Writing **reproducible** and **scalable** bioinformatics pipelines using nextflow, docker and github

Matthieu Foll
Nov. 12th 2015

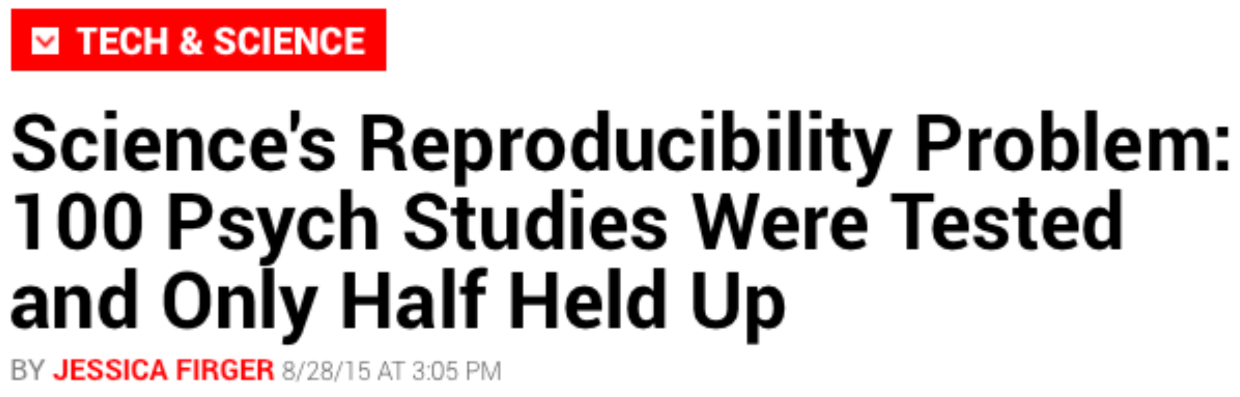
International Agency for Research on Cancer
Lyon, France

Reproducibility

- Main principle of scientific method
- Trending topic in science and mass media
- Easy for computational research? Not really:
  - Software used: versions, availability, open-source?
  - Custom scripts: version, interpreter/compiler?
  - Pipeline “orchestration”
  - Operating system

International Agency for Research on Cancer

World Health Organization



TECH & SCIENCE

Science's Reproducibility Problem: 100 Psych Studies Were Tested and Only Half Held Up

BY JESSICA FIRGER 8/28/15 AT 3:05 PM

Reproducibility and the conduct of research

Issues

🟥 **Data dredging**
Also known as p-hacking, this involves repeatedly searching a dataset or trying alternative analyses until a 'significant' result is found.

🟩 **Omitting null results**
When scientists or journals decide not to publish studies unless results are statistically significant.

🟦 **Underpowered study**
Statistical power is the ability of an analysis to detect an effect, if the effect exists – an underpowered study is too small to reliably indicate whether or not an effect exists.

⬛ **Errors**
Technical errors may exist within a study, such as misidentified reagents or computational errors.

🟧 **Underspecified methods**
A study may be very robust, but its methods not shared with other scientists in enough detail, so others cannot precisely replicate it.

🟨 **Weak experimental design**
A study may have one or more methodological flaws that mean it is unlikely to produce reliable or valid results.

Possible strategies

Open data Openly sharing results and the underlying data with other scientists.	🟥 🟩
Pre-registration Publicly registering the protocol before a study is conducted.	🟥 🟩 🟦 🟨 🟧
Collaboration Working with other research groups, both formally and informally.	🟦 🟨 🟧
Automation Finding technological ways of standardising practices, thereby reducing the opportunity for human error.	🟧 ⬛
Open methods Publicly publishing the detail of a study protocol.	🟨 🟧 ⬛
Post-publication review Continuing discussion of a study in a public forum after it has been published (most are reviewed before publication).	🟨 🟧
Reporting guidelines Guidelines and checklists that help researchers meet certain criteria when publishing studies.	🟨 🟧 ⬛

Improving reproducibility will ensure that research is as efficient and productive as possible. This figure summarises aspects of the conduct of research that can cause irreproducible results, and potential strategies for counteracting poor practice in these areas. Overarching factors can further contribute to the causes of irreproducibility, but can also drive the implementation of specific measures to address these causes. The culture and environment in which research takes place is an important 'top-down' overarching factor. From a 'bottom-up' perspective, continuing education and training for researchers can raise awareness and disseminate good practice.

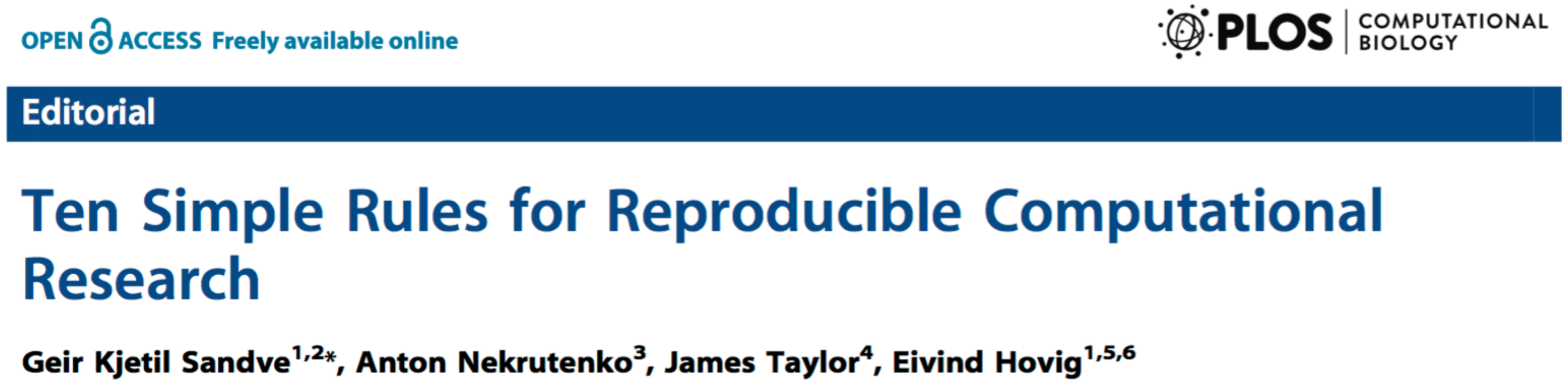
Figure taken from the report of the symposium, 'Reproducibility and reliability of biomedical research', organised by the Academy of Medical Sciences, BBSRC, MRC and Wellcome Trust in April 2015. The full report is available from http://www.acmedsci.ac.uk/researchreproducibility.

Reproducibility: how?

- For every result, keep track of how it was produced
- Avoid manual data manipulation steps
- Version control all custom scripts (*avoid custom scripts?*)
- Archive the exact versions of all programs used
- Record all intermediate results
- Provide public access to scripts, runs, and results

International Agency for Research on Cancer

World Health Organization



OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Editorial

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve[1,2*], Anton Nekrutenko[3], James Taylor[4], Eivind Hovig[1,5,6]

Useful reproducibility

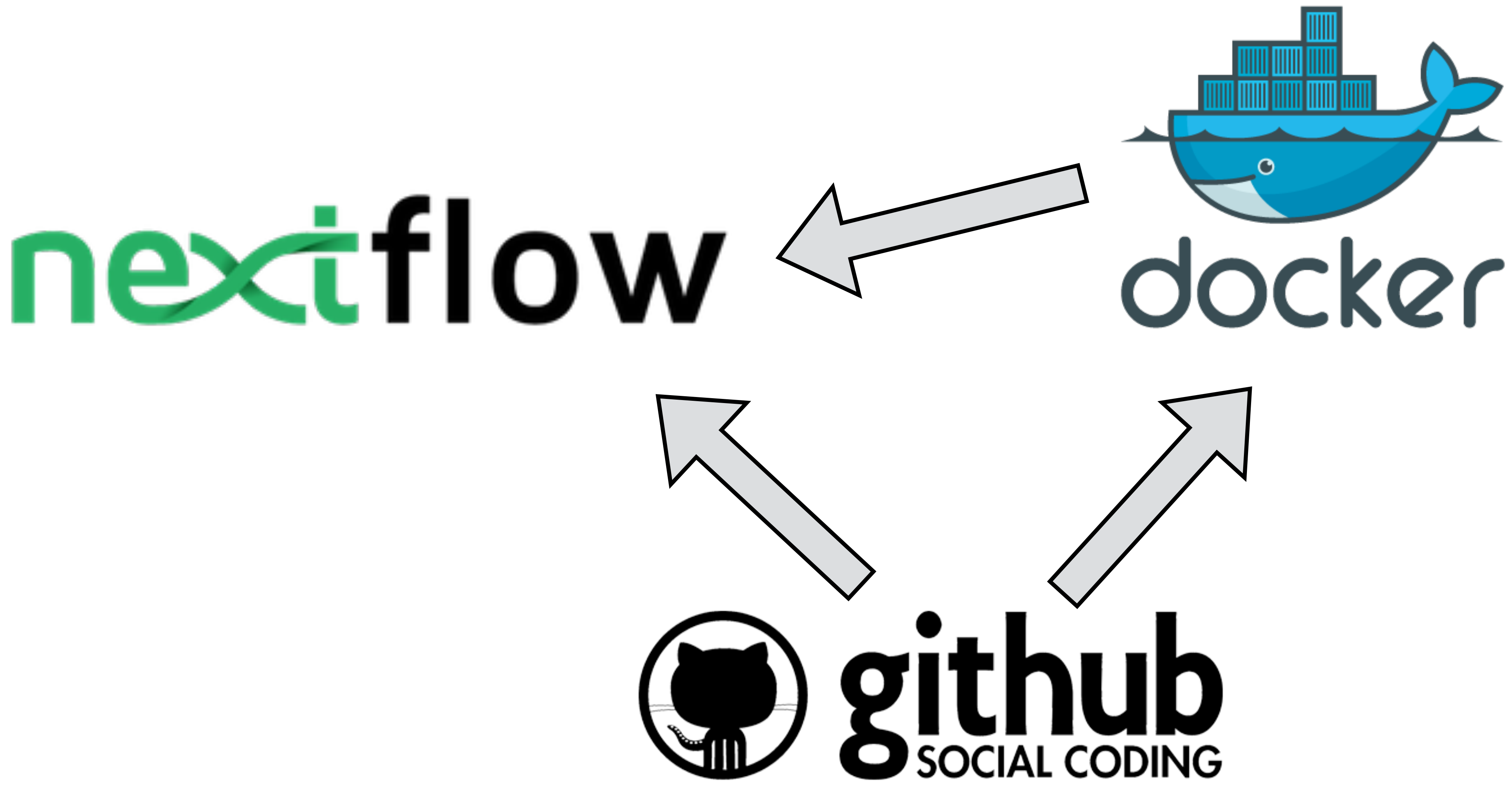
- Virtual Machine can make analyses reproducible, but not necessarily in a useful way (black box)
    - Put all data, scripts, software in a virtual machine, push a button and you get the same results
    - Problem: you can't install a VM for each software you want to use

"It's not really for the benefit of other people. Experience shows the principal beneficiary of reproducible research is you the author yourself"
Jon Claerbout

Distributing a reproducible pipeline

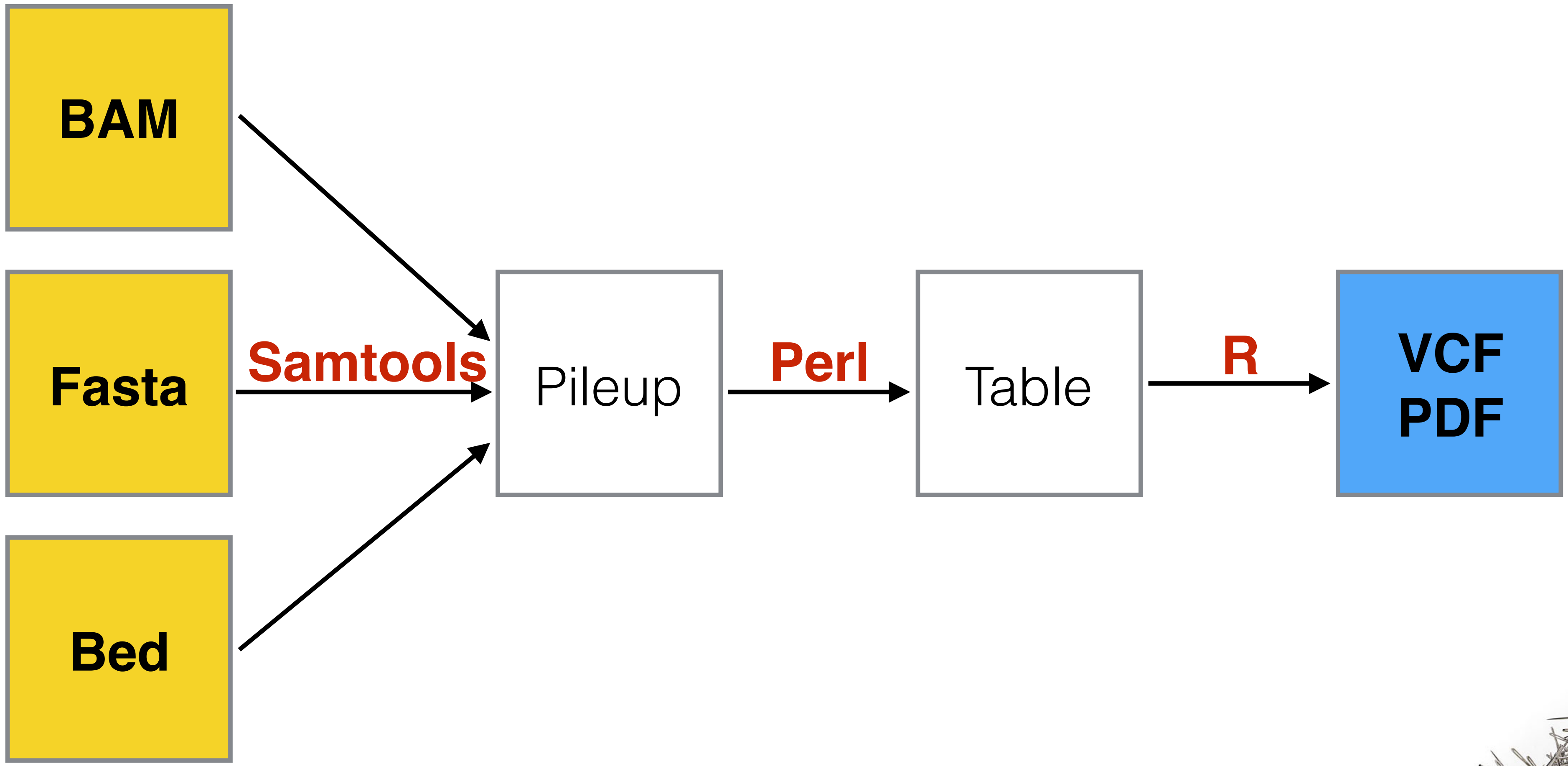
- Easy to install all the necessary tools, can choose specific versions (the ones used in the published paper)
- **Scalable**: usable without modifying the code on a laptop or a HPC Cluster
- Automatically archive versions of all programs used and record all intermediate results
- Easy to use

One solution

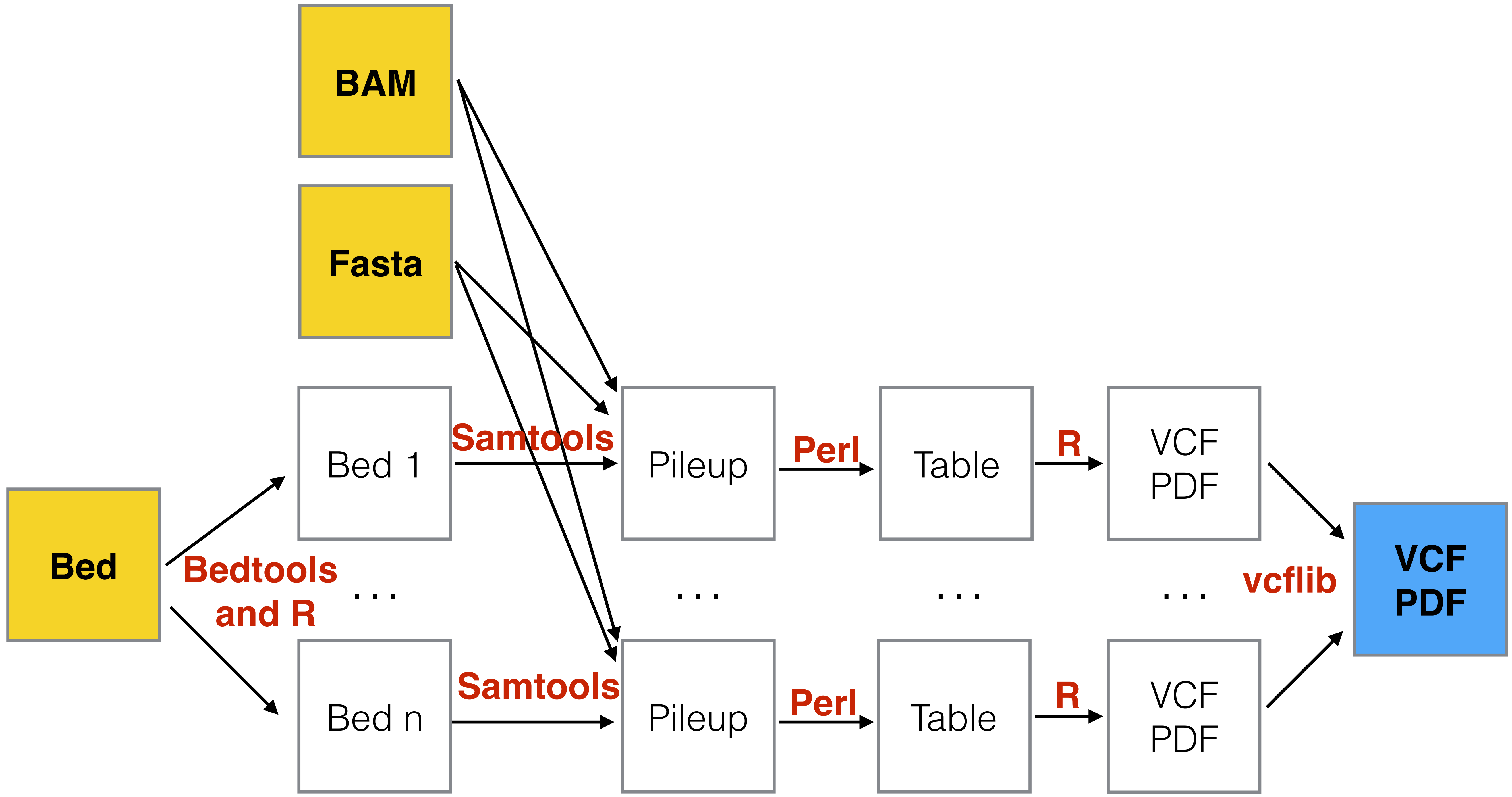


Case study: needlestack

- New somatic variant caller we are developing
- Requires: samtools, bedtools, vcflib, R, perl, bash, one perl script, two R scripts



needlestack in parallel



International Agency for Research on Cancer

World Health Organization

- Web interface to *git* version control system
  - keep track of all changes in your source code
- Provides many additional tools for collaboration:
  - Bug tracking, task management, feature requests, wikis
  - Public repositories are free

- "Docker is a tool that can package an **application and its dependencies** in a virtual container that can run in isolation on any Linux server, whether on premises, public cloud, private cloud etc."
- DockerHub can host images (free when public)
- Created by (in France) by Solomon Hykes, first release March 2013, open source.
- Raised 100M$ this year and worth ~1B$

International Agency for Research on Cancer

World Health Organization

nextflow

- Dataflow programming model (i.e. “pipeline”)
- Portable:
  - abstraction layer between pipeline's logic execution layer
  - can be executed on multiple platforms without changing the code
  - support SGE, LSF, SLURM and PBS/Torque schedulers and cloud platforms.
- Reproducible: supports docker and GitHub

International Agency for Research on Cancer

World Health Organization

The impact of Docker containers on the performance of genomic pipelines

Paolo Di Tommaso[1,2], Emilio Palumbo[1,2], Maria Chatzou[1,2], Pablo Prieto[1,2], Michael L. Heuer[3] and Cedric Notredame[1,2]

2015

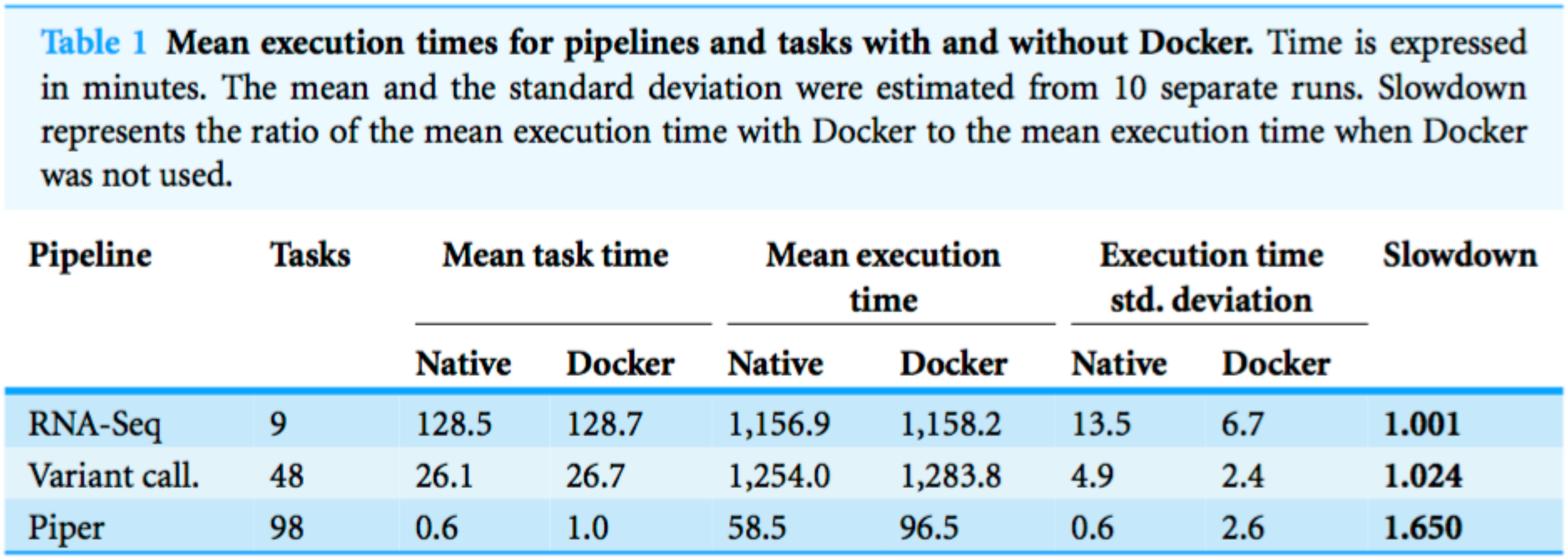


Table 1 Mean execution times for pipelines and tasks with and without Docker. Time is expressed in minutes. The mean and the standard deviation were estimated from 10 separate runs. Slowdown represents the ratio of the mean execution time with Docker to the mean execution time when Docker was not used.

Pipeline	Tasks	Mean task time		Mean execution time		Execution time std. deviation		Slowdown
		Native	Docker	Native	Docker	Native	Docker	
RNA-Seq	9	128.5	128.7	1,156.9	1,158.2	13.5	6.7	**1.001**
Variant call.	48	26.1	26.7	1,254.0	1,283.8	4.9	2.4	**1.024**
Piper	98	0.6	1.0	58.5	96.5	0.6	2.6	**1.650**

International Agency for Research on Cancer

World Health Organization

Moving forward

- Too much automation is too much
- Some people tried this but ended up with overly complicated solutions (personal opinion):
  - NGSeasy: https://github.com/KHP-Informatics/ngseasy
  - bcbio: https://github.com/chapmanb/bcbio-nextgen
  - exscalibur: https://github.com/cribioinfo
- Implement our pipelines in this simple framework

Future

- Cloud computing: Amazon EC2, Google, Microsoft
- StarCluster and elasticluster: automate the creating of cluster computing environment in the cloud
- Docker announced "Swarm" Nov. 3rd 2015

  "Swarm is the easiest way to run Docker applications at scale on a cluster. It turns a pool of Docker Engines into a single, virtual Engine. You don't have to worry about where to put containers, or how they talk to each other – it just handles all that for you."