine and Drain.
  - **Slower** than state of the art algorithm, but acceptable
  - **Most accurate** algorithm
  - **Works very well** with universal patterns
- We reproduced Zhu et al experiments and proposed an **enhanced setup**.
  - Universal pattern collection
  - Enhanced ground truth, available on GitHub
  - Adjusted Rand Index



# Questions ?

