

# TrackML challenge implementation in Codalab

---

*Victor Estrade*

*For the TrackML organizers*

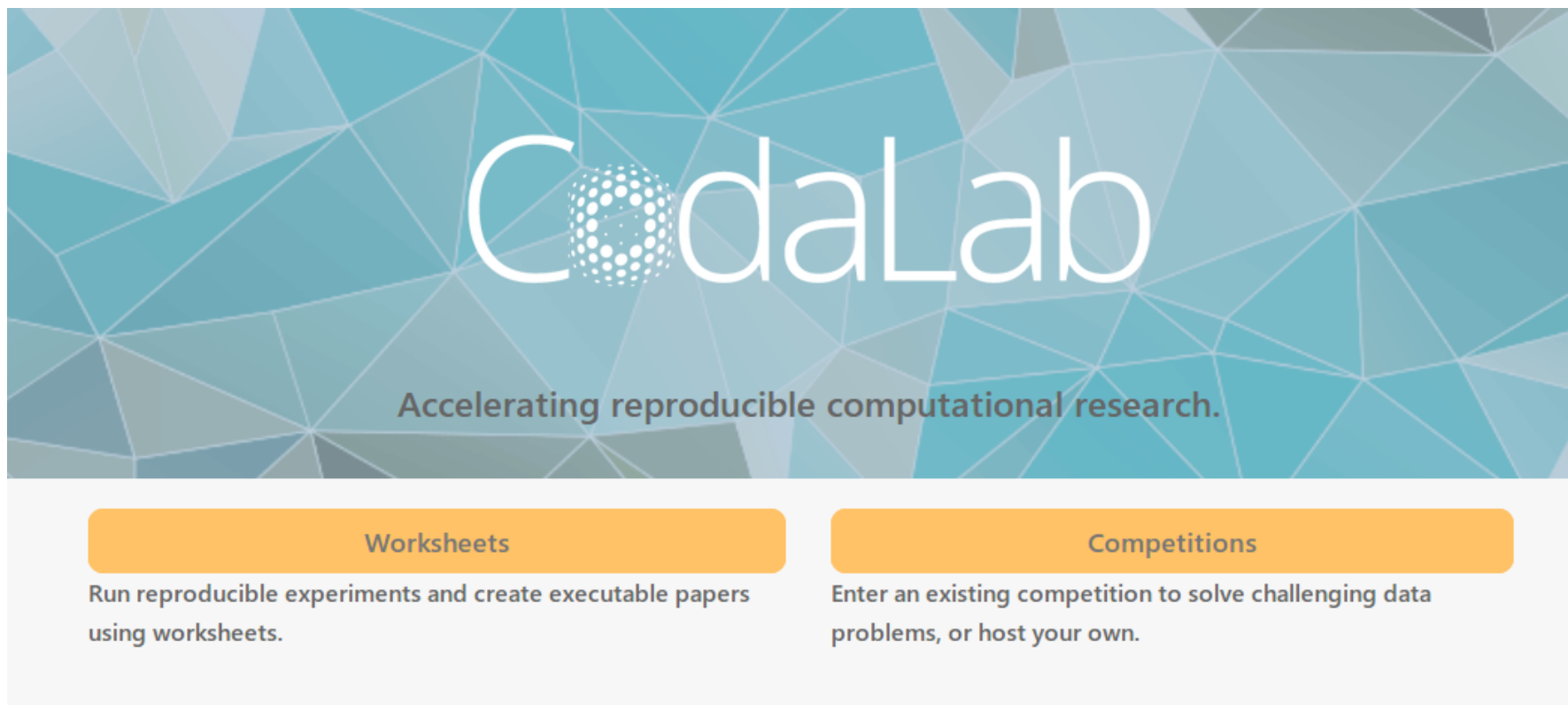
*TrackML Grand Finale, CERN, 02.07.2019*



université  
PARIS-SACLAY



# WHAT IS CODALAB ?

The image shows a promotional graphic for CodaLab. At the top, the word "Codalab" is written in a white, sans-serif font against a background of teal and light blue geometric shapes. The letter 'o' is replaced by a white sphere with a grid of dots. Below the logo, the tagline "Accelerating reproducible computational research." is centered in a smaller, dark grey font. At the bottom, there are two orange rounded rectangular buttons. The left button is labeled "Worksheets" and contains the text "Run reproducible experiments and create executable papers using worksheets." The right button is labeled "Competitions" and contains the text "Enter an existing competition to solve challenging data problems, or host your own."

**Codalab**

Accelerating reproducible computational research.

**Worksheets**

Run reproducible experiments and create executable papers using worksheets.

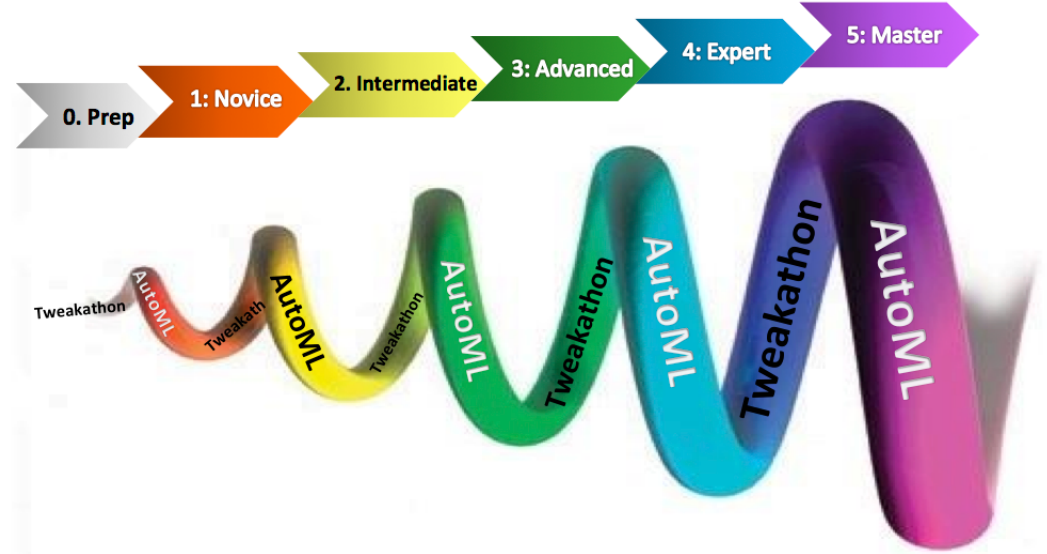
**Competitions**

Enter an existing competition to solve challenging data problems, or host your own.



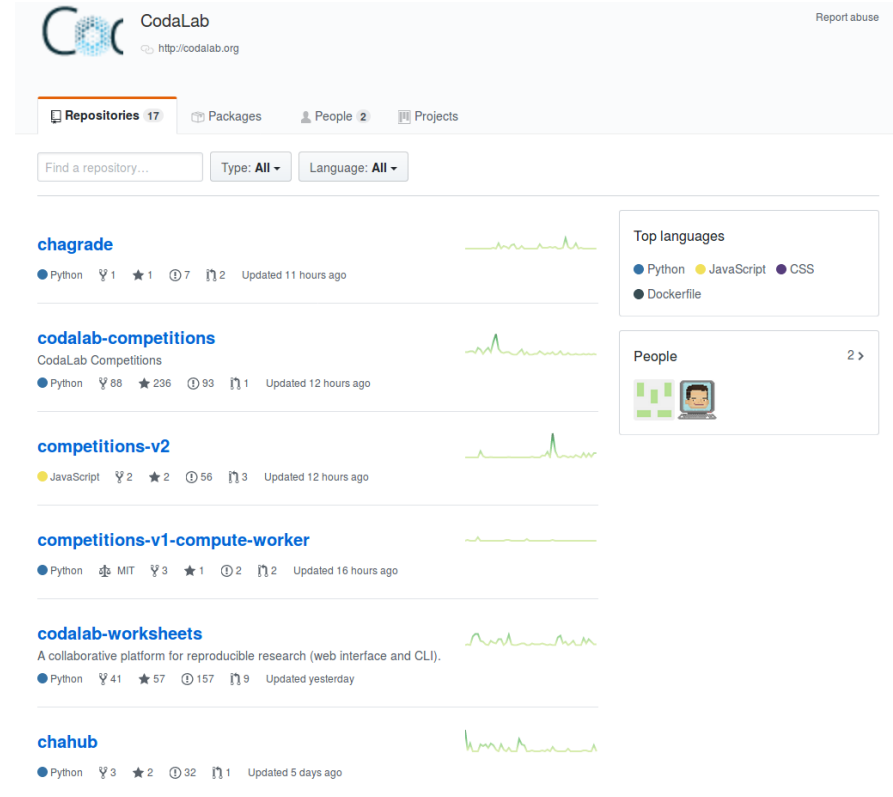
# CODALAB

- AutoML
- AutoDL
- Reinforcement learning
- Teaching
- Or anything else actually !



# DO IT YOURSELF

- Open source
- Toolbox
  - Competition
  - Workers
  - Competition examples
- Prediction or code submission



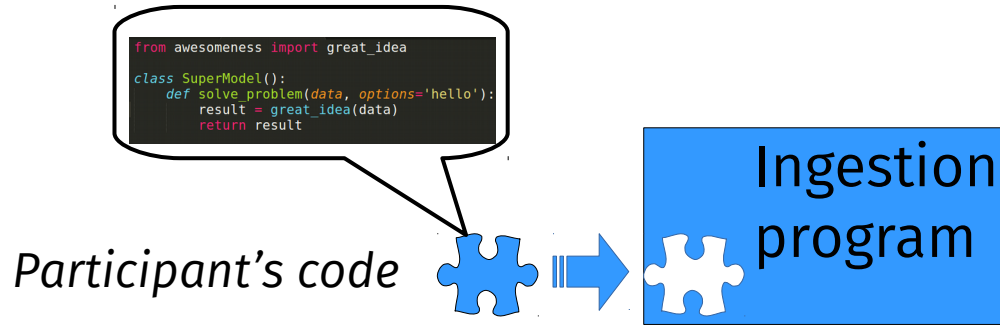
The screenshot shows the CodaLab website interface. At the top, the CodaLab logo and name are visible, along with the URL <http://codalab.org> and a "Report abuse" link. Below the header, there are navigation tabs for "Repositories 17", "Packages", "People 2", and "Projects". A search bar labeled "Find a repository..." is present, along with filters for "Type: All" and "Language: All". The main content area displays a list of repositories, each with a title, a small green line graph, and a list of languages. The repositories listed are:

- chagrade**: Python, 1 watch, 1 star, 7 issues, 2 forks, updated 11 hours ago.
- codalab-competitions**: CodaLab Competitions, Python, 88 watch, 236 stars, 93 issues, 1 fork, updated 12 hours ago.
- competitions-v2**: JavaScript, 2 watch, 2 stars, 56 issues, 3 forks, updated 12 hours ago.
- competitions-v1-compute-worker**: Python, MIT, 3 watch, 1 star, 2 issues, 2 forks, updated 16 hours ago.
- codalab-worksheets**: A collaborative platform for reproducible research (web interface and CLI), Python, 41 watch, 57 stars, 157 issues, 9 forks, updated yesterday.
- chahub**: Python, 3 watch, 2 stars, 32 issues, 1 fork, updated 5 days ago.

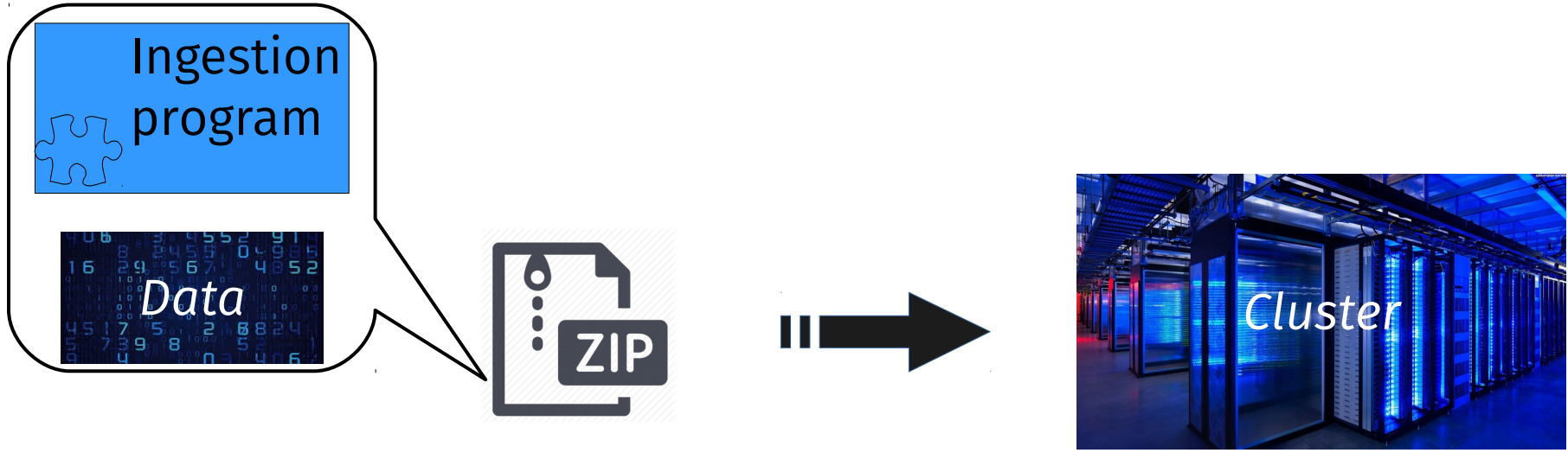
On the right side, there are two sidebars. The "Top languages" sidebar shows Python, JavaScript, and CSS. The "People" sidebar shows a profile picture and a "2 >" link.



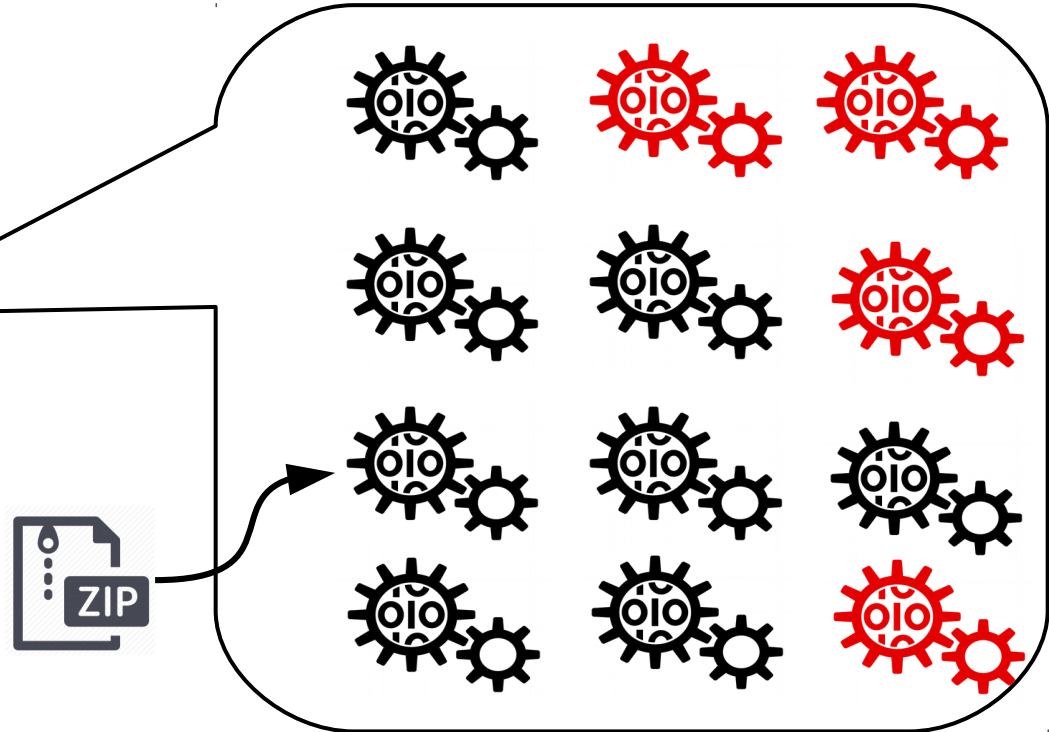
# CODE SUBMISSION PIPELINE



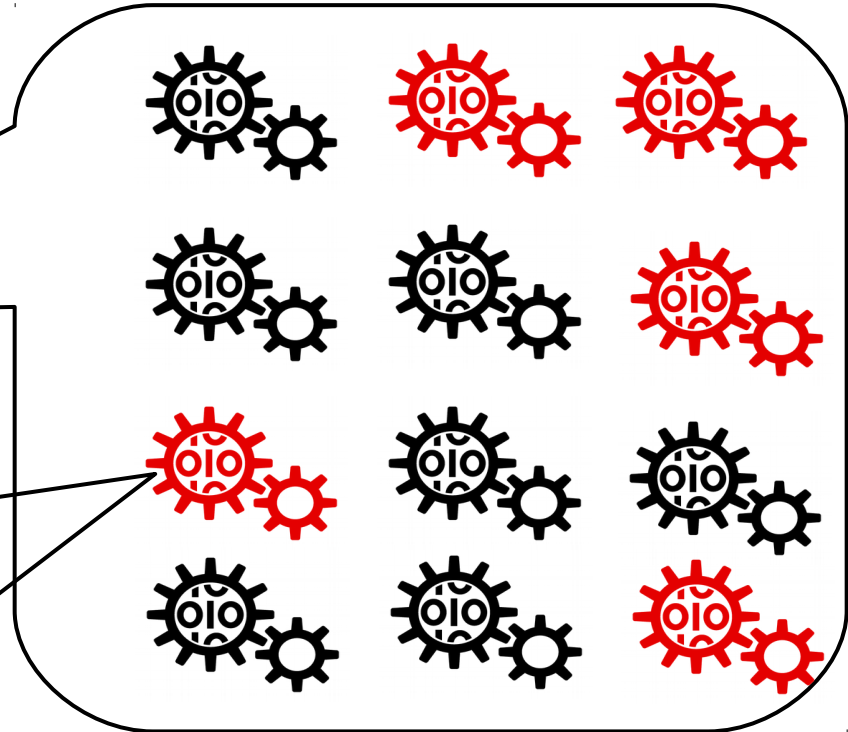
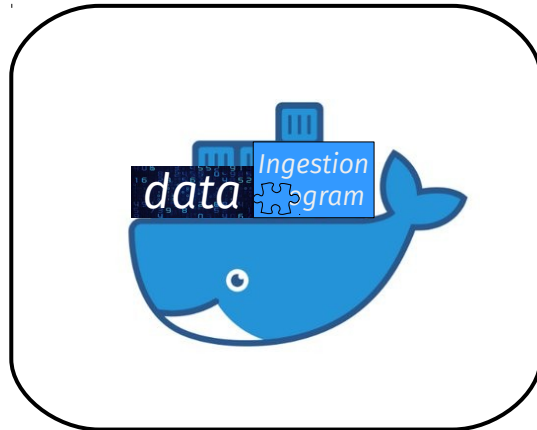
# CODE SUBMISSION PIPELINE



# CODE SUBMISSION PIPELINE

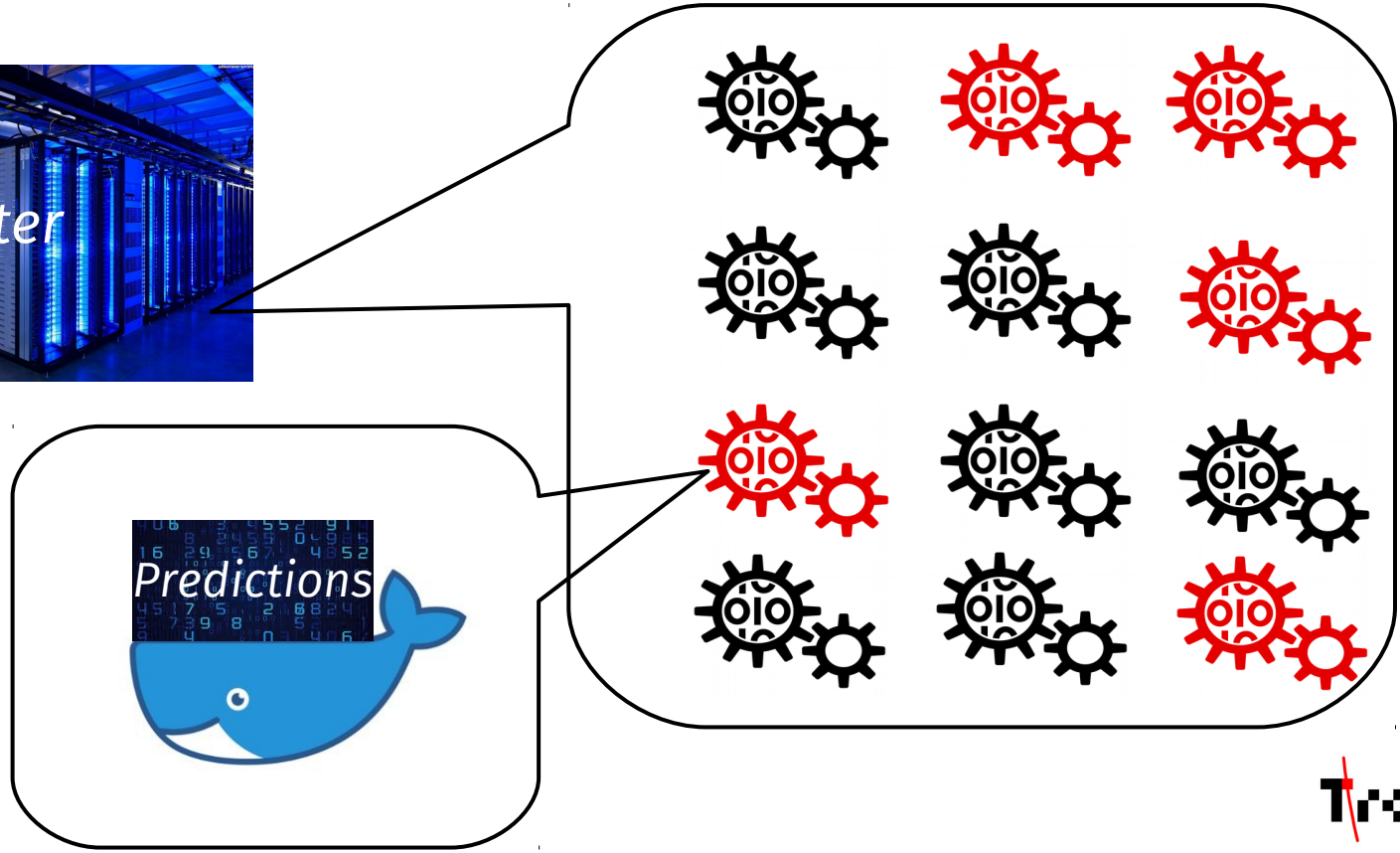


# CODE SUBMISSION PIPELINE





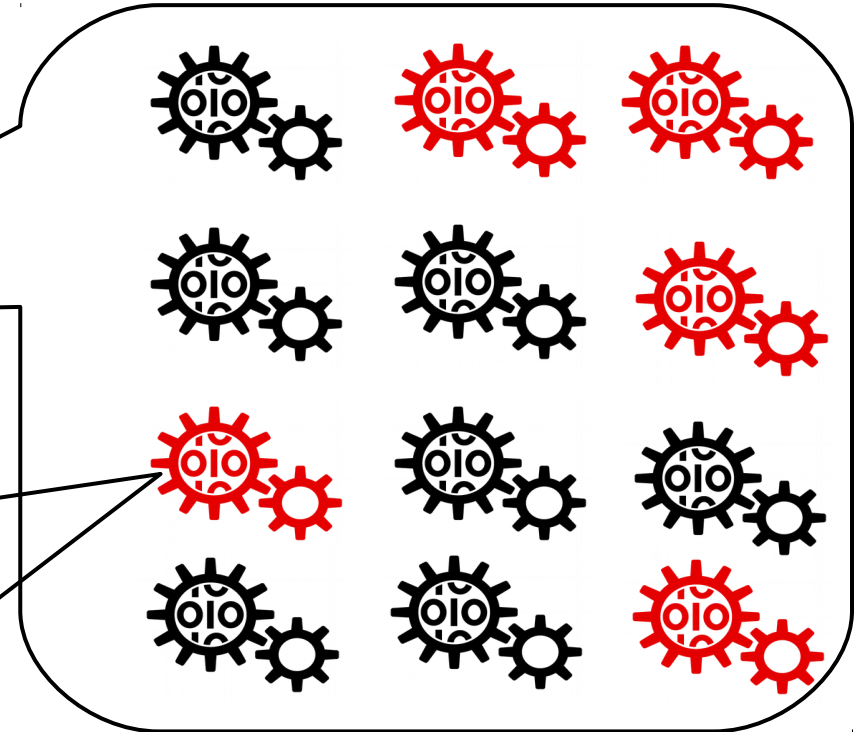
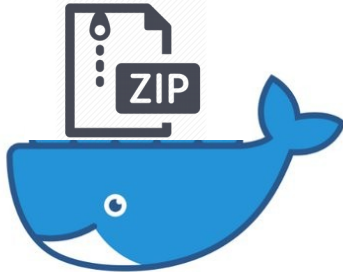
# CODE SUBMISSION PIPELINE



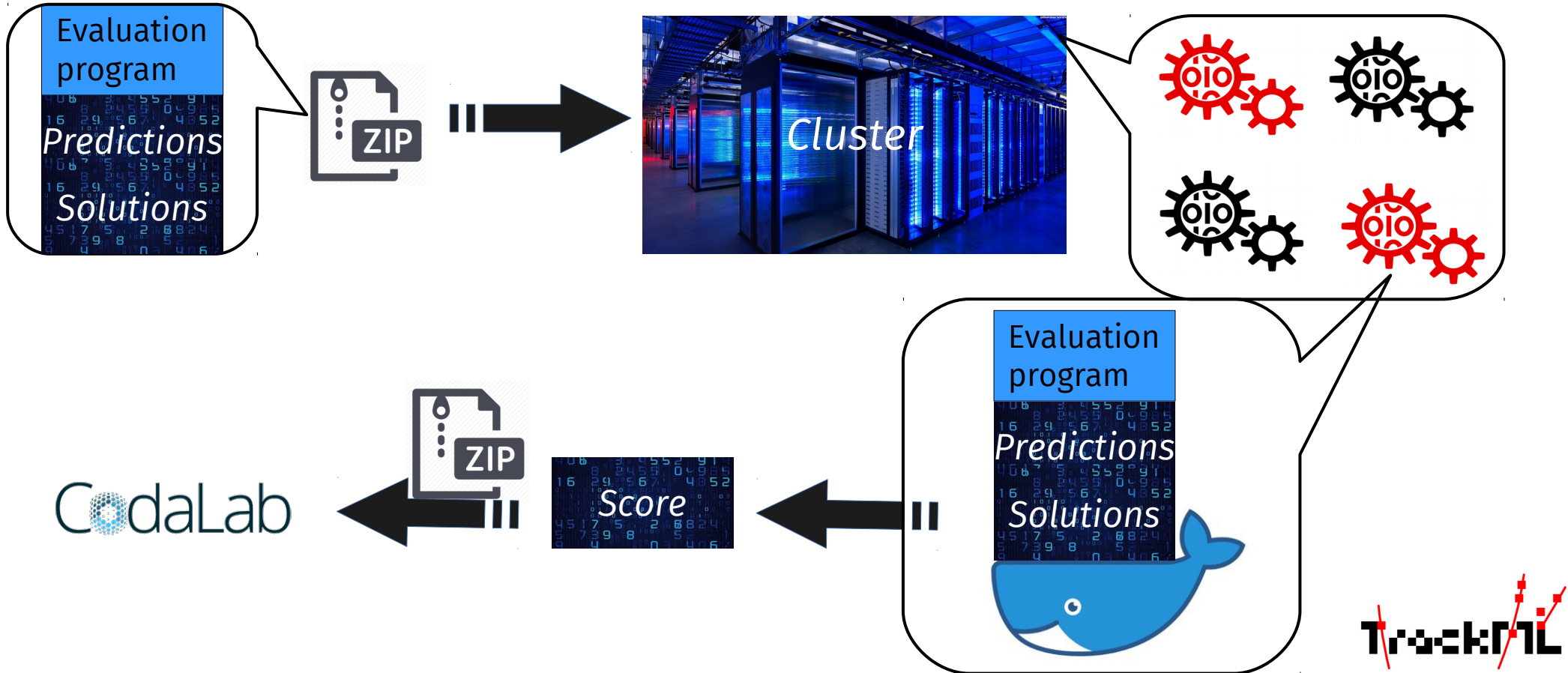
# CODE SUBMISSION PIPELINE



CodaLab



# CODE SUBMISSION PIPELINE



# CHALLENGE RECIPE

- Write a detailed pipeline on paper
- Choose docker image (choose available libraries)
- Write ingestion program
- Write evaluation program
- Test locally
- Test on the platform
- Write documentation and tutorial for the participants



# BIG DATA CONSTRAINTS

- Exemple
  - 40 000 000 0000 events per year / 200 000 CPUs = 157 sec / events
- Limited computation power
- Limited time
- Limited memory

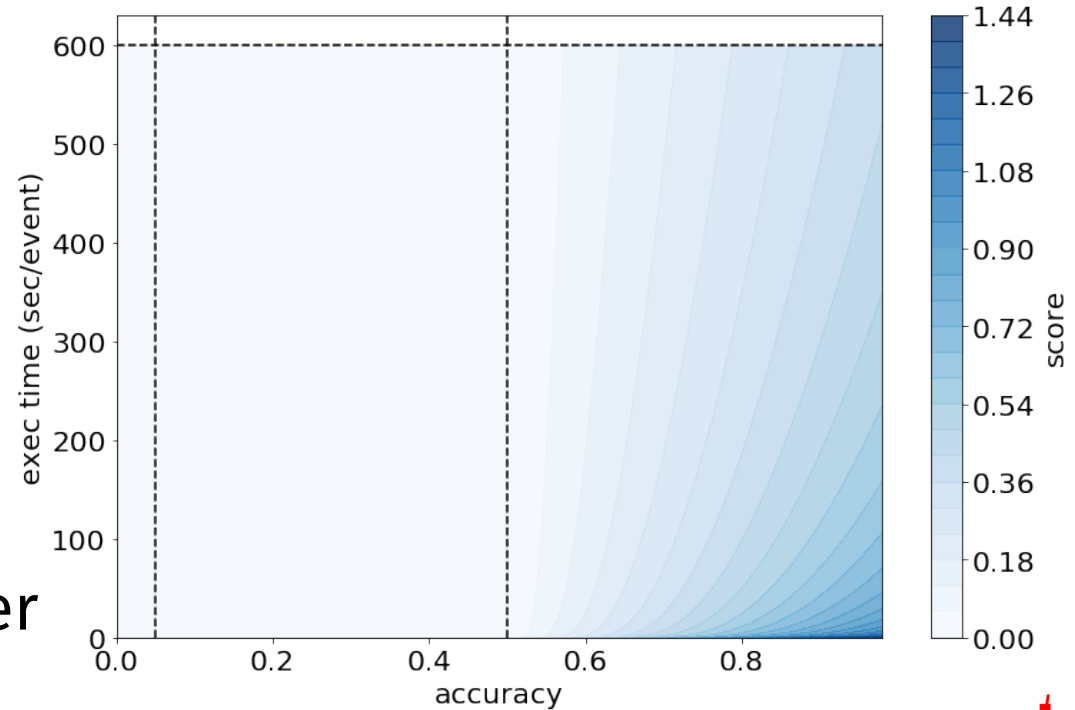


Requires code submission



# FAST ALGORITHM WANTED

- Objective : Throuput !
- Big incentives on speed
- 2 CPU (4 threads)
  - Multi processing is easy
  - Focus on multi threading
- 4GB memory
- Constraints enforce with docker



# MULTIPLE LANGUAGE

- Python is the standard in data science community
- But we want throughput !
- C/C++
- R is also quite popular and was used during phase 1
- 1st idea : starting a sub-process

program input output



# TUTORIAL AND WORKING EXAMPLES

- Starting-kit is key for dev and participants
  - With good documentation & clean code !
- Enable participant to run the challenge at home
  - with docker and a simplified ingestion program
  - Avoid congestion on the cluster
- Beta testing (multiple times if needed)
  - Hackathon, workshop, internal challenge (with debug flag ON)
  - Get outside point of view to find missing information
- **Decision** : run private debug version of the challenge



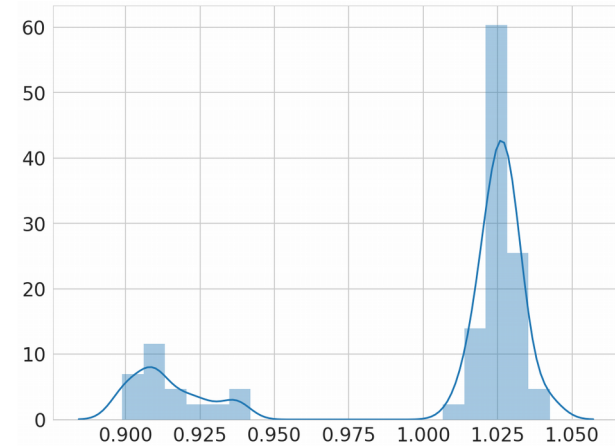
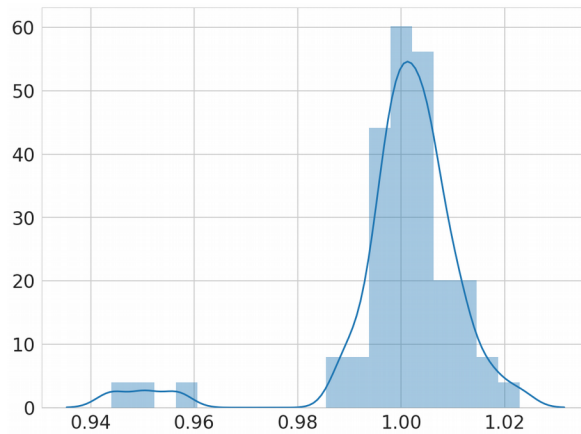
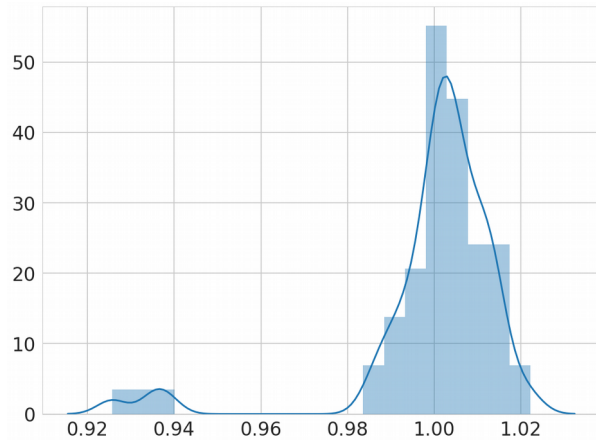


# REPRODUCIBILITY

- The public score is the reward / keeps participant motivated
- Accuracy score is deterministic but speed measurement is noisy
- 1 event can hold in the L3 cache
- 1st time we need to measure exec time with precision under 1 sec
- I/O issues
  - Need to measure time without the I/O.
- **Decision** : bind Python with the other languages



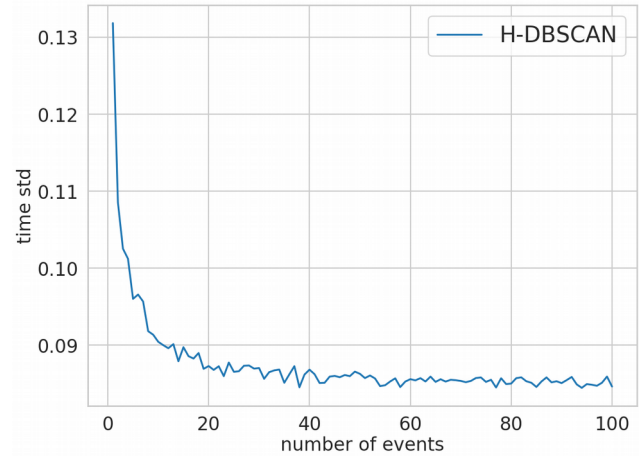
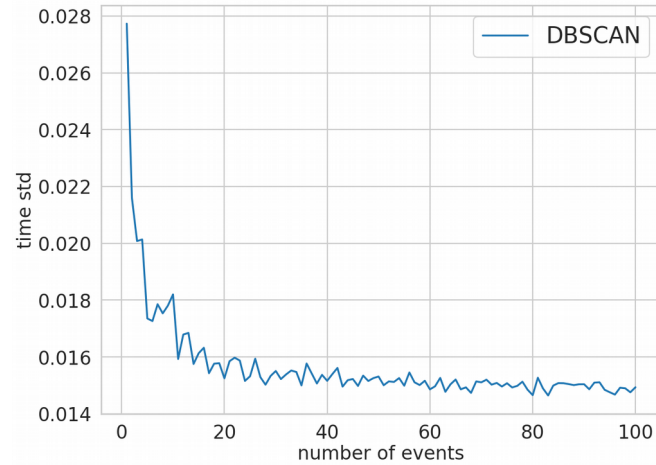
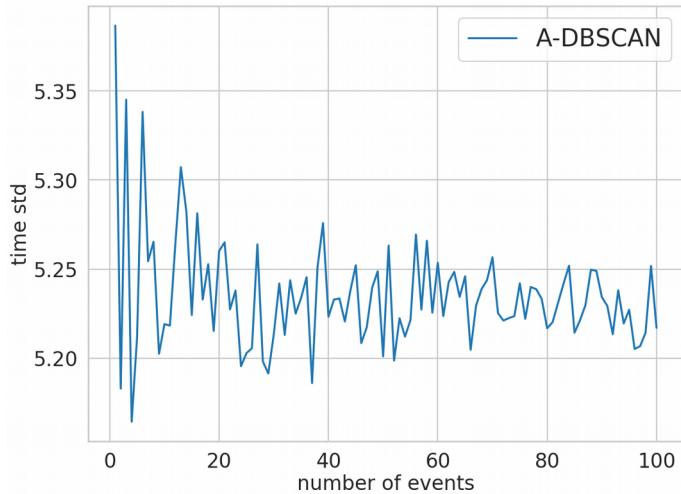
# LUCKY & UNLUCKY RUNS



- 10% faster is good ! It can increase the score by 1% (1 sec) to 3.6% (4 min)
- **Decision** : be carefull with the private leaderboard



# NUMBER OF EVENT



- **Decision : 50 events**



# STRESS TEST

- No fire, no shutdown, no crashed submissions 😊
- But some submissions took a while to run (2 or 3 times slower) 😞
- **Decision** : early reject bad submissions ( $< 5\%$  accuracy,  $> 600$  sec)
- **Decision** : for private leaderboard
  - multiple runs
  - on an « idle » machine

# CHEATING ?

- Cheating on accuracy is hard (no access to solution data)
- Cheating on time ?
- Tradeoff : Avoid leakage and allow logs to help debug
- Use internet to run on external computers
- Attack through Python or Docker security vulnerabilities



# UNEXPECTED

- Binary C++ code with python source code
- Solution that runs so fast !



# CONCLUSION

## What did we learn ?

- Measuring execution time is not trivial
- Beta test your challenge to get valuable feedback

## Take home message

- Building challenge has never been easier
  - And will become even easier in the future
- Contact Isabelle Guyon ([guyon@chalearn.org](mailto:guyon@chalearn.org)) or me ([estrade@lri.fr](mailto:estrade@lri.fr))



# QUESTION ?

*Thank you for your attention !*

