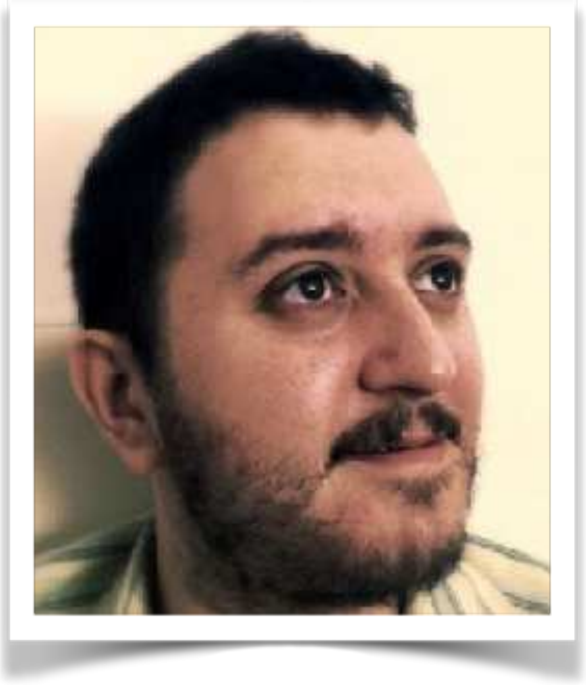


Toni Hermoso

Bioinformatician at
the core facility (CRG)

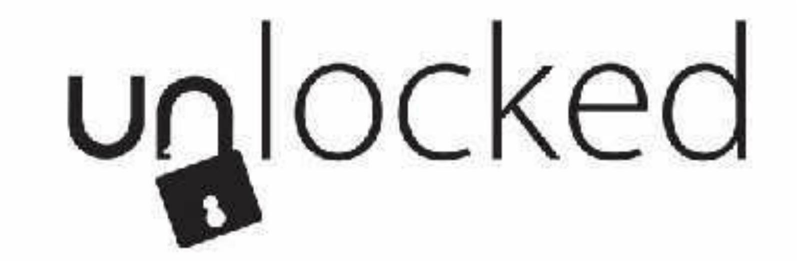
Guillaume Filion

Group leader
genome architecture (CRG)

In the beginning were computers and the Internet.

Open access in research

Open means Transparent



Open means Accessible

Open access publications

2000

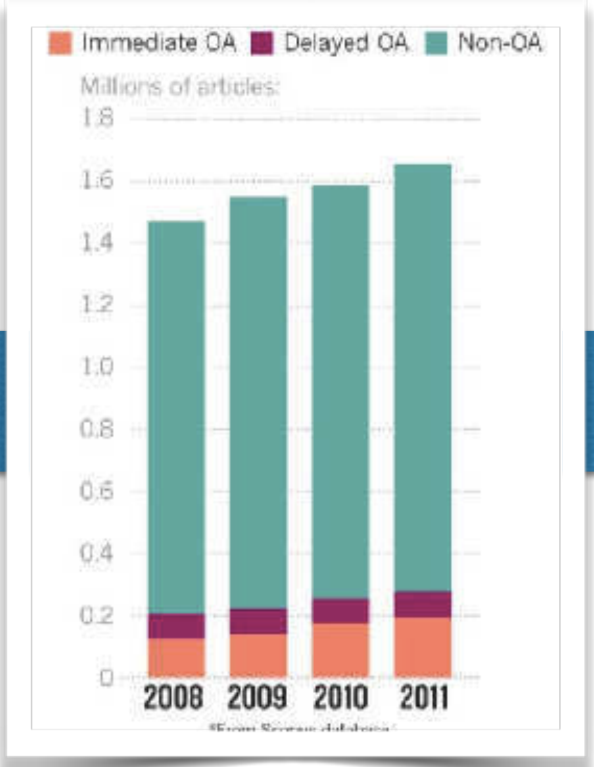
PLOS

2002

Budapest open access initiative

2003

Berlin declaration on open access
The Internet has fundamentally changed the practical and economic realities of distributing scientific knowledge (…)



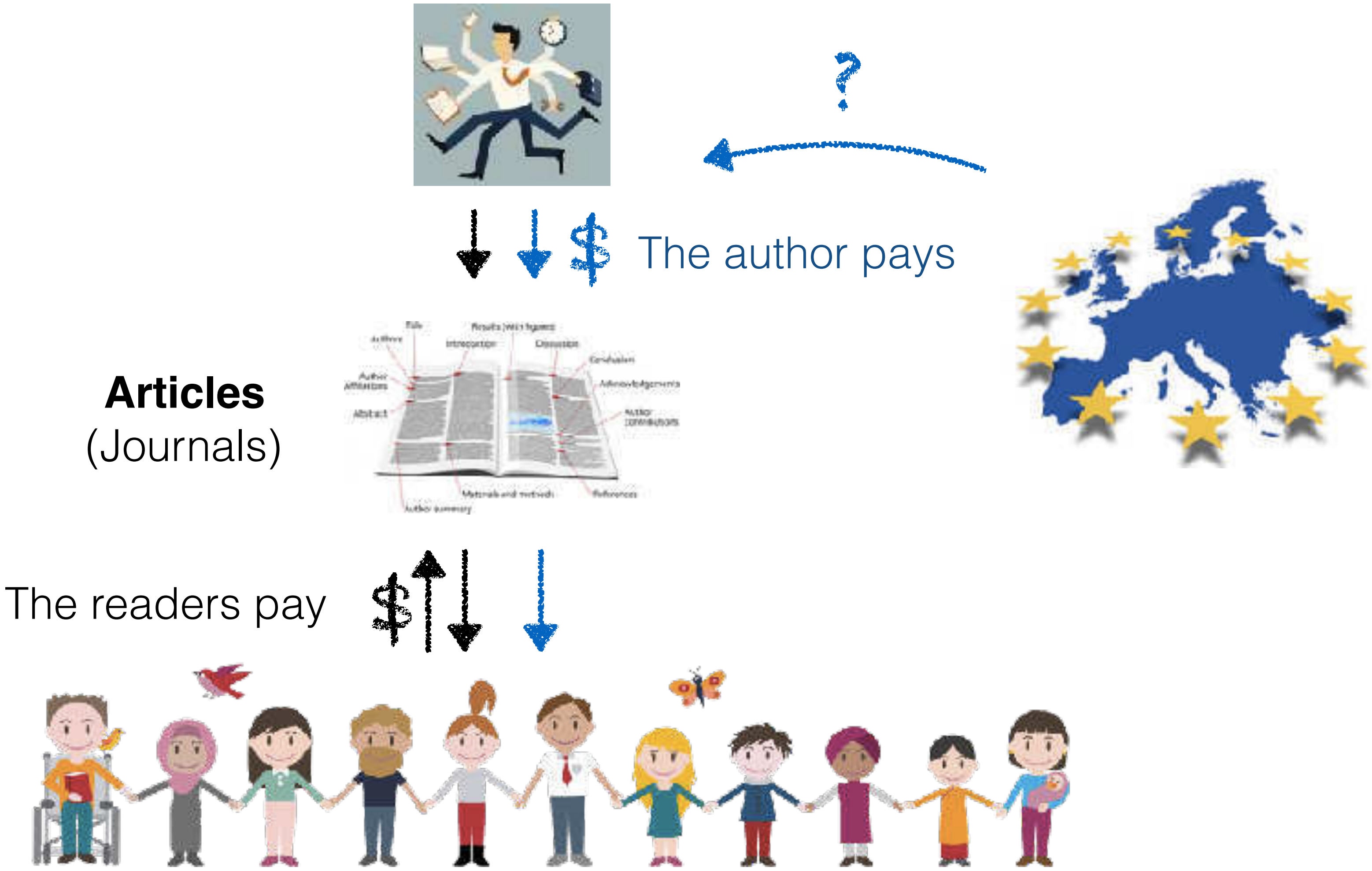
2014

Mandatory open access for H2020-funded research

2015

bioRχiv beta

The business model



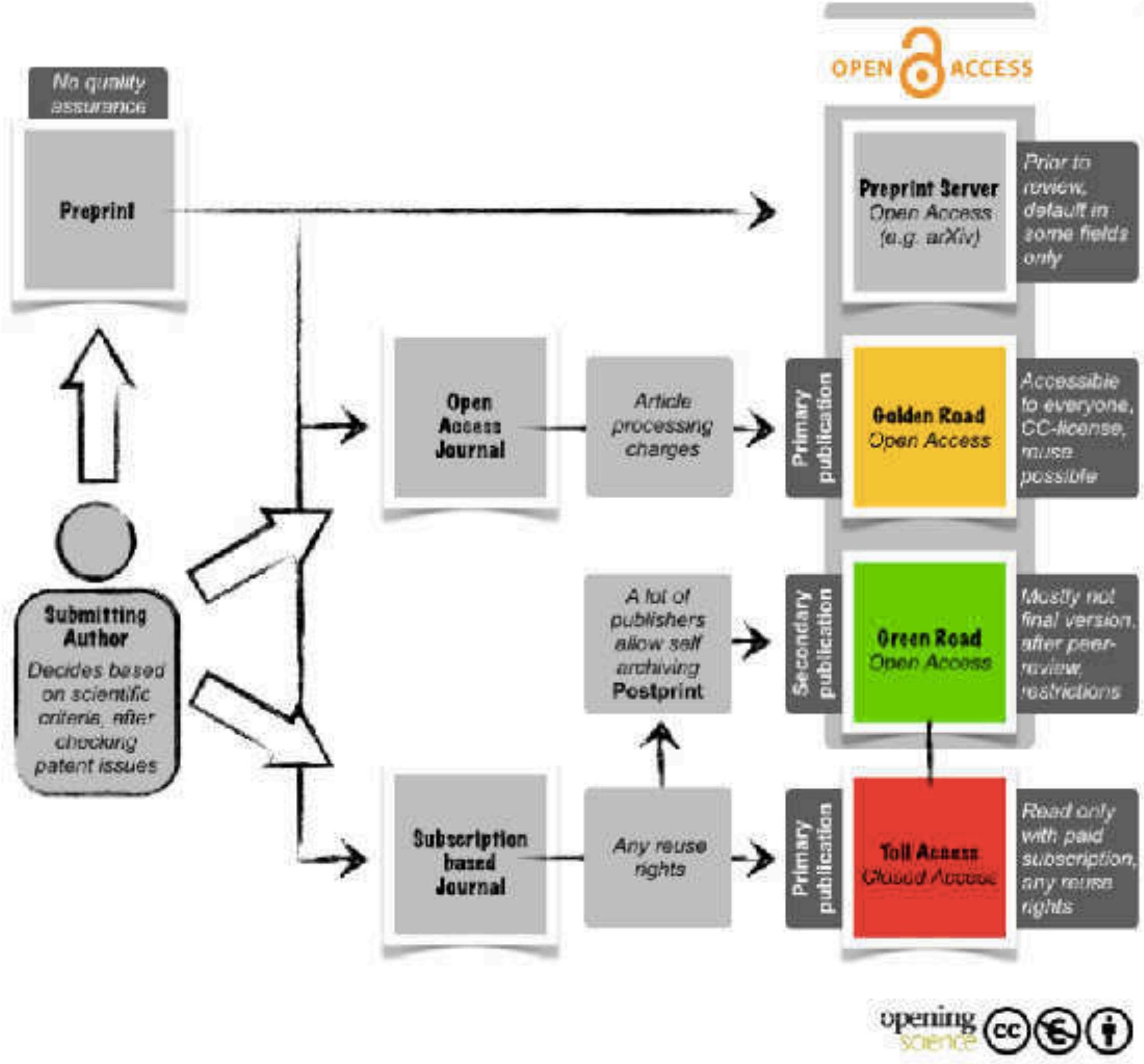
?

$ The author pays

Articles
(Journals)

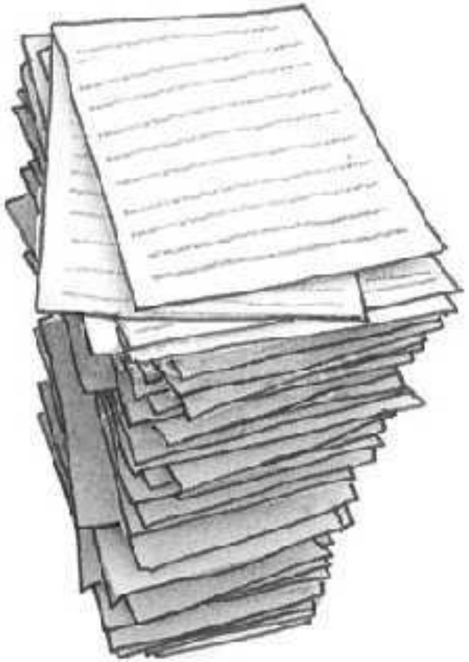
The readers pay $

Publishing today


OPEN ACCESS
No quality assurance
Preprint
Submitting Author
Decides based on scientific criteria, after checking patent issues
Open Access Journal
Article processing charges
Subscription based Journal
Any reuse rights
A lot of publishers allow self archiving Postprint
Preprint Server Open Access (e.g. arXiv)
Prior to review, default in some fields only
Primary publication
Golden Road Open Access
Accessible to everyone, CC-license, reuse possible
Secondary publication
Green Road Open Access
Mostly not final version, after peer-review, restrictions
Primary publication
Toll Access Closed Access
Read only with paid subscription, any reuse rights
opening science

source: http://book.openingscience.org/tools/open_access_state_of_the_art.html

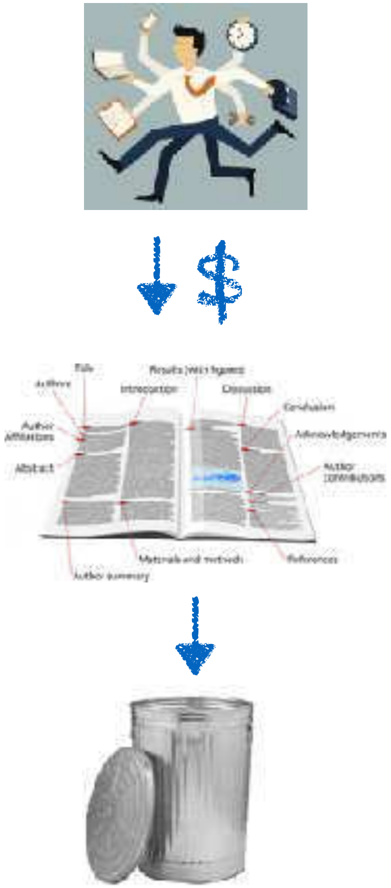
Costs of open access (1)



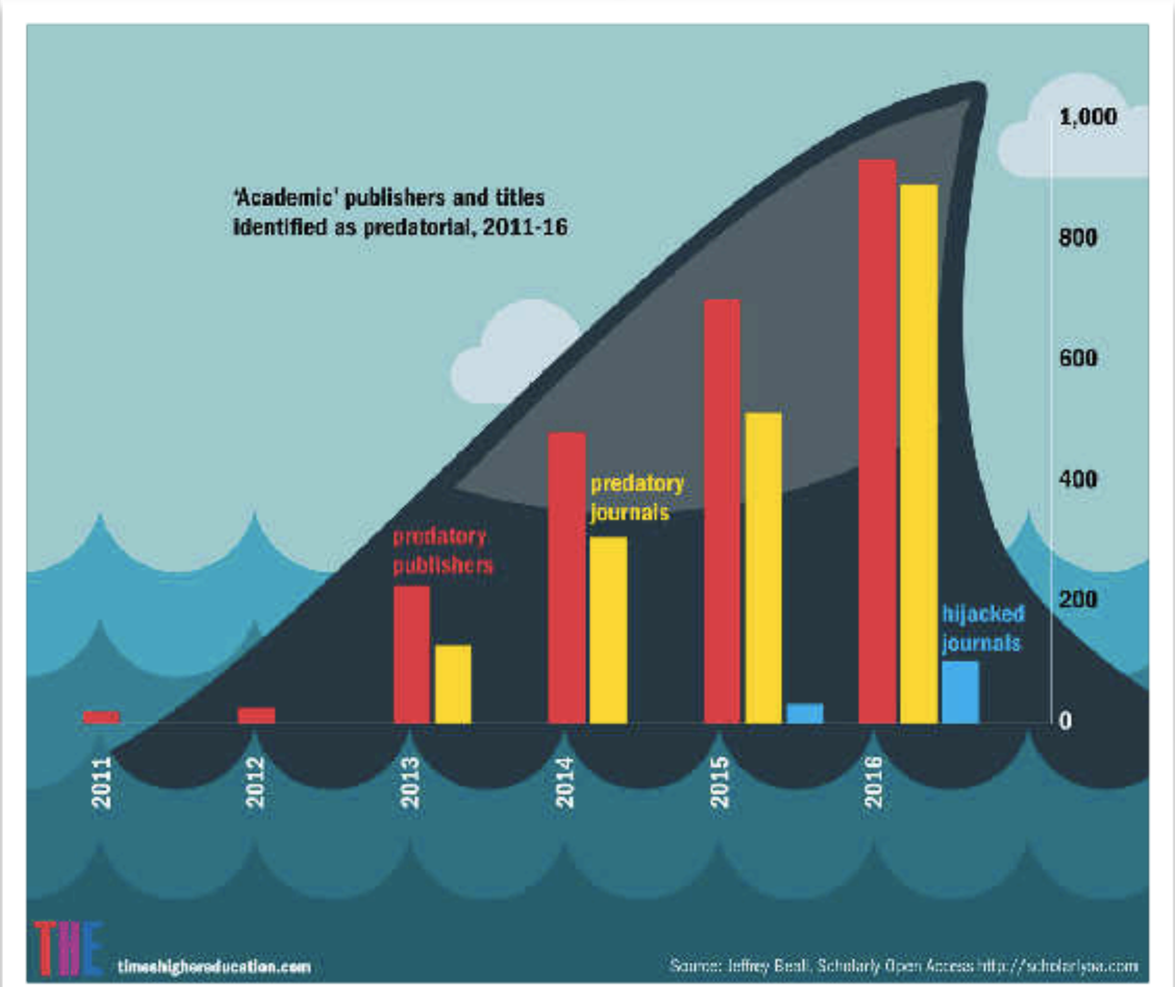
Every day on PubMed
2600 new articles.
This is ~13 million € fees.

Who pays?

Costs of open access (2)

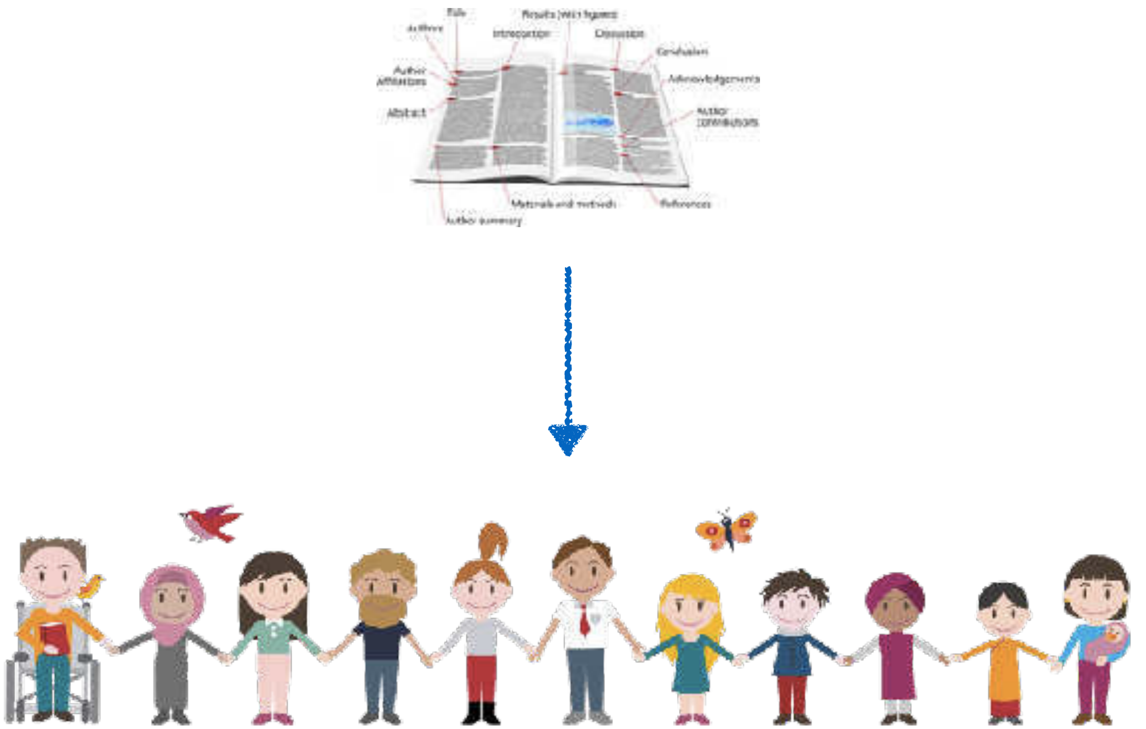


Predatory journal

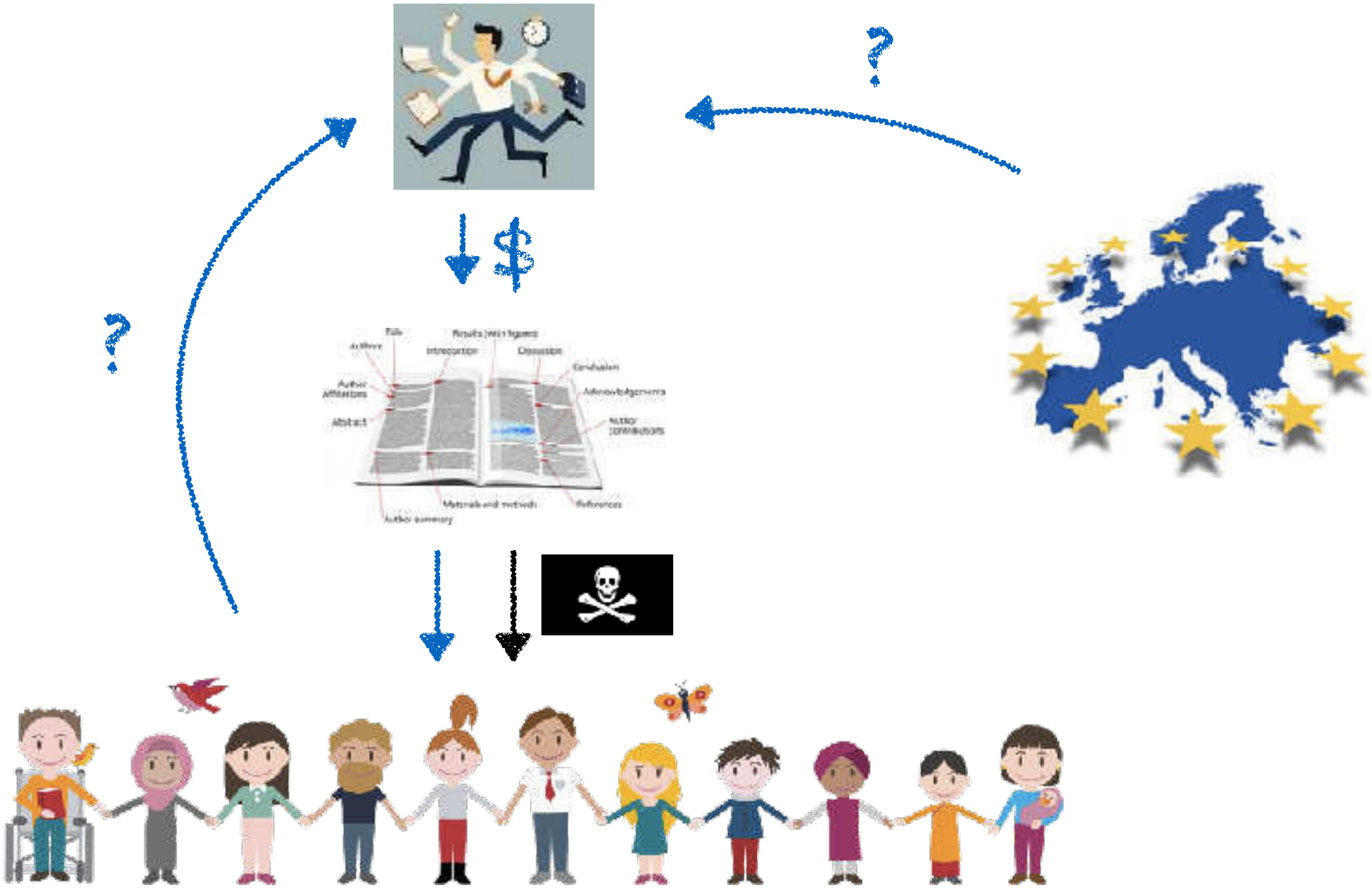


Source: Jeffrey Beall, Scholarly Open Access http://scholarlyoa.com

Benefits of open access (1)



Benefits of open access (2)

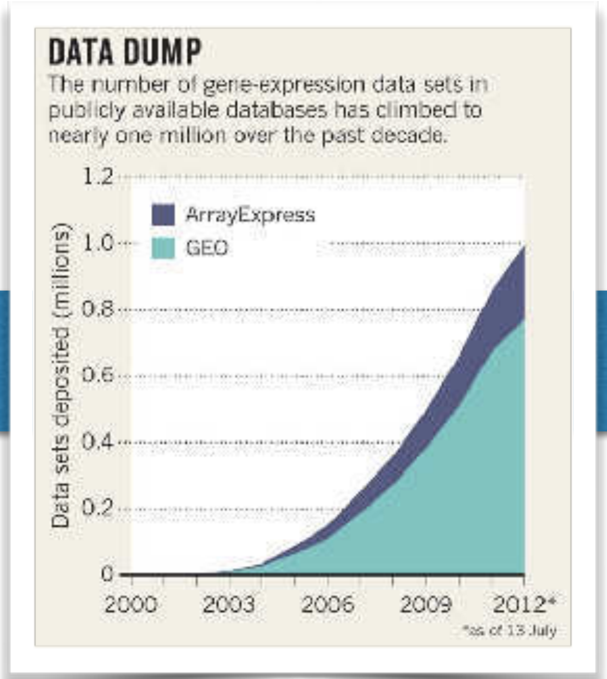


Why publish open access?

Open access data

GEO
Gene Expression Omnibus

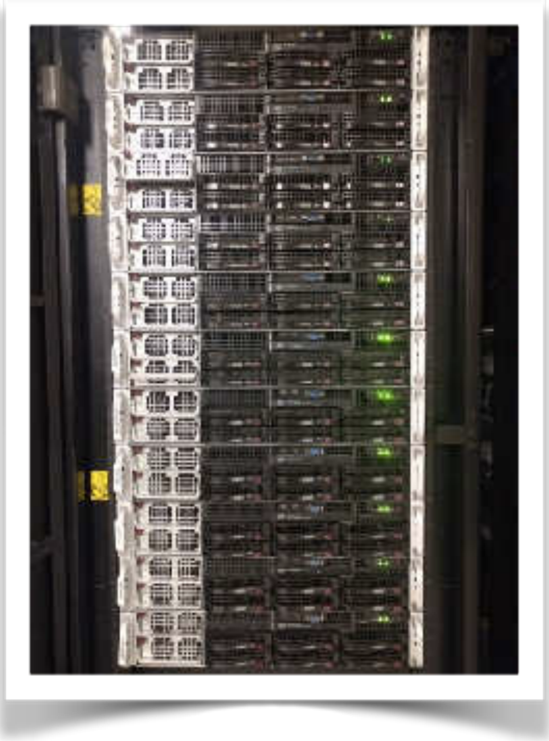
2002 2003


DATA DUMP

The number of gene-expression data sets in publicly available databases has climbed to nearly one million over the past decade.

Costs of open data (1)

ENA
European Nucleotide Archive



ENA is > 5000 TB
Cost much smaller
than publications

Who pays?

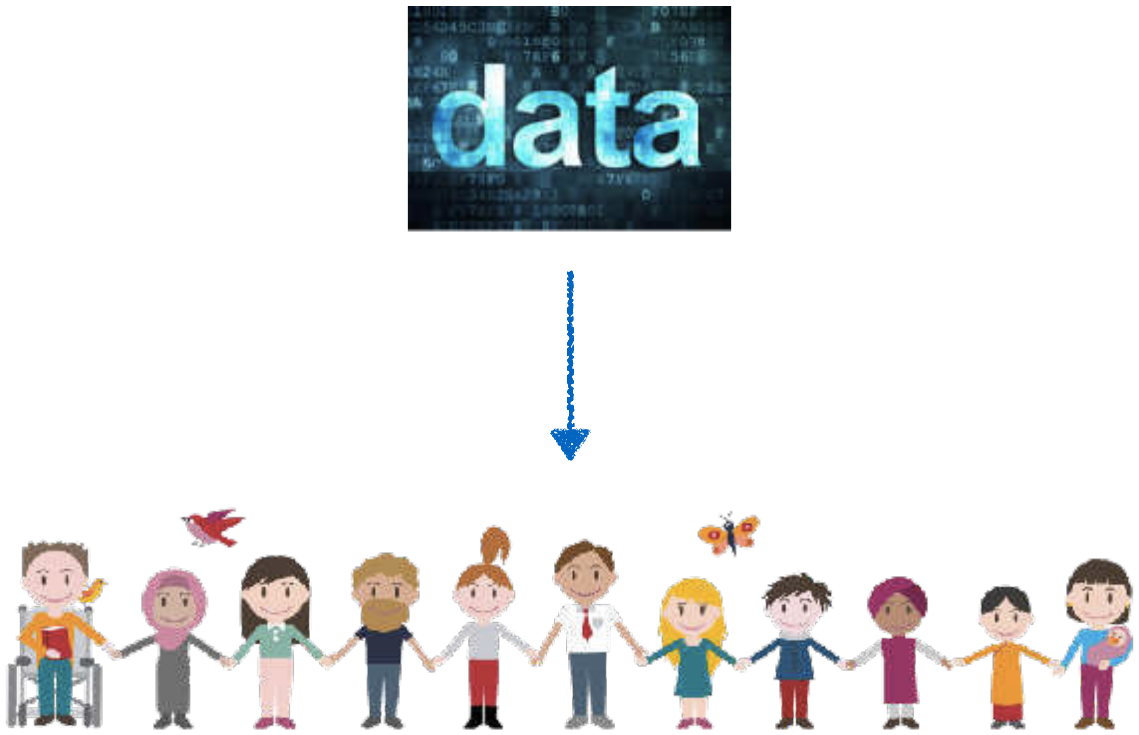
Costs of open data (2)

CONFIDENTIAL

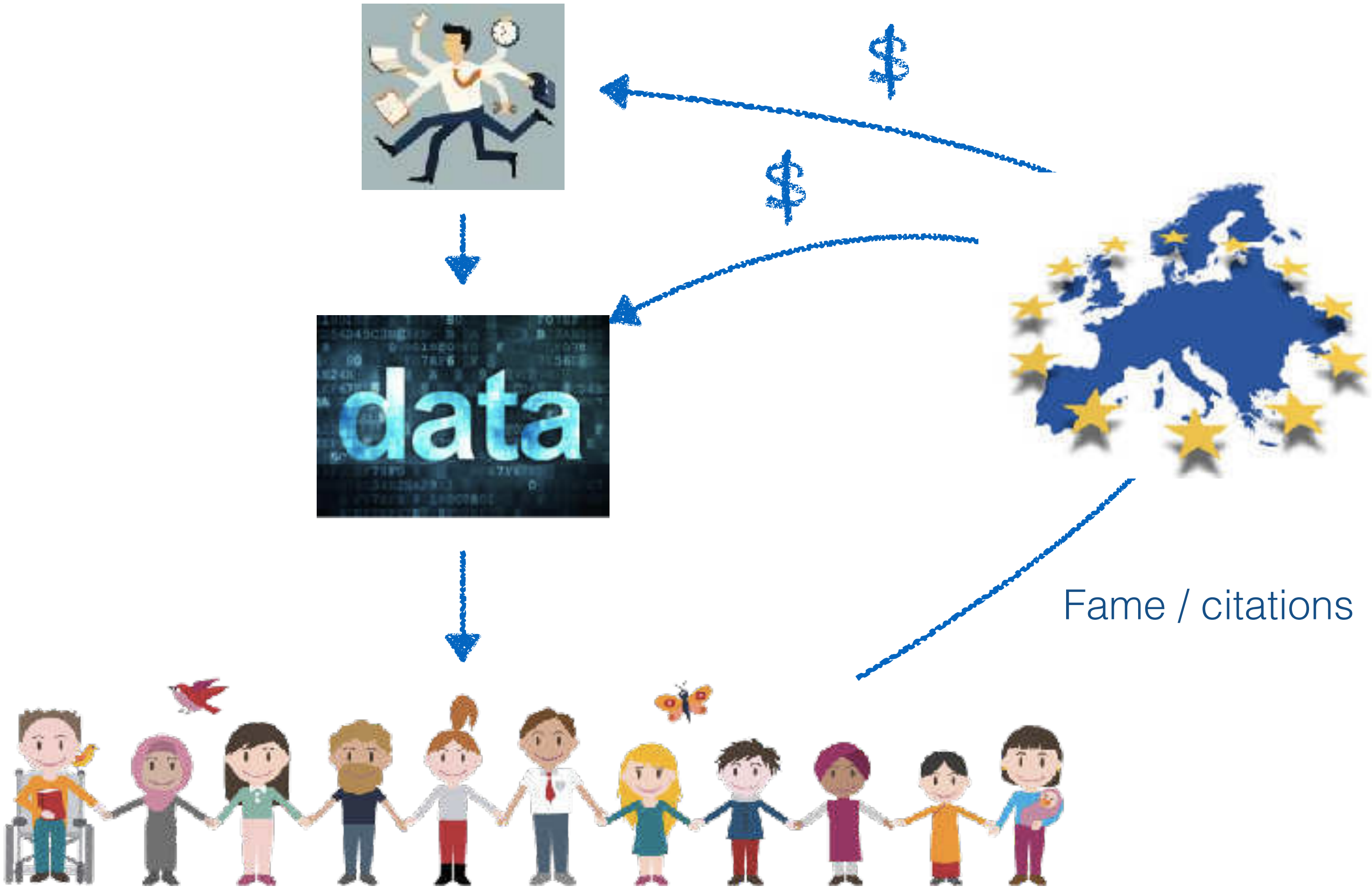
Confidential data cannot be open

Opening personal data may backfire

Benefits of open data (1)

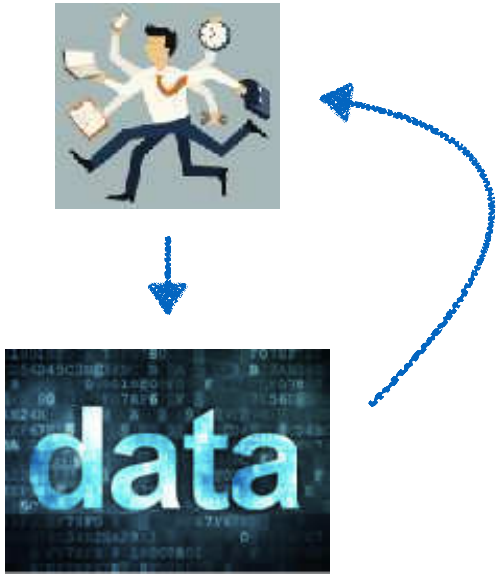


Benefits of open data (2)



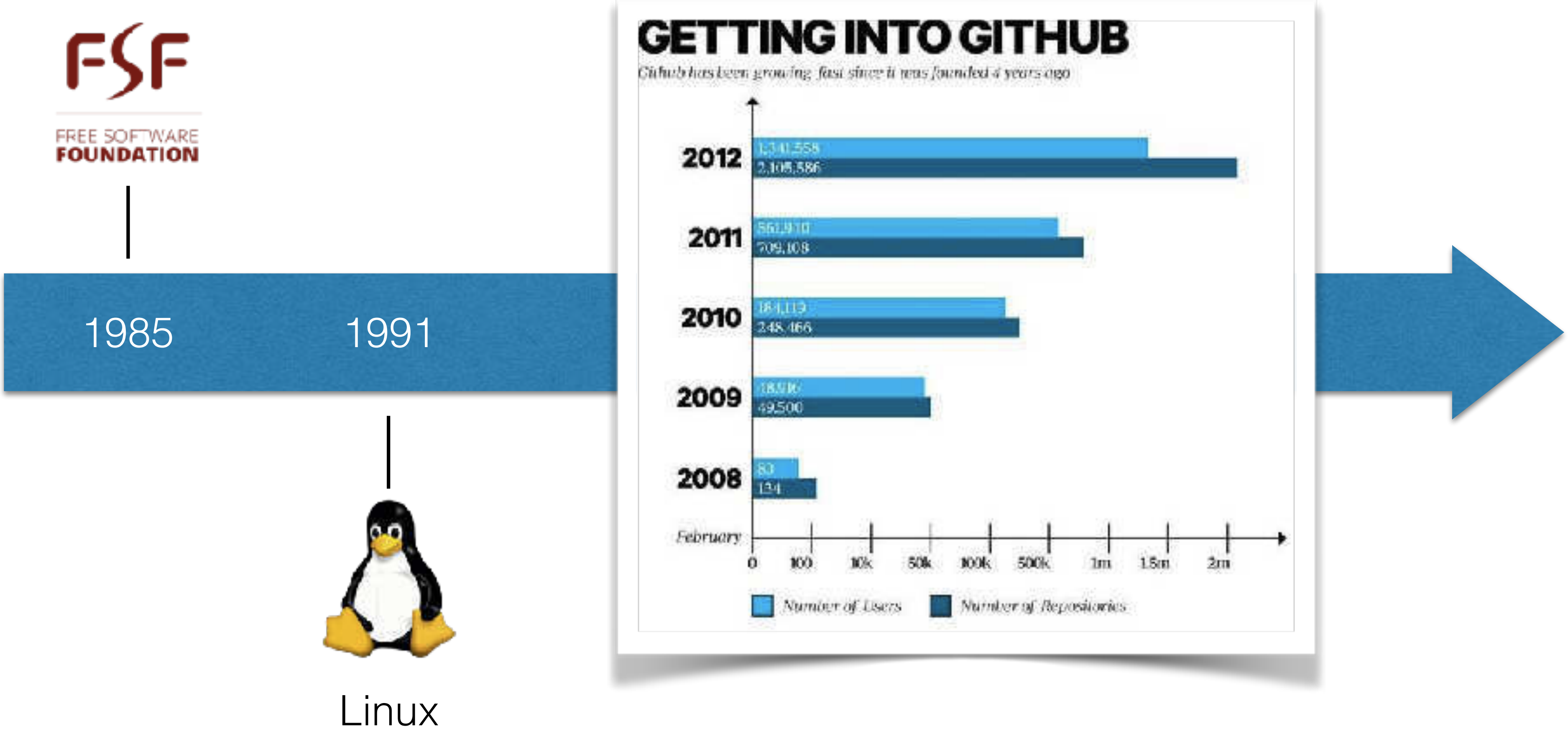
Fame / citations

Benefits of open data (3)



Quality
Safety
Cost

Open access code



Costs of open code

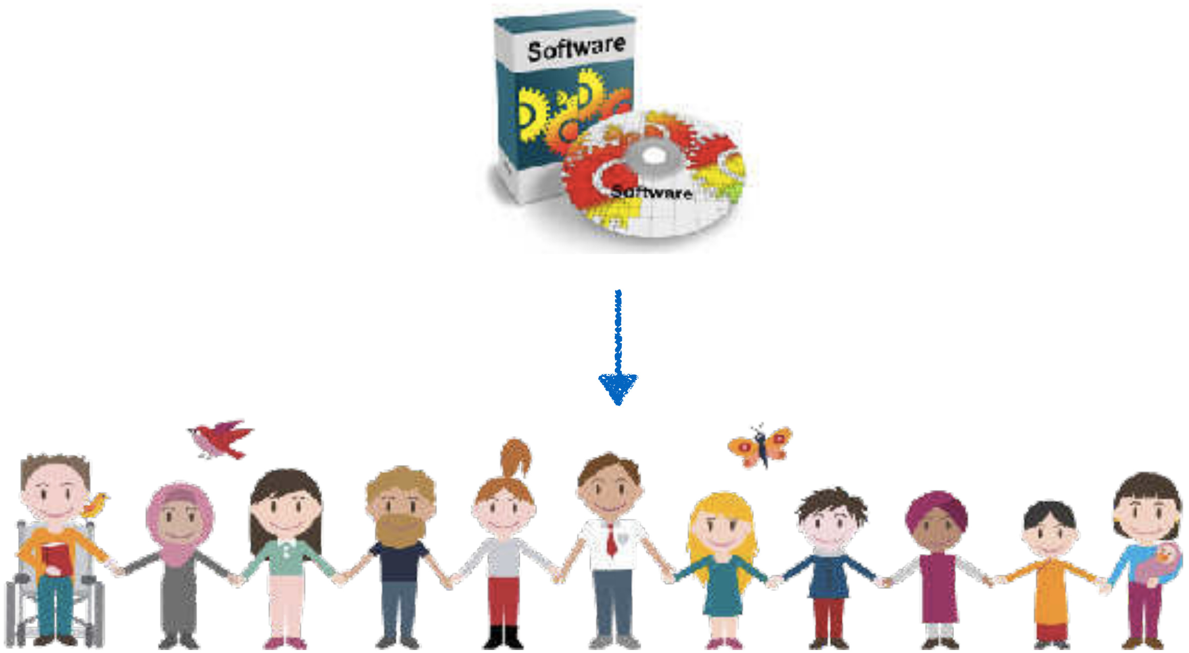
User support / new features

Write portable code

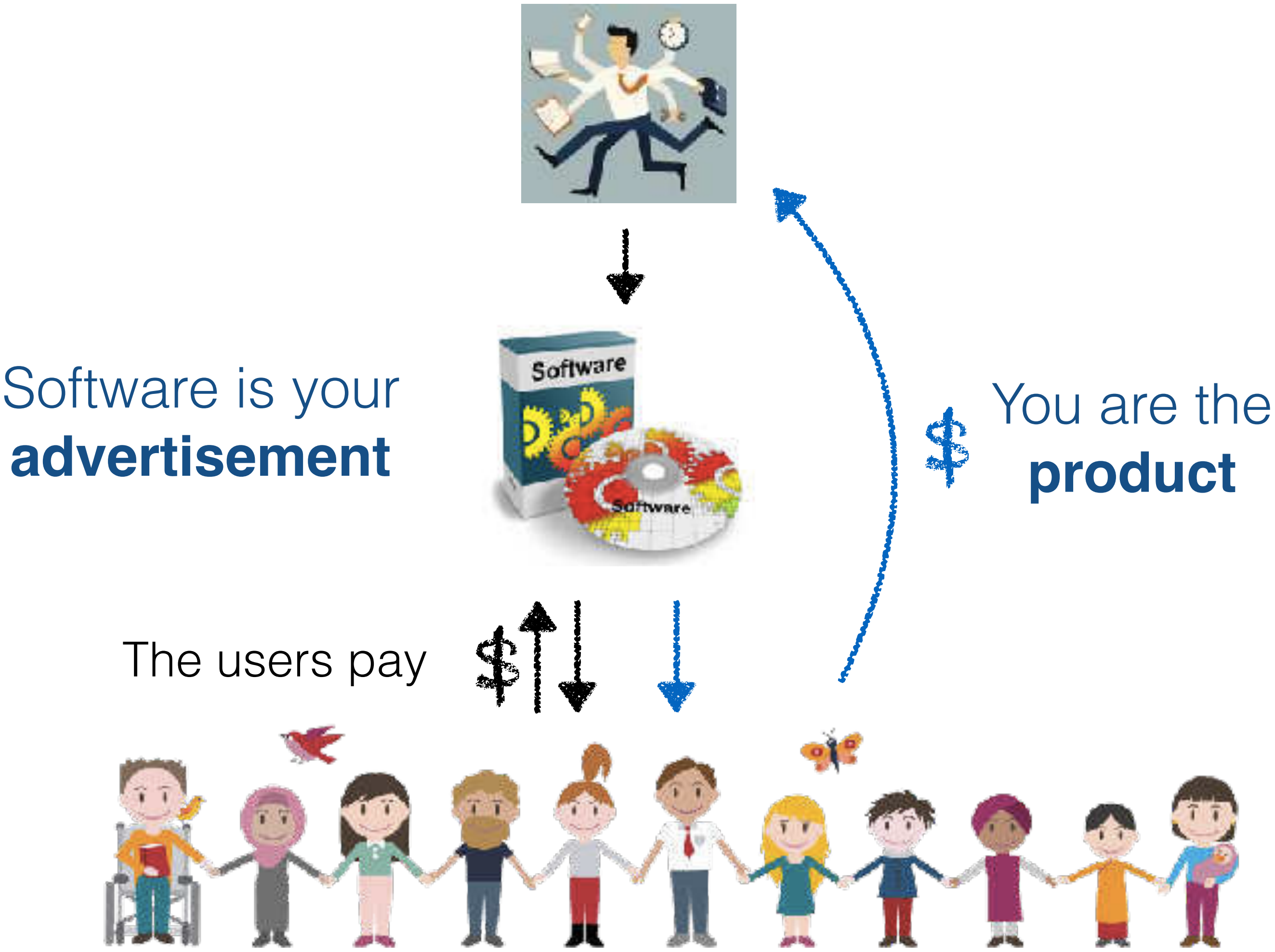
Non profit

Who pays?

Benefits of open code (1)



Benefits of open code (2)

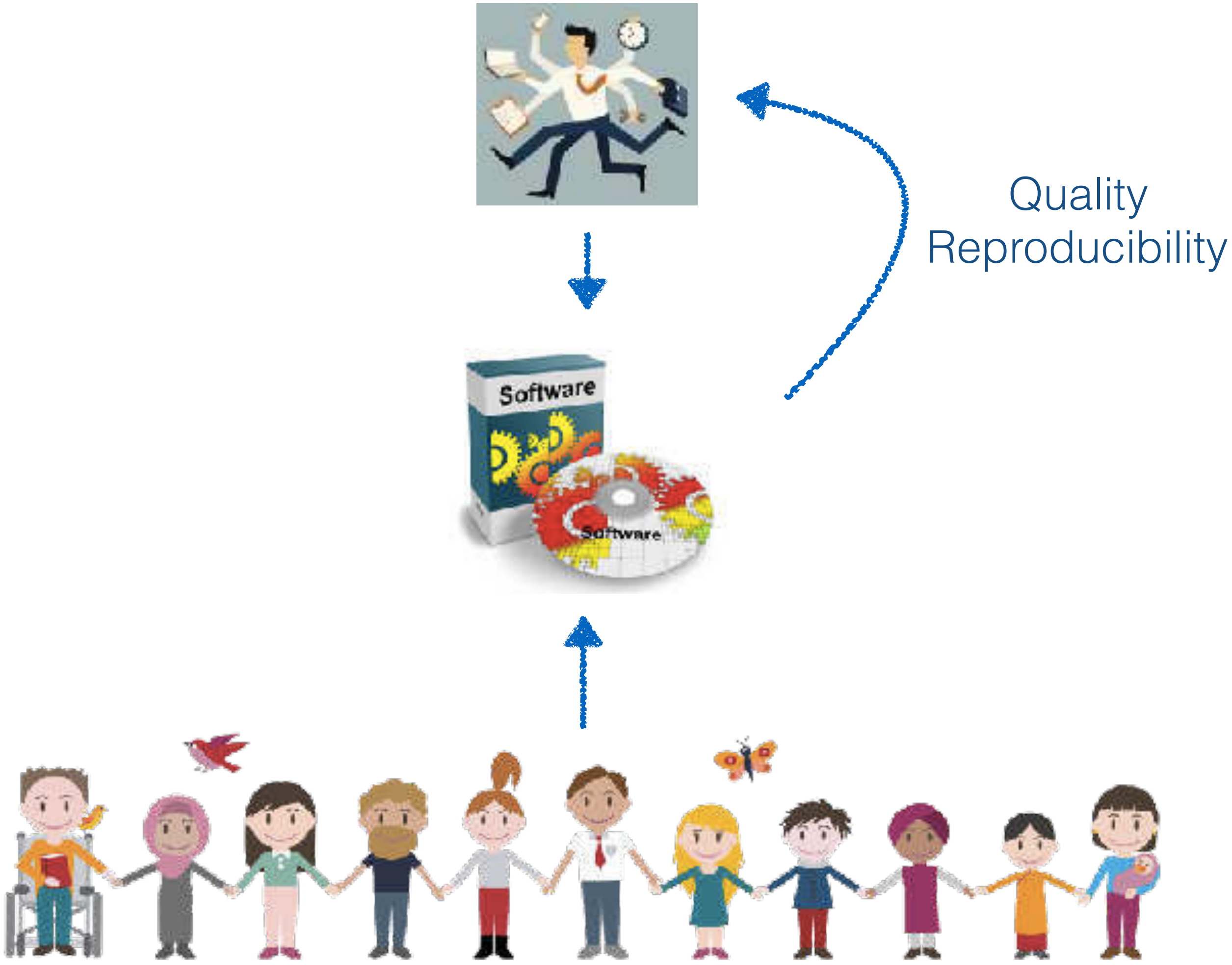


Software is your **advertisement**

You are the **product**

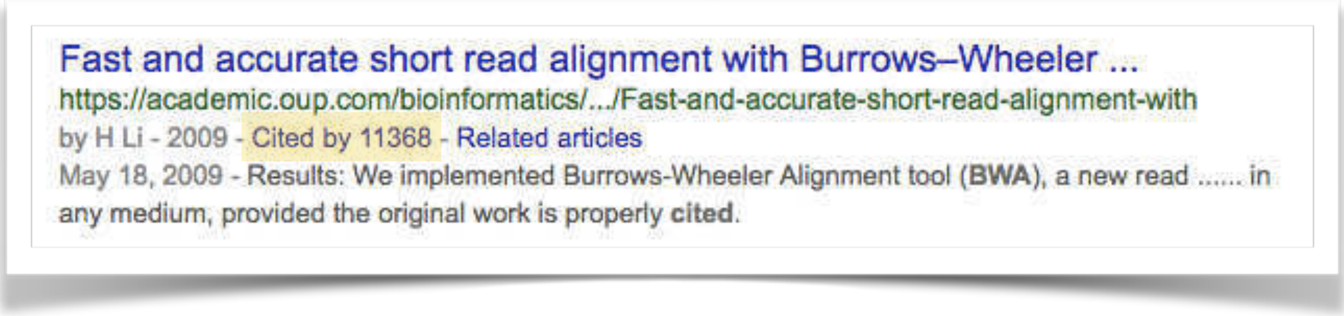
The users pay

Benefits of open code (3)



Quality
Reproducibility

Benefits of open code (4)



Fast and accurate short read alignment with Burrows–Wheeler ...
https://academic.oup.com/bioinformatics/.../Fast-and-accurate-short-read-alignment-with
by H Li - 2009 - Cited by 11368 - Related articles
May 18, 2009 - Results: We implemented Burrows-Wheeler Alignment tool (**BWA**), a new read ...... in any medium, provided the original work is properly **cited**.

Open access software and data can **boost** your research.

But how to do it right?

Open Science. Good practices in Bioinformatics

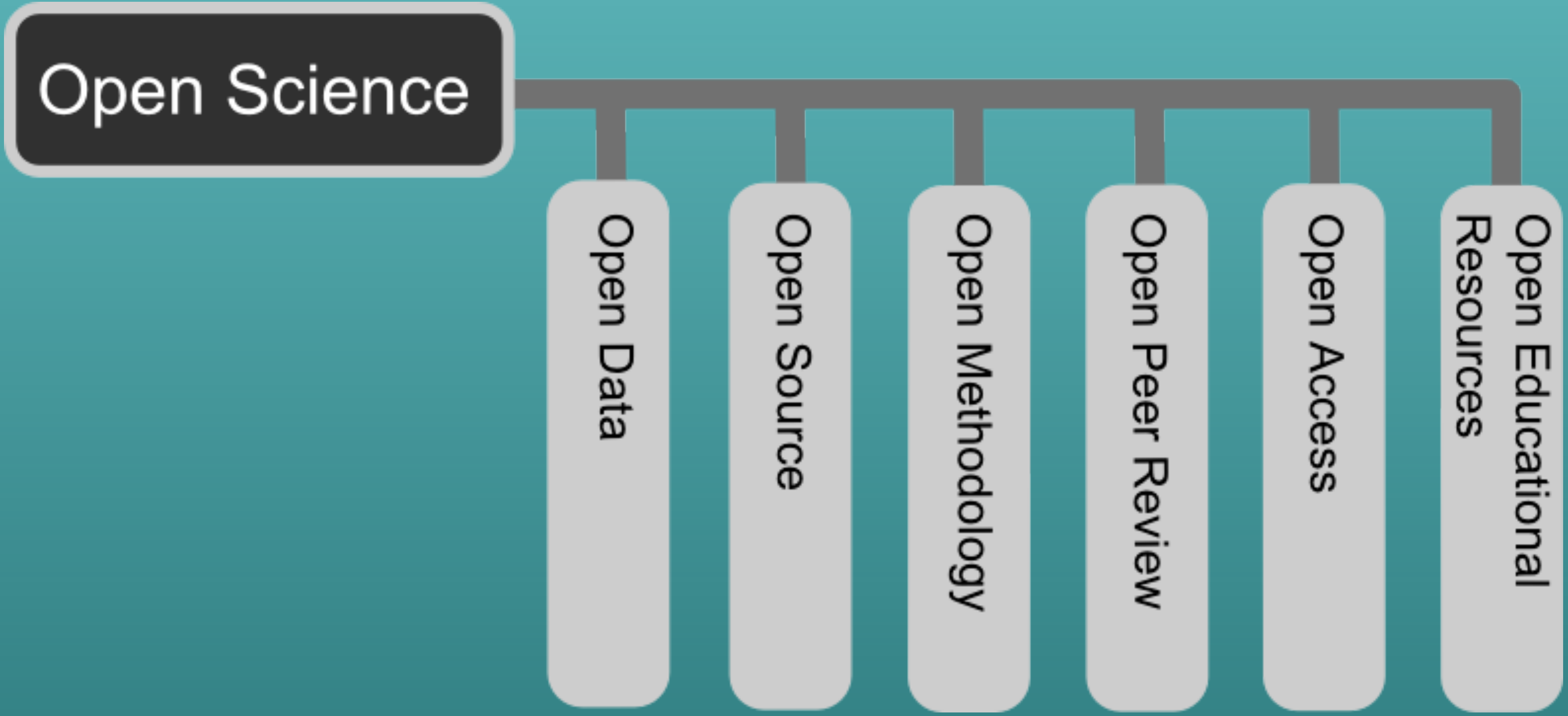
Toni Hermoso Pulido (@toniher)

Bioinformatics Core Facility

Centre for Genomic Regulation (BCN)

https://biocore.crg.eu

Open Science



Open Science

Open Data

Open Source

Open Methodology

Open Peer Review

Open Access

Open Educational Resources

The six principles of Open Science

Document

Write it down or …

it didn't happen!

Document: Why?

- Organise ideas
- Understanding code and steps in the future for you and others
- Fixing errors
- Help in future publication

Document: Where?

- File System (e.g. README or TODO files)
- Control Version System
  - Git, SVN, etc.
- Content Management System
  - Wiki CMS, Drupal, etc.

Document: How?

- Plain text
- Format
    - Unstructured
        - Free
        - Markdown
        - Wikitext

Document: How?

- Format
  - Structured
    - Config files
      - XML, JSON, INI, YAML
    - Templates (e.g. in wikis)
    - Database Management Systems (Relation or NoSQL)

Tag and track

I never said so!

Tag and track: Why?

- Convenient backup
- Error tracking and reversion
- Checking history
- Allowing collaboration on different time points
- Publication of specific snapshots

Tag and track: Where?

- Code, documentation:
  - Control Version System (Git, SVN, etc.)
    - Interfaces:
      - Github
      - Gitlab (local installation)
  - Wiki CMS (e.g. [Semantic] MediaWiki)
- Data, files
  - Plain Git (small files) or Git with large files
  - Document Management Systems

Tag and track: Concepts

- Revision, Version, Commit
- Branch
- Tag, Release
- Fork, Pull request

Tag and track: Publish

- Working and executable code
  - Docker & Singularity hubs
- Identify Content & Code (DOI)
  - Figshare
  - Zenodo (with Github)
- Bio specific repositories
  - Sequence Read Archive (SRA)
  - GEO Archive (Genome Expression Data)
  - ENA, EGA and others. Detail

Reproduce

Run it again, Sam!

Reproduce: Why?

- Nowadays not only textual statements but also code and data
- Peers and collaborators should be able to reproduce by themselves
  - Check errors
  - Improve code, data
  - Test in different conditions

Standing on the shoulders of giants

Reproduce: How?

- Code requirements, recipes
  - Scripts
  - Test frameworks
  - Package managers (e.g. Conda)
  - Jupyter
- Virtualisation
  - Hypervisor: VirtualBox, VMWare, etc.
  - Containers: Docker, Singularity

Reproduce: Note on python

- pyenv & pyenv-virtualenv
  - `pyenv install x.y.z`
  - `pyenv virtualenv x.y.x myvenv`
- pip
  - `pip freeze > requirements.txt`
  - `pip install -r requirements.txt`

Reproduce: Other languages

- Perl: perlbrew
- PHP: phpbrew
- Java: jenv
- NodeJS: nvm
- etc.

Reproduce: Conda

- Popular package manager
  - Takes care also of binaries, libraries
- Bioconda: specific Bioinformatics recipes

Reproduce: Jupyter

- Former *IPython Notebook*
- Combines in a single notebook documentation (Markdown), comments and executable code with its output
- Underlying notebook format is a JSON text file
    - Can be exported into PDF, HTML, etc.

Reproduce: Jupyter

- Apart from Python (2 or 3), now also different languages with *Kernels*:
  - R, Perl5, Perl6, Javascript, more…
- Additional widgets (e.g. for charts)
- Convenient for sharing code and training
- Jupyter gallery in Github

Reproduce: Docker

- Allows shareable Linux systems that can be run in any machine were Docker is installed
- Build images with a script file (Dockerfile), very similar to a Linux command-line script
- Repository of Docker images
    - You can reuse, adapt, extend
    - Don't reinvent the wheel

Reproduce: Docker

- Microservices principle
  - 1 Image -> n Containers -> n Services
  - n Services -> 1 full application
- Example: BLAST Web application
  - Web server container
  - Database container
  - BLAST application running container
- Making it work together:
  - system scripts
  - Docker compose
  - etc.

Reproduce: Singularity

- Like Docker but more suitable for HPC environments
- No need of a Docker daemon running / less problematic for security
- Docker images convertible into Singularity ones
  - Conversion script
- Singularity Repository

Recomendations to containerize your bioinformatics software

Pipelines & Workflows

Guilty by association

Pipelines & Workflows: Why?

Unix Philosophy

D. McIlroy, P.H.Salus

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.

Pipelines & Workflows: How?

- Traditionally from Shell script files
- Frameworks or applications
  - Web-based
    - Galaxy
  - GUI and command-line
    - Apache Taverna
  - Command-line
    - Nextflow
- Common Workflow Language

Pipelines and Workflows: Nextflow

- Concepts
  - Processes
    - Any pipeline or program (in any language)
    - In local disk or in containers (Singularity, Docker)
  - Channels
    - FIFO queue
    - Normally files in a filesystem

Pipelines and Workflows: Nextflow

- Concepts
  - Config files
    - Different config files, calling one to another can be created for adapting to different scenarios
  - Executors
    - Local machine
    - HPC cluster: SGE, Univa, SLURM, etc.
    - Cloud systems: Amazon Cloud, Apache Ignite

Questions?

Comments?

Diversity

There's more than one way to do it

Criteria

- Kind of tasks
- Team profiles
- Infrastructure and privacy
- Previous knowledge and time

Criteria: Tasks

- Data Analysis
- Interface / Web programming
- Teaching/Training

- Environment (where can be acheived)
  - Interface/Web
  - HPC
  - etc.

Criteria: Profiles

  - Wet lab scientists
  - Statisticians, programmers
  - Citizens

- Personal and working situations
  - Interns, PhD students, PostDocs
  - Technicians (full-time, temporary)
  - Project funding length

Criteria: Infrastructure, privacy

- Data transfer
  - Cluster vs Cloud
- Sysadmin or devops support
- Human or clinical data involved
- Funding vs time

Criteria: Knowledge

- Programming language(s)
  - Python, R, JavaScript, Java, Perl
- Availability of libraries / reusing
- Frameworks, platforms
  - Learning curve
  - Bus factor