Reproducible Biometrics Evaluation and Testing with the BEAT Platform

André Anjos, Philip Abbet and Sébastien Marcel

April, 1st. 2014
How many times?

“I think you should be more explicit here in step two.”
Crossed a publication and openly decided to **ignore it because it would be too hard to apply** those doubtful results on your research?
Worked day and night to incorporate some results on your own work but:

- There were **untold parameters** that needed adjustment and you couldn’t get hold of them?
- Realized the proposed algorithm worked only on the specific data shown at the original paper?
- Realized that something did not quite add up in the end?
Had a **new student to take over** the work from another student that left and had to start from scratch - *months* into programming to make things work again?
Would have liked to **replay to someone about your work** , but you couldn’t really remember all details when you first made it work? Or you **could not make it work** at all?
Enter “Reproducible Research” (RR)\textsuperscript{1}

One term that aggregates work comprising of:

- a \textit{paper}, that describe your work in all relevant details
- \textit{code} to reproduce all results
- \textit{data} required to reproduce the results
- \textit{instructions}, on how to apply the \textit{code} on the \textit{data} to replicate the results on the \textit{paper}.

\textsuperscript{1}http://reproducibleresearch.net
Levels of Reproducibility\textsuperscript{2}

With respect to an independent researcher:

0. Irreproducible
1. Apparently unable to reproduce
2. Reproducible, with extreme effort (> 1 month)
3. Reproducible, with considerable effort (> 1 week)
4. Easily reproducible (∼ 15 min.), but requires proprietary software (e.g. Matlab)
5. Easily reproducible (∼ 15 min.), only free software
6. \textbf{Easily reproducible (∼ seconds), only requires a web-browser}

\textsuperscript{2}Reproducible Research in Signal Processing: What, why and how, Vandewalle, Kovacevic and Vetterli, 2012
Taking RR to the next level: the BEAT platform

The BEAT (Biometrics Evaluation and Testing) platform provides easy online access to experimentation and testing for Biometrics. You define what data and modules you would like to use, we make sure the system runs and provides you with a result. Data from different experiments can be easily compared and searched.
Taking RR to the next level: the BEAT platform

- Web platform: always **accessible**, no need to install extra software
- Intuitive: **graphically** connect blocks to run experiments
- Social: **engagement** gets you **more processing power**
- Productive: search the state-of-the-art by any filtering criteria
- Private:
  - No need to handle large-scale databases
  - Can run on *un*-distributable data (e.g. forensic databases)
- Assurance
  - **fair (reproducible) evaluations** of algorithms
  - **online certifications** for all produced results
- Free: build on **open-source** software and standards
BEAT Platform: Original Concept

Back-end (Computing Cloud)
- Storage
  - Databases
  - Uploaded Algorithms
  - User Results

Interconnect

Front-end
- My results
- Scheduler
- Ranking
- Uploader
- Downloads
- Documentation

Front-end Server

BEAT :: My results
Algorithm XYZ (Classifier)

* Privacy controls for results included!
Architectural Choice

Internet

Web front-end

Web server

Database

Algorithm repository

Scheduler

Toolchain

Data cache

Data

Data

Data

Computation back-end
A key concept in experimentation is the idea of *Toolchains*.
Blocks

Toolchains are composed of interconnections of Blocks

- **Inputs:** Each one accepts one data format
- **Outputs:** Each one produces one data format
- **Configuration:**
  - Parameter #1
  - Parameter #2
  - ...
  - Parameter #N

---

**Block**

- **Input API**
- **Algorithm**
- **Output API**
- **Storage**
  - Data
  - Data
  - Data
Blocks: Features

- Blocks can run **arbitrary** code
  - Potential to implement back-ends to support compiled code, any scripting language
  - We picked Python as a default back-end to start the project
  - The platform itself, is also written in Python
- Blocks typically have inputs and outputs
- Data transmitted from block to block is formally defined (*Data Formats*)
- Database blocks are special - they only have outputs, provided by administrators of the platform
- Result blocks don’t output to any other block
Confidentiality

BEAT is designed as an *opt-in* platform

- All your actions and results are kept private until you choose to change visibility
- Once you change the visibility of any item, associated items are *frozen* so your results are kept reproducible
- If anything changes on the underlying platform (OS, packages, toolchains, databases), your results are *outdated* - but still valid
Toolchain Example: trivial Eigen-faces
Simple: no evaluation (test) set, threshold \textit{a posteriori}
Toolchain Example: trivial Eigen-faces

Translation as a BEAT toolchain

- All relations are explicit: data and algorithms
- *Expected* database is divided into three-blocks leading to the training data, and validation data (split into gallery and probing sets)
- Notice that a Toolchain only defines the connections and data types into and out of Blocks
- Check for yourself: https://www.beat-eu.org/platform/toolchains/bob/eigenface/
Caching

- BEAT keeps track of data transmitted through all stages of the processing
- It caches the data in a large disk array (∼10 Tb)
- The cache is invalidated automatically when things change:
  - Operating System or installed packages are updated
  - Toolchain changes
  - Database version changes
- One cached item is valid for a specific combination of all of the above
- If a cached item is available, it is used to speed-up processing.
Proof of Concept: DCT+GMM Face Recognition

(Click to launch external video)
Attestation ("Certification")

An attestation mechanism is available in the platform.

- Allow 3rd. party verification of results obtained with a given configuration
- Allow for a scientific review process to take place in confidentiality
Forking and Modification

- One of the main mechanisms for *sharing* is the ability to *fork* a toolchain.
- By *forking*, you get a new toolchain that shares all properties of a given toolchain, except the ownership.
- You can modify only your own forks for toolchains.
One of the jobs of BEAT is to keep track of details for all experiments posted.
Omni-search

You can use the Omni-search bar (on the top) to type-in query strings.
Open for pre-registration (now)

Pre-registered users will benefit from early platform access when the service becomes available.

https://www.beat-eu.org/platform/preregister/

- In-browser graphical toolchain and code editors
- More social features: notifications, reputation system, discussion forum
- Better search result categorization. Output search results into data to re-use on your publications
- Full parallelization support
- Initial Hardware commissioning (end of April/2014):
  - 120 dedicated processing cores with 8 Gb RAM per core
  - 20 Tb of cache
  - 10 Gb/s link between cache and processing nodes
Acknowledgement for support

**European Commission**

**Biometrics Evaluation and Testing (BEAT)**
http://www.beat-eu.org

**Swiss State of Wallis**

**Swiss center for biometrics research and testing**
http://www.biometrics-center.ch