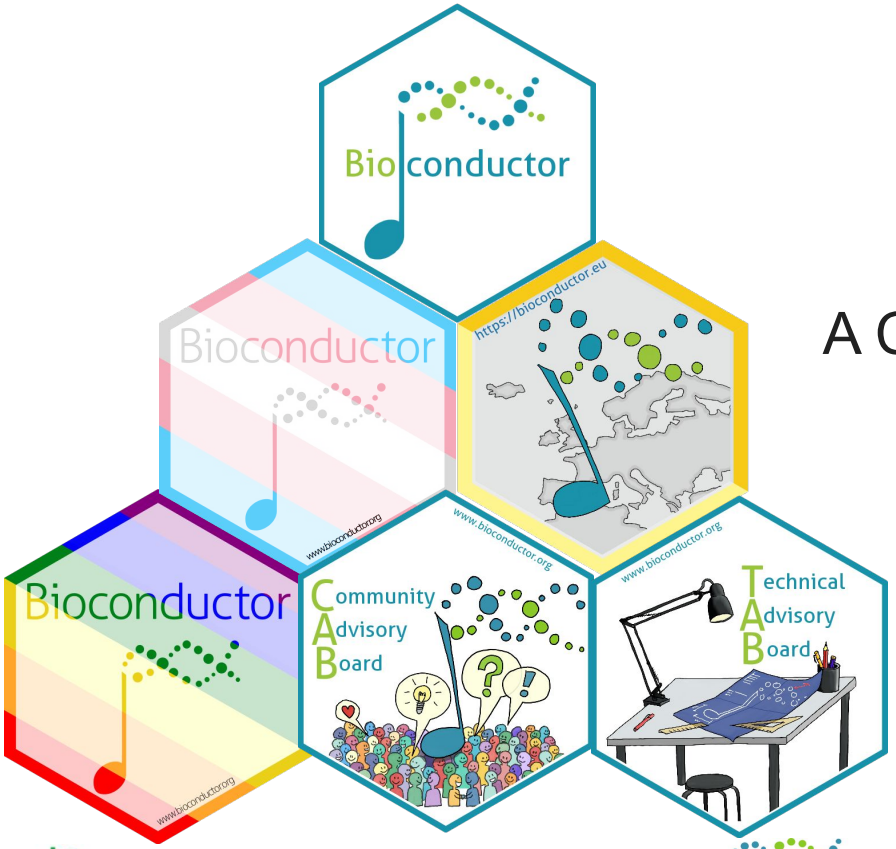


Bioconductor:

A Case Study in Open Source Innovation and Social Impact in Life Sciences

Dr Maria Doyle
Bioconductor Community Manager

National Open Source Innovation Summit
Feb 7th 2025

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

LERO

My role as Bioconductor Community Manager

- Support 1,000+ developers and 1M+ users worldwide.

- Lead global expansion (e.g., Latin America, Africa).

- PI on projects, driving training and community growth.

Bioconductor: Building on R's Open Source Legacy

1991: Ross Ihaka, Robert Gentleman begin work on a project that will become R

1993: The first announcement of R

1995: R available by ftp

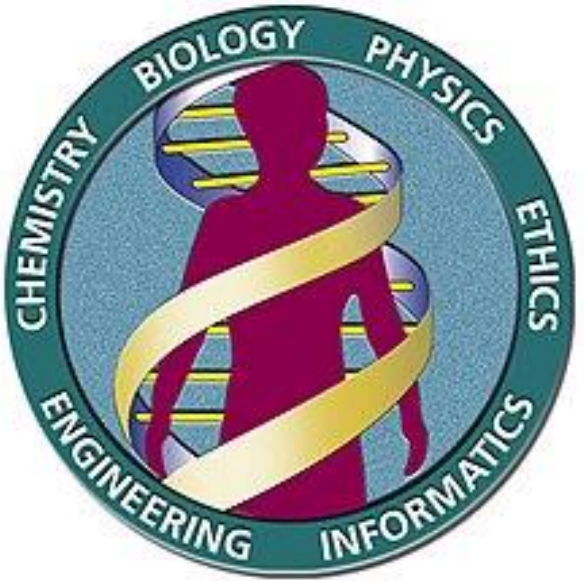
1996: A mailing list is started and maintained by Martin Maechler at ETH

1997: The R core group is formed

2000: R 1.0.0 is released

Rising demand for scalable biological analysis

- Coincided with technology for biology at scale: Human Genome Project (started 1990s)
- Confluence of
  - high throughput biology requiring new statistical methods
  - first open source statistical language - R



Bioconductor Early Years

- Established by Robert Gentleman in 2001
- Open-source R package repository, version control SVN and package checks.
- Specialized data classes for biological data -> interoperability
- Became the de-facto repo for statisticians publishing new methods

BioConductor: software for bioinformatics

About Bioconductor
Main Page
Bioconductor FAQ
Contributors
What's New?

Software
Released Packages
Developmental Packages
Change log

Data
hgu95A Human data package
hgu6800 Human data package
mgu74A Mouse data package
Sources Of Data

Services
Annotation
Workshops

Project
Collaboration

The Bioconductor project produces an open source software framework that will assist biologists and statisticians working in bioinformatics, with primary emphasis on inference using DNA microarrays. The team is based at the Biostatistics Unit of the Dana Farber Cancer Institute at Harvard Medical School/Harvard School of Public Health. Participation of interested developers/investigators at other institutions is encouraged.

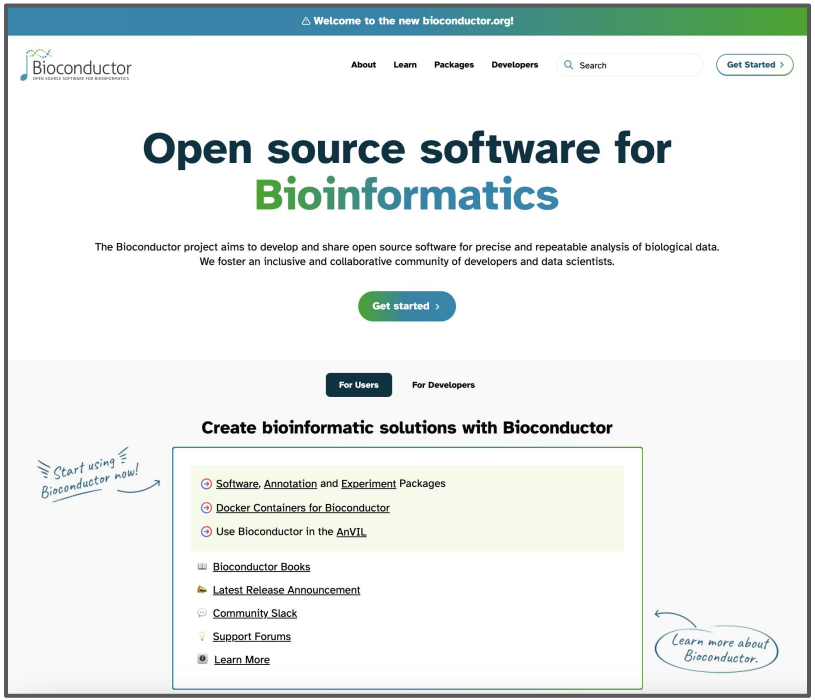
Basic features of the project include

- commitment to full open source discipline, with distribution via a SourceForge-like platform. All contributions are expected to exist under an open source license such as GPL2 or BSD
- commitment to design-by-contract principles, emphasizing interoperation of independently designed and engineered components that may be reused in other contexts
- emphasis on the interactive statistical computing and software development environment R (www.r-project.org) as a repository for analytical algorithms, a full-featured programming language, and a paradigm for package-based object-oriented environment design and implementation.

Upcoming Bioconductor Release: April 29, 2002!

Bioconductor in 2025

- Now version 3.20
- Community-driven
- Core value supporting users to become developers.
- 2,000+ contributed packages, providing tools for biological data analysis, visualisation, and comprehension.
- Downloads >1 million per year



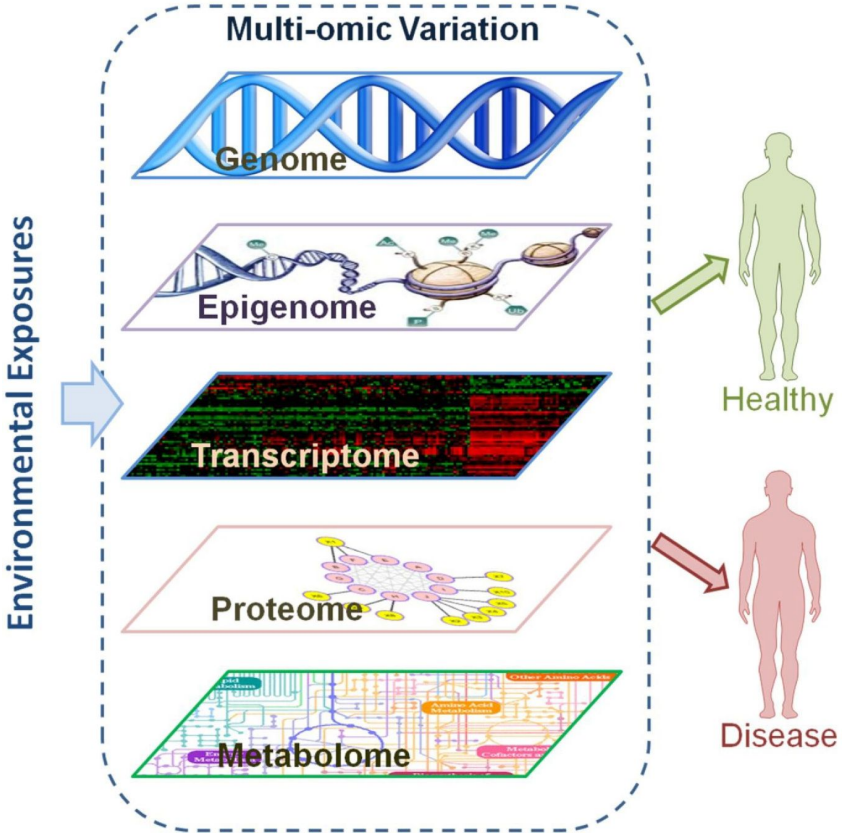
Why Bioconductor Matters

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

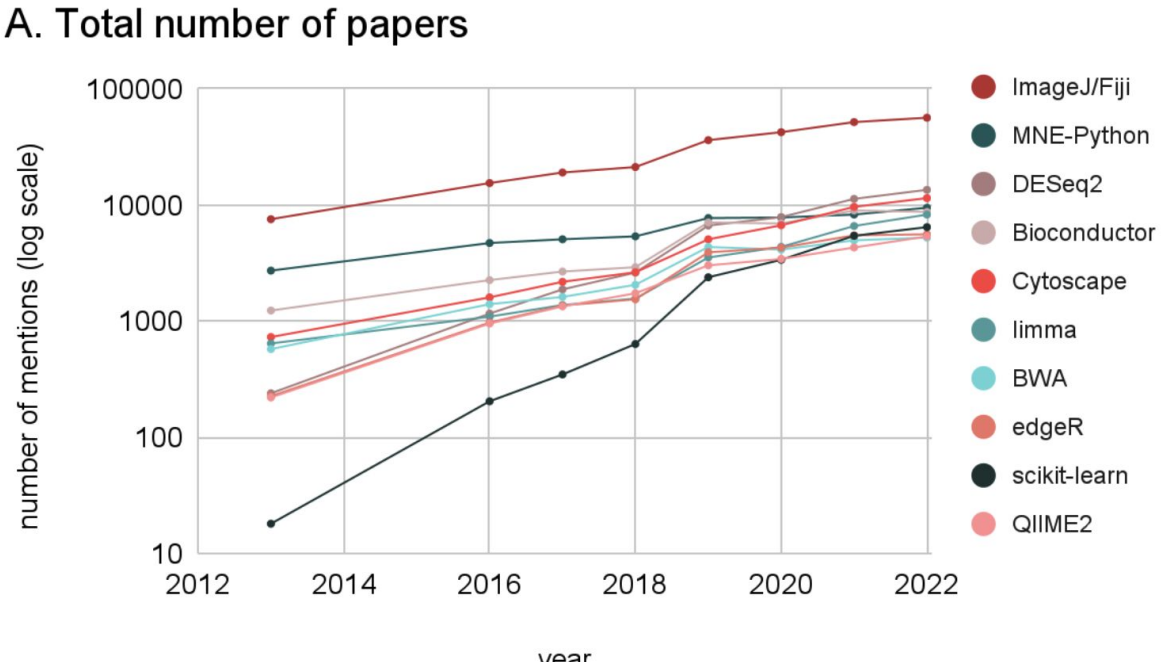
In the past 2 decades, the capacity to measure biology molecules in detail has vastly increased

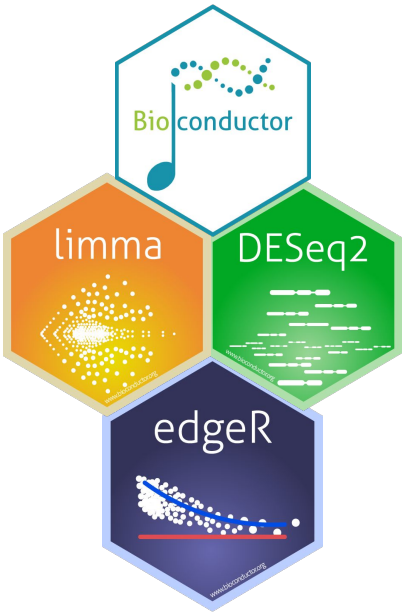
Large volumes of data require advanced, efficient, fast, user-friendly statistical tools

- Advancing biology
- Understanding of disease
- Discovery biomarkers and therapeutic targets
- Impacting Biology, Medicine, Biodiversity, Agriculture, Marine…



Data shows Bioconductor is essential software





4/10 top cited biomedical OSS are Bioconductor

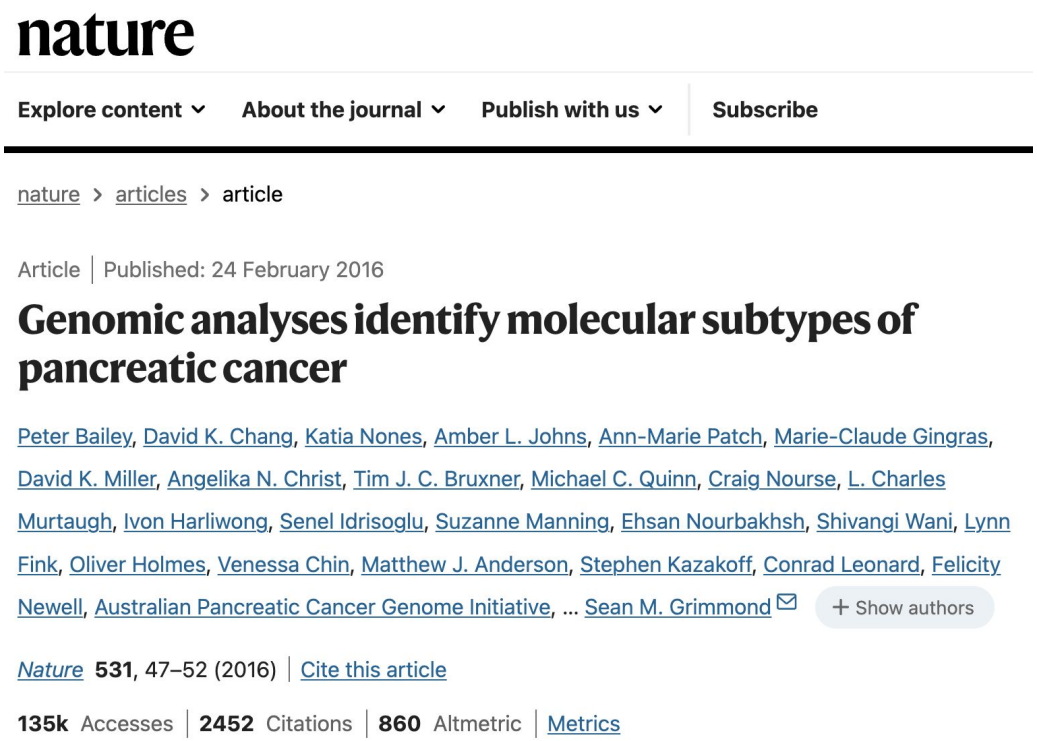
CZI impact report : https://doi.org/10.5281/zenodo.11201216

Bioconductor is used in a wide range of research fields

Bioconductor has impact on health

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Research study using Bioconductor software



nature

Explore content | About the journal | Publish with us | Subscribe

nature > articles > article

Article | Published: 24 February 2016

Genomic analyses identify molecular subtypes of pancreatic cancer

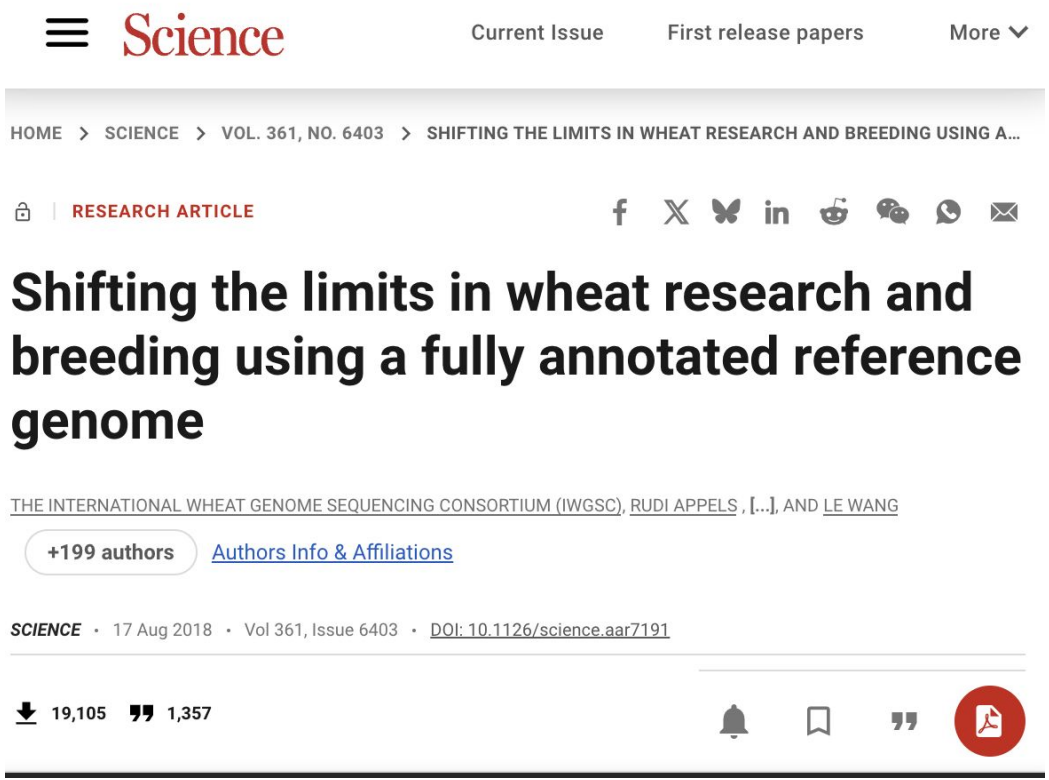
Peter Bailey, David K. Chang, Katia Nones, Amber L. Johns, Ann-Marie Patch, Marie-Claude Gingras, David K. Miller, Angelika N. Christ, Tim J. C. Bruxner, Michael C. Quinn, Craig Nourse, L. Charles Murtaugh, Ivon Harliwong, Senel Idrisoglu, Suzanne Manning, Ehsan Nourbakhsh, Shivangi Wani, Lynn Fink, Oliver Holmes, Venessa Chin, Matthew J. Anderson, Stephen Kazakoff, Conrad Leonard, Felicity Newell, Australian Pancreatic Cancer Genome Initiative, ... Sean M. Grimmond + Show authors

Nature **531**, 47–52 (2016) | Cite this article

135k Accesses | **2452** Citations | **860** Altmetric | Metrics

Bioconductor has impact on agriculture

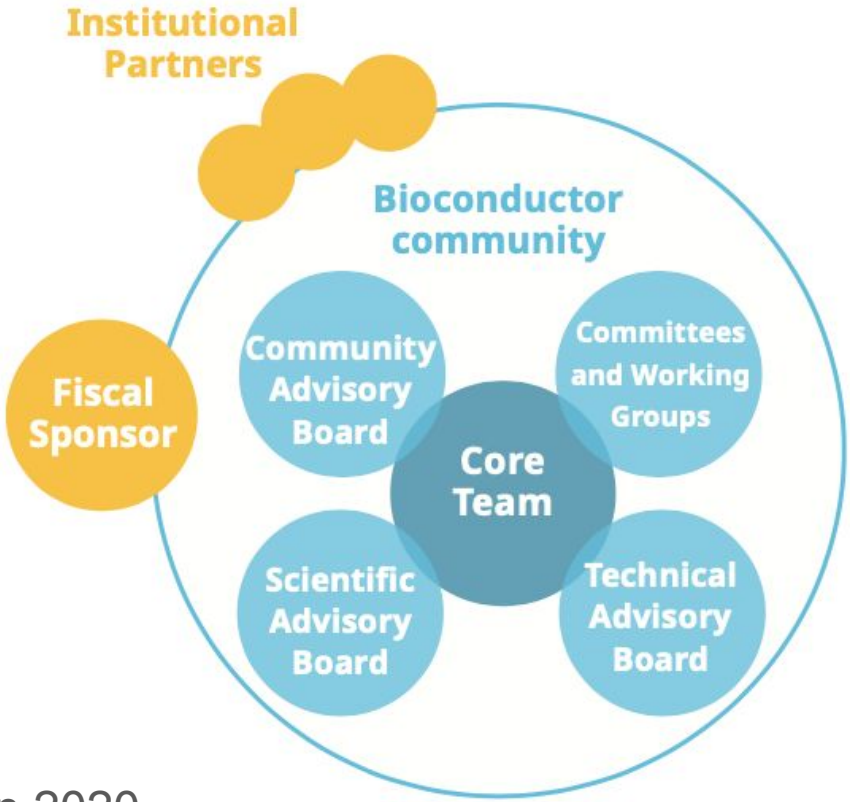
Research study using Bioconductor software



Bioconductor Governance & Community

Community members volunteer to join boards and working groups

- Technical Advisory Board
- Community Advisory Board*
- Working groups and Committees - 50+ members

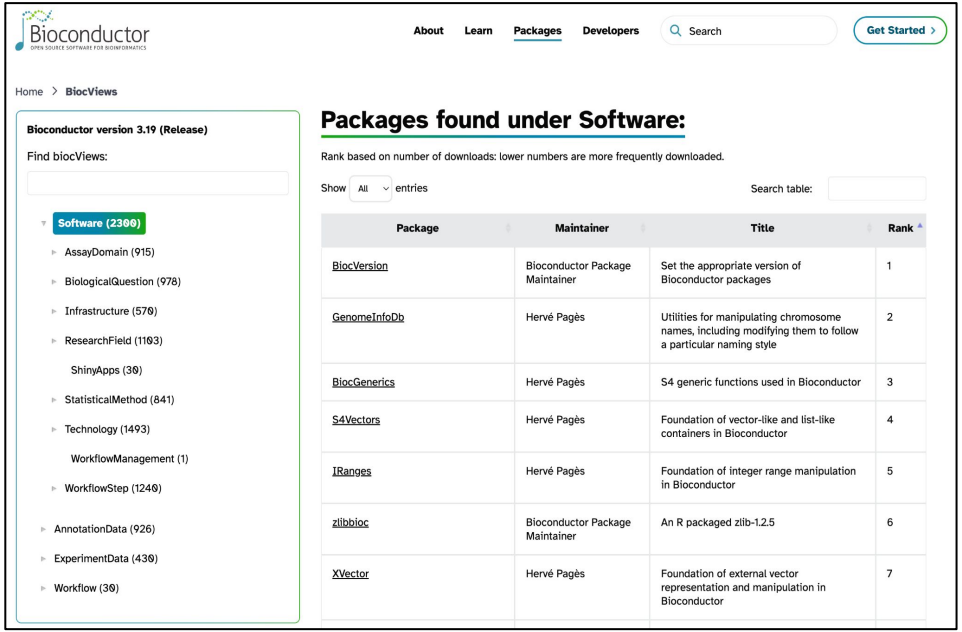


* Co-founded by Aedin Culhane from UL in 2020

Image: CSCCE

A Thriving Ecosystem for Biological Data

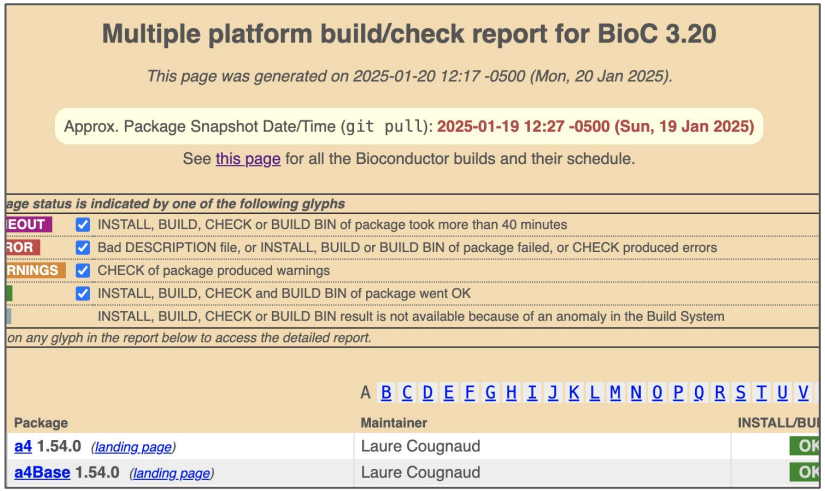
- Supports analysis of many omics (genomics, transcriptomics, proteomics…)
- >2,000 R packages; range of methods Preprocessing, normalization, statistical analysis, and functional annotation
- Easy on-ramp: data, methods, annotation
- Underlying methods in R, C, C++, Fortran, Python, Julia
- Visualization, graphics, interactive tools



Reproducibility, Reusability and Efficiency

- Bioconductor is package **repository** (SVN -> github)
- **Rigorous review** process* for packages (both automated and manual)
- **Daily build and check*** of all packages on multiple OS platforms. Issues reported to developers
- **Standardized data structures** and interoperability
- All packages required to have **documentation** & vignette tutorial with examples*
- FAIR before FAIR

* Stable, Quality, Trusted, Respected



Multiple platform build/check report for BioC 3.20

This page was generated on 2025-01-20 12:17 -0500 (Mon, 20 Jan 2025).

Approx. Package Snapshot Date/Time (`git pull`): **2025-01-19 12:27 -0500 (Sun, 19 Jan 2025)**

See this page for all the Bioconductor builds and their schedule.

age status is indicated by one of the following glyphs

EOUT	INSTALL, BUILD, CHECK or BUILD BIN of package took more than 40 minutes
ROR	Bad DESCRIPTION file, or INSTALL, BUILD or BUILD BIN of package failed, or CHECK produced errors
RNINGS	CHECK of package produced warnings
	INSTALL, BUILD, CHECK and BUILD BIN of package went OK
	INSTALL, BUILD, CHECK or BUILD BIN result is not available because of an anomaly in the Build System

on any glyph in the report below to access the detailed report.

A B C D E F G H I J K L M N O P Q R S T U V

Package	Maintainer	INSTALL/BUI
a4 1.54.0 (landing page)	Laure Cougnaud	OK
a4Base 1.54.0 (landing page)	Laure Cougnaud	OK

Continuous Innovation for Cutting-Edge Research

- Twice yearly releases for cutting-edge tools and methods
- Ensures relevance and usefulness for current research
- Adapts to the evolving field of genomics and biomedical research

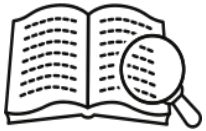
Release	Date	Software packages	R
3.20	October 30, 2024	2289	4.4
3.19	May 1, 2024	2300	4.4
3.18	October 25, 2023	2266	4.3
3.17	April 26, 2023	2230	4.3
3.16	November 2, 2022	2183	4.2

Packages are deprecated if unmaintained. More deprecated packages recently in 3.18-3.20

High-quality Documentation and Training

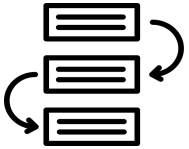
Package function help pages

Books



Package vignettes

Workshops

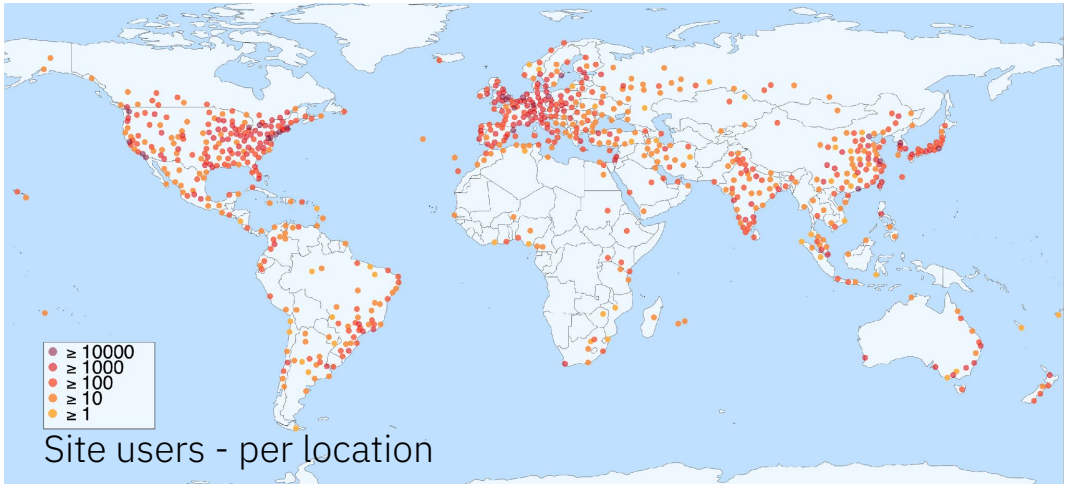


Workflows

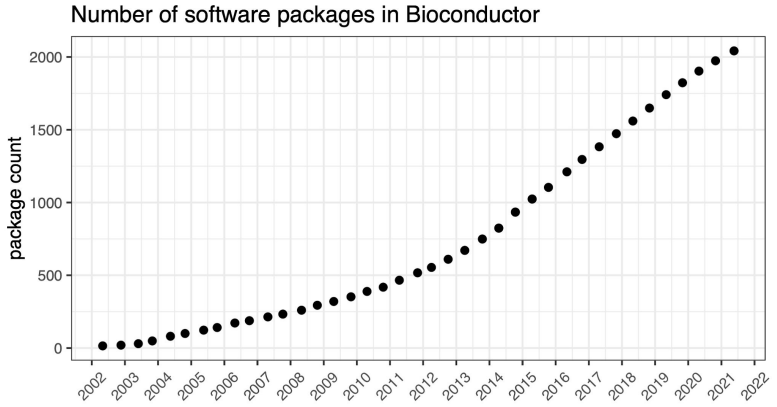
Courses

Bioconductor Training Preprint: https://doi.org/10.48550/arXiv.2410.01351

The Community Makes It Happen



10,000s users

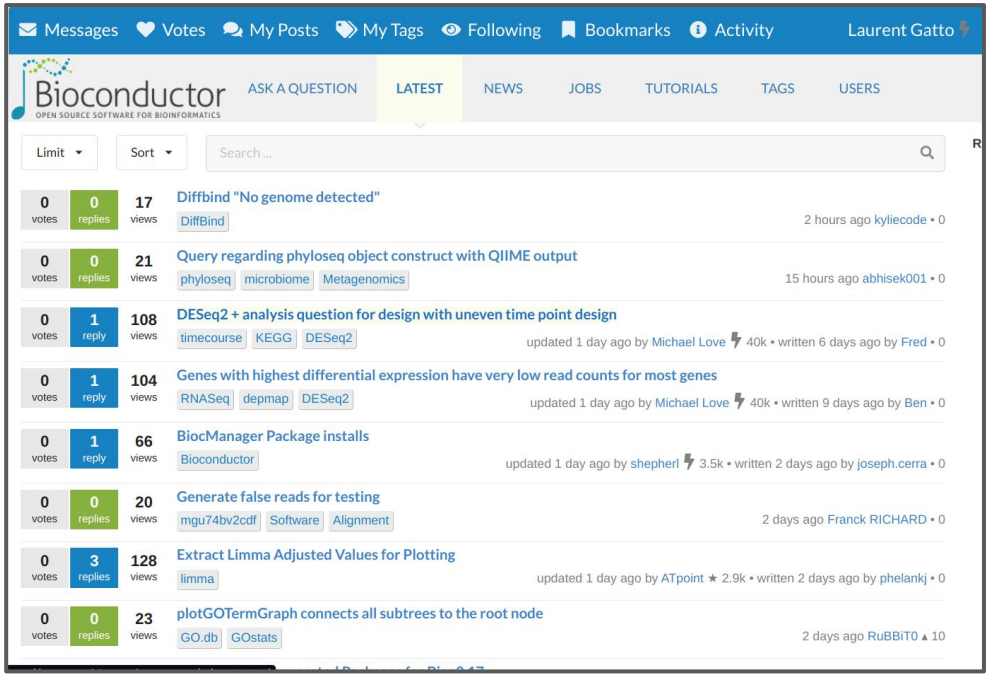


>2,000 packages contributed

Strong Community and Knowledge Sharing: Online forums for users and developers

In 2023:

- **Support Site** (opposite)
  18,297 new users, 1,679 ‘top-level’ posts, and 4,531 comments
- **Bioconductor Slack**
  2,980 members, 100+ channels
- **Developers mailing list**
  2,048 subscribers, 68 posts per month, 28 authors per month
- **GitHub issues**

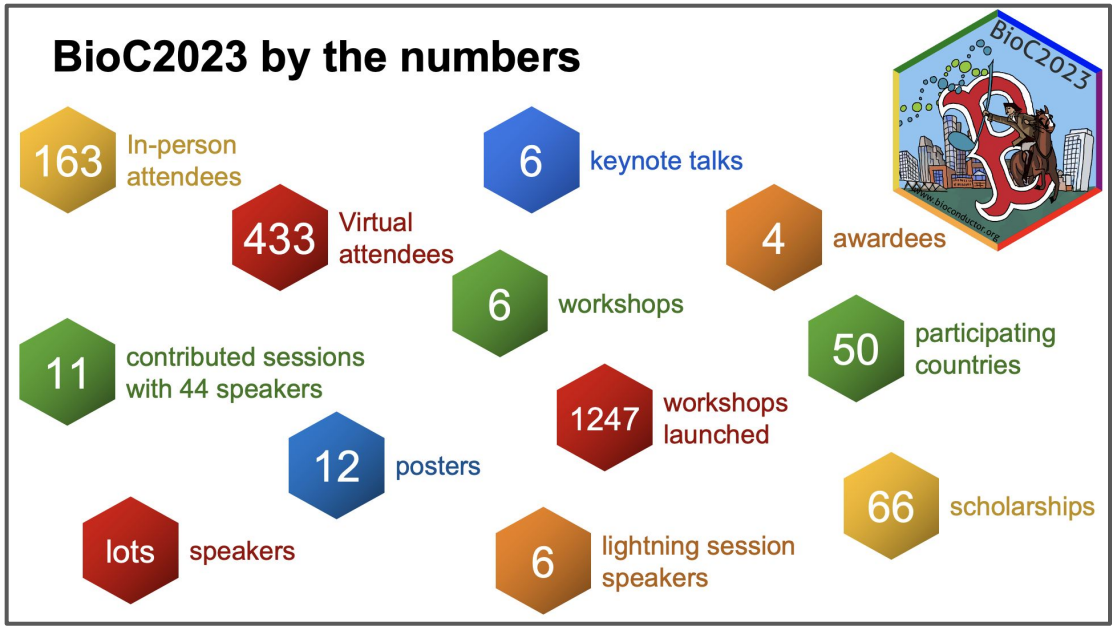


Strong Community and Knowledge Sharing: 3 annual conferences

North American conference
BioC2024,
July 24-26,
Grand Rapids, USA

European conference
EuroBioC2024,
September 4-6,
Oxford, UK

Australasian conference
BioCAsia2024
November 7-8,
Sydney, Australia



CZI Grants supporting Bioconductor Community

CZI EOSS Funding Successes

5 projects funded in cycle 6 (2024-2026)

12 projects funded since 2020 (cycles 1-6)

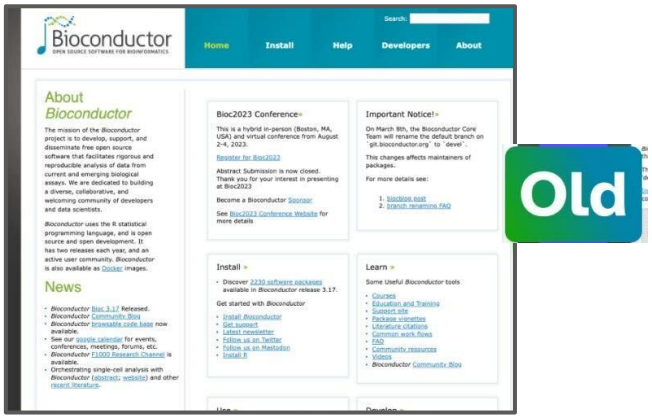
Blog post: Bioconductor CZI EOSS6 grants

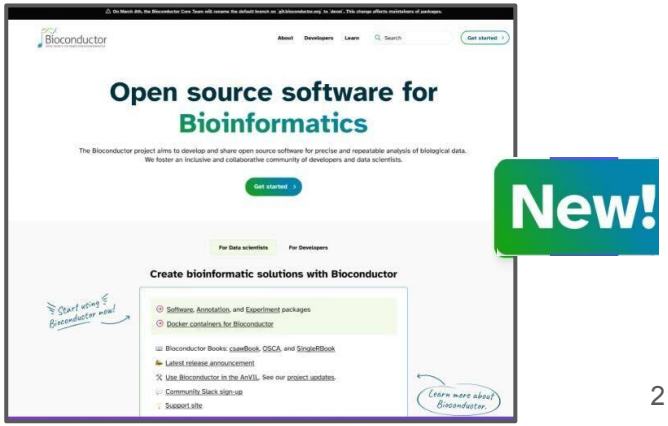
Locations of Bioconductor PIs funded in cycle 6

CZI EOSS4: Bioconductor Website Redevelopment

To improve accessibility for new users and users with disabilities

- 558 community members from 56 countries gave feedback in a survey
- 16 community members participated in 6 working group meetings





Nearform

CZI EOSS4: Bioconductor Website Redevelopment

New design increased accessibility

“It also can use screen reader which I use extensively ; I am just so excited that this new site is more accessible for people with disabilities."

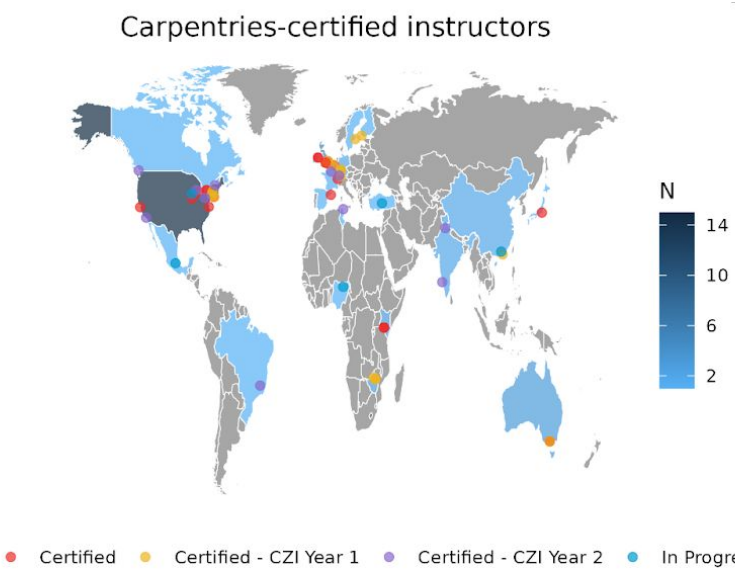
"The clutter-free layout is extremely accessible to me as a colorblind individual."

Wave - Errors + Contrast Errors + Alert (lower is better)			
Page	Existing Design	New Design	Change
Homepage	86	3	-83
About	56	4	-52
Developers	42	3	-39
Install	124	7	-117
Help	94	2	-92

CZI EOSS4: Bioconductor Carpentry Global Training Program

Bioconductor became a Carpentries member organisation in August 2022

- 30+ instructors trained over 2 years
- Workshops in North America, Europe, and Asia to 144+ registrants



CZI EOSS4: Bioconductor Carpentry Global Training Program

*“Swapping the chilly embrace of Swedish winter for the familiar heat of my hometown in the **Brazilian Amazon Forest** was like a long warm hug.*

***Yet, what made this homecoming truly special was the chance to give back to my community, to share some of the knowledge I'd amassed over the last years**.*

Overall, as my first workshop ever organized, teaching turned out to be a more gratifying experience than I expected and I am already planning for the next one.”

CZI EOSS6: new grants supporting Bioconductor community

Delivering High-Quality Bioconductor Training for a Worldwide Community

- **PIs**: Aedin Culhane and Maria Doyle
- **Collaborators:** Laurent Gatto (Institut de Duve), Charlotte Soneson (Friedrich Miescher Institute for Biomedical Research), Kozo Nishida (Tokyo University of Agriculture and Technology), Trushar Shah (International Institute of Tropical Agriculture), Umar Ahmad (Bauchi State University), Zedias Chikwambi (African Institute of Biomedical Sciences and Technology)

Supporting and Sustaining Bioconductor Developers

- **PIs:** Maria Doyle and Aedin Culhane
- **Collaborators:** Lori Shepherd (Roswell Park Comprehensive Cancer Center), Vince Carey (Harvard Medical School), Robert Shear (Harvard Medical School), Sean Davis (University of Colorado Anschutz School of Medicine)

Q & A

Thank You!

Join our community or share Bioconductor with your networks

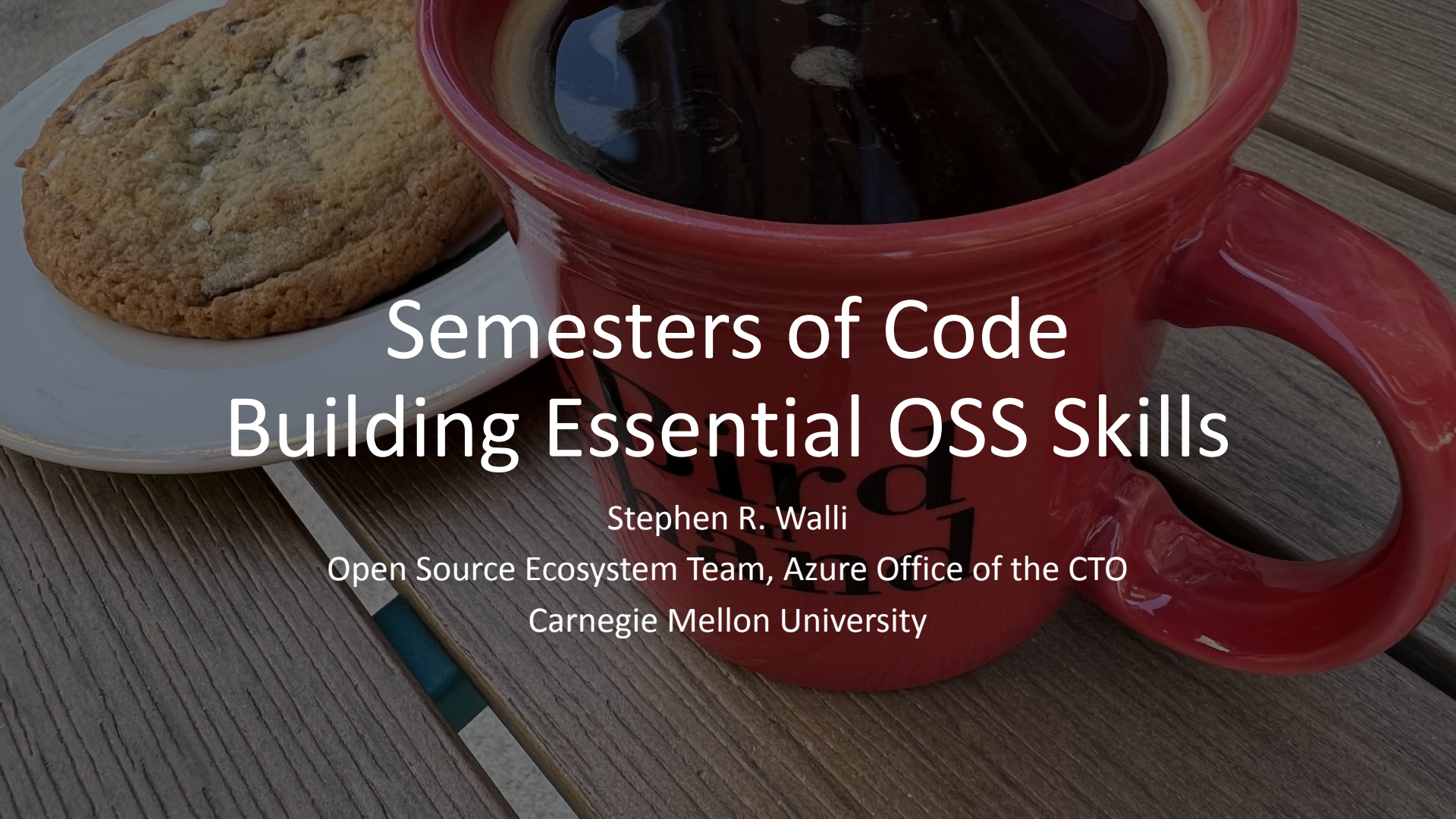
Learn more: www.bioconductor.org

Email: maria.doyle@ul.ie

National
Open Source
Innovation
Summit

Stephen Walli

Principal Program Manager, Microsoft OSPO



Semesters of Code
Building Essential OSS Skills

Stephen R. Walli
Open Source Ecosystem Team, Azure Office of the CTO
Carnegie Mellon University

A history of collaborating in standards & OSS & nonprofits with other engineers

- IEEE Founding member P1003 Project Management Committee (1990-1999)
- Founder and VP Engineering in a VC-backed ‘open source’ startup (1995-1999) acquired by Microsoft in an asset acquisition in 1999
- Technical Director, Outercurve Foundation (2010-2013)
  - ”The Rise and Evolution of the Open Source Software Foundation” (with Paula Hunter)
- Linux Foundation
  - Microsoft Board & Board alternate for a number of years
  - Founding member, Open Container Initiative, (2015-2017)
  - Founding member, Board Chair, Confidential Computing Consortium, (2019-2022)
  - Founding member, eBPF Foundation, (2021-2022)
  - Mentoring Microsoft members in LF Energy, the Green Software Foundation, …
  - Member of the OpenSSF Governance Committee (2023-2024)
- Eclipse Foundation
  - Microsoft Board member (2021-2024)
  - Founding member, Steering Committee Chair, Software Defined Vehicle WG, (2021-2022)
- IEEE WG Chair, P3190, Recommended Practices for OSS Projects (2022)
- Along the way ISO, ECMA, OCP, LF Cloud Foundry, OpenStack, LF Open Manufacturing Platform (and JDF), and OMG/Digital Twin Consortium …

The Problem

“Software is eating the world.”

—Marc Andreesen, 2011

2012 Octoverse	**4.6M+ repositories**
2016 Octoverse	**19.4M+ repositories**
2018 Octoverse	**96M+ repositories**
2019 Octoverse	**140M+ repositories**
2020 Octoverse	**200M+ repositories**

“We are drowning in software, most of it mediocre, duplicative, and bad.”

—Not Marc Andreesen

The Industry Problem Statement

1. There is a startling lack of understanding of open source software in production
2. The knowledge of software engineering practice is being diluted and narrowed to tool practices across the industry (when anyone can be a net producer of software now)

AND

3. Software ‘maintenance’ (sustainability, security*, …) is an accelerating problem

*Software Transmitted Disease is the new new STD

The A-ha Moment!

- Well-run open source licensed projects are natural labs for software engineering experience
- What if we create an undergraduate course that taught:
  - Basic Software Engineering Theory
  - Healthy Open Source Software Project Practices
  - Intellectual Property Basics for Software Engineers
- Then create student projects in active open source licensed software projects with mentors as the lab/homework for the course
- Semesters of Code is born!

There is educational history …

- 20,000 students have come through Google Summer of Code
- Academics have taught “open source” for 15-20 years

Johns Hopkins University
The First Experiments

Choose Your Own Open Source Adventure

Johns Hopkins Intersession – January 2021 (COVID Lockdown)

- Lab 1: Build 125 year's worth of software value in an hour (httpd)
  https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/labs/1#lab-1-build-125-years-worth-of-software-value-in-an-hour-or-so
- Healthy open source software projects (On-ramps and practices)
  https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/lessons/3
- Lab 2: Evaluating Projects (Perl, Python, Node, Semester.ly)
  https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/labs/2#lab-2-evaluating-projects

The First Course (Fall 2021, 2022)

EN.601.270 Open Source Software Engineering (Semesters of Code I) (E, 3 credits)

The course will provide students a development experience focused on learning software engineering skills to deliver software at scale to a broad community of users associated with open source licensed projects. The class work will introduce students to ideas behind open source software with structured modules on recognizing and building healthy project structure, intellectual property basics, community & project governance, social and ethical concerns, and software economics.

The practical side of the course will engage and mentor students directly in OSI-licensed project communities to provide hands-on learning experiences of practices covered in the classroom modules, and team building experience working in the project.

Prerequisites: EN.601.220 Intermediate Programming & EN.601.226 Data Structures (see appendix)

Time: TuTh 10:30-11:45a ET

Limit: 35, CS majors only

Sizing Student Projects

Assumptions:

- 30-45 students
- Projects:
  - Fall 2021: PASS, Lutece, Powershell, Semester.ly, OpenCRAVAT
  - Fall 2022: PASS, Lutece, Powershell, enarx, OHDSI (“Odyssey”), .NET, PatternFly
- 2+ mentors define 5+ student projects per open source project
- There are ~14 weeks (includes a reading week break)
- 4-5 hours/week of student time **means ~50-70 hours per student project**

Ideas for Student Projects

- **Low-hanging fruit:** These projects require minimal familiarity with the codebase and basic technical knowledge. They are relatively short, with clear goals.
- **Fun/Peripheral:** These projects might not be related to the current core development focus but create new innovations and new perspective for your project.
- **Core development:** These projects derive from the ongoing work from the core of your development team. The list of features and bugs is never-ending, and help is always welcome.
- **Infrastructure/Automation:** These projects are the code that your organization uses to get its development work done; for example, projects that improve the automation of releases, regression tests and automated builds. This is a category in which a student can be really helpful, doing work that the development team has been putting off while they focus on core development.
- **Risky/Exploratory:** These projects push the scope boundaries of your development effort. They might require expertise in an area not covered by your current development team. They might take advantage of a new technology. **There is a reasonable chance that the project might be less successful, but the potential rewards for everyone make it worth the attempt.**

The General Evaluation

- **50% The Student Project**
  - **25% for each half term (25 October)**
  - **The first half term grade can't be improved**
- 20% x2 in class mid-terms
- 10% In Class Participation
  - Discussions around regular short reading assignments (most weeks)
  - There's one simple homework around project evaluations

The Mid-term Mentor Evaluation

- Did they get to a successful build in a reasonable way/time?
- Do they check in regularly via whatever project channels exist?
- Do they show progress towards goals (even artificial sub-goals) at a regular cadence?
- Are they accomplishing work – not just about the learning process, do they get work done?
- Does it feel like they will reach the original goal by term's end? (Have expected outcomes changed?)
- Other notable observations
- Excellent? Good? Needs Improvement?

Semesters-of-Code Is Not GSoC

- It is a single course in a normal school semester – not a summer job
- GSoC plans for ~150 hours per student over the summer for a student project – we have a reasonable expectation of ~50-70 hours
- GSoC projects (being larger) tend to be 1:1 mentor to student project
- GSoC has a longer student matching period & ramp up period for larger student projects
- GSoC mentors expect to spend about 2-3 hours a week for each student project – GSoC projects are bigger in scope
- Google pays students stipends – a key student outcome is a check!

New in Fall 2022

- Doubled the student population from 20 to 40
- Experimenting with concurrent distance learning by adding students at University of Galway (with local administrative support)
- Added more student projects from industry (as opposed to research)
- Experimenting with multiple students working in the same student project (side-by-side, not directly collaborating)
- Moving to an active learning style in the classroom

Pedagogy

The Basics (from a Professional Teacher)

- Structuring of the lessons and lesson plans
- Every lesson has a set of learning objectives (which is very useful when building tests)
- Building the course curriculum flow
- Building the rubric and thinking about grading

All Learning Is Social Learning

- Lave, ‘Cognition in Practice’, 1988
- Lave & Wenger, ‘Situated Learning’, 1991, (Five Apprentice Situations Studied)
- **Legitimate Peripheral Participation Requirements**
  - Legitimate – Real opportunity for learning
  - Peripheral – The learner was outside learning their way in-group
  - Participation – The learning strategy involved the learner doing something

So, apprenticeship and mentorship

- Eventually leads to Wenger’s ‘Communities of Practice’ work, 1998

Active Learning & Carl Wieman

"Nearly all techniques labeled as active learning include those features known to be required for the development of expertise; in this case, thinking like an expert in the discipline. The **active learning methods are designed to have the student working on tasks that simulate an aspect of expert reasoning** and/or problem-solving **while receiving timely and specific feedback from fellow students and the instructor** that guides them on how to improve."

… **So, mentorship and social learning**

Large-scale comparison of science teaching methods sends clear message (2014)
www.pnas.org/cgi/doi/10.1073/pnas.1407304111

Learnings after the First Two Iterations at Hopkins

- Scaling for more students means scaling mentors & student projects
  - Scaling the mentoring workshops to set the bar
  - Automate student project onboarding and student project matching
  - Smooth the student on-ramp into the open source project
- Scaling for more schools/programs means scaling instructors
  - There is still ongoing curriculum tuning
  - Building a pipeline of industry-based instructors & instructor workshops
- **Mentors notice a growing problem with student time commitments**
  - **Students all have 4-5 other courses banging for their attention**
  - **70-80 hours is normal for project work and students are starting to struggle**

Carnegie Mellon University
Evolving the Experiments

Summer 2023 “Internship” Course

- A set of students lost their internships in the economic downturn
- We created a full internship experience within CMU SCS S3D
- Working with OpenStack (OIF) and a couple of Pittsburgh area startups we defined large projects for teams of 4-5 students to tackle as a team
- Students are each working as a team 20-40 hours per week (for 12 weeks)
- 2 project mentors per team, and weekly coaching time with the instructors
- LOTS of curriculum tuning for more active learning opportunities
- Student teams present status out to class regularly through summer
- CMU co-teaches classes (I get to learn with a professing professional!)

Remarkably Better Student Outcomes!

- Undergraduate students can solve big problems together with steep learning curves and complex toolchains
- Students mentor one another (without guidance) on the tool chain complexity! THIS WAS SURPRISING & APPARENTLY BACKED BY SCIENCE*
- Student confidence grows dramatically through the summer
- Mentors are excited tackling “unbudgeted” work in their open source projects
- Re-ran the course Summer 2024 with CMU-Qatar in a 10-week format,OpenStack & Eclipse projects, front packing the lessons in the first 6 weeks, and full remote project work for the last 4 weeks **with consistent excellent student outcomes†**

*Leveraging Learning Collectives: How Novice Outsiders Break into an Occupation, Ece Kaynak, 2023 https://doi.org/10.1287/orsc.2020.14214

† A CMU student team write-up from Summer 2024 on LinkedIn, and their final presentation at the close of course (16-min video).

Why is this exciting?

Why Are These Ideas Exciting?

- Peter Naur, “Programming as Theorem Building”, 1985 https://doi.org/10.1016/0165-6074(85)90032-8
- Jean Lave, “Cognition in Practice: Mind, Mathematics and Culture in Everyday Life”, 1988
- Jean Lave & Etienne Wenger, “Situated Learning: Legitimate Peripheral Participation”, 1991
- Carl Wieman’s work at Stanford and UBC in 2000s
- Leveraging Learning Collectives paper (previous slide)
- Successful modern project and non-profit cultures in open source and standards (IETF, IEEE, Eclipse, OpenStack)

Peter Naur and ‘Programming as Theory Building’, 1985

http://pages.cs.wisc.edu/~remzi/Naur.pdf

“… the programmer’s knowledge transcends that given in documentation …”

1. The programmer knows how the real-world maps to the program, and which parts of the world are relevant to the program or not.
2. The programmer can explain all design decisions. “The justification is and must always remain the programmer’s direct, intuitive knowledge or estimate.”
3. The programmer knows how best to modify the program to meet new requirements. This depends on recognizing similarities between new and old situations.

The Consequences If Naur Is Correct Are Everywhere

- RTFM … as long as the manual is current/correct
- “Comment your program” vs “comments can’t be trusted”
- “Your comments should reflect and explain your design”
- “When a design debate ends, the document is a record of when the shooting stopped, not a clear explanation.” [It’s a ceasefire line.]
- Literate programming systems (Knuth and programs as literature) [1990s]
- Formal methods (Z notation/VDM/TLA+ and programs as math) [1990s]
- TDD [2003] is predicated on a re-statement of the program’s theory as test assertions developed before the program
- Naur: If you don’t understand the theory of the program, changes create debt, until the program becomes unmaintainable – you create the proverbial Big Ball of Mud.

Naur's Solution

"**What is required is that the new programmer has the opportunity to work in close contact with the programmers who already possess the theory….** This problem of education of new programmers in an existing theory of a program is quite similar to that of the education problem of other activities where the knowledge of how to do certain things dominates over the knowledge that certain things are the case, such as writing and playing a music instrument. **The most important educational activity is the student's doing relevant things under suitable supervision and guidance."**

i.e., Legitimate Peripheral Participation, and Jean Lave's pedagogical theory that all learning is social learning

All **Software** is Social Software

- Consider the well-run, OSI-licensed project communities you know
- Project governance is a practice – not just its documentation
  - The development process through PRs and Issues/email is mentorship
  - The release/delivery process is a social practice of managing the release
  - Engaging new community members through forums, meet-ups, email is a social practice
  - Running a nonprofit to remove risk requires establishing social norms beyond the charter for how decisions in committee happen so as to support the project work

N.B. This isn't just about OSI-licensed software project communities

Three On Ramps for Community Building

How do you encourage people to use your project?

(Because that's where you'll find bugs reports & tutorials & developers)

(How do you make it easy to install/configure/use the software?)

How do you encourage people selfishly to experiment?

(Because these are your future contributors)

(How do you make it easy to build/test/experiment?)

How do you encourage people to share their work?

(Because contribution flow is the growth and success of your project)

(How do you make it easy to contribute?)

Three On Ramps for Community Building

How do you teach/mentor people to use your project?

(Because that's where you'll find bugs reports & tutorials & developers)

(How do you make it easy to install/configure/use the software?)

How do you teach/mentor people selfishly to experiment?

(Because these are your future contributors)

(How do you make it easy to build/test/experiment?)

How do you teach/mentor people to share their work?

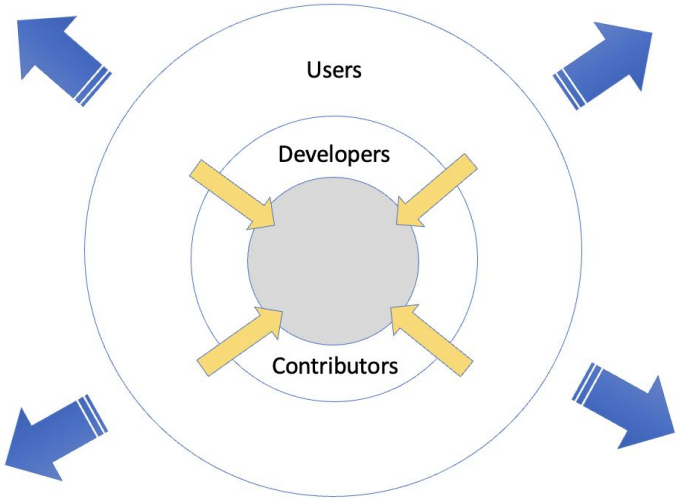
(Because contribution flow is the growth and success of your project)

(How do you make it easy to contribute?)

Frameworks for Building Software/Community/Nonprofit

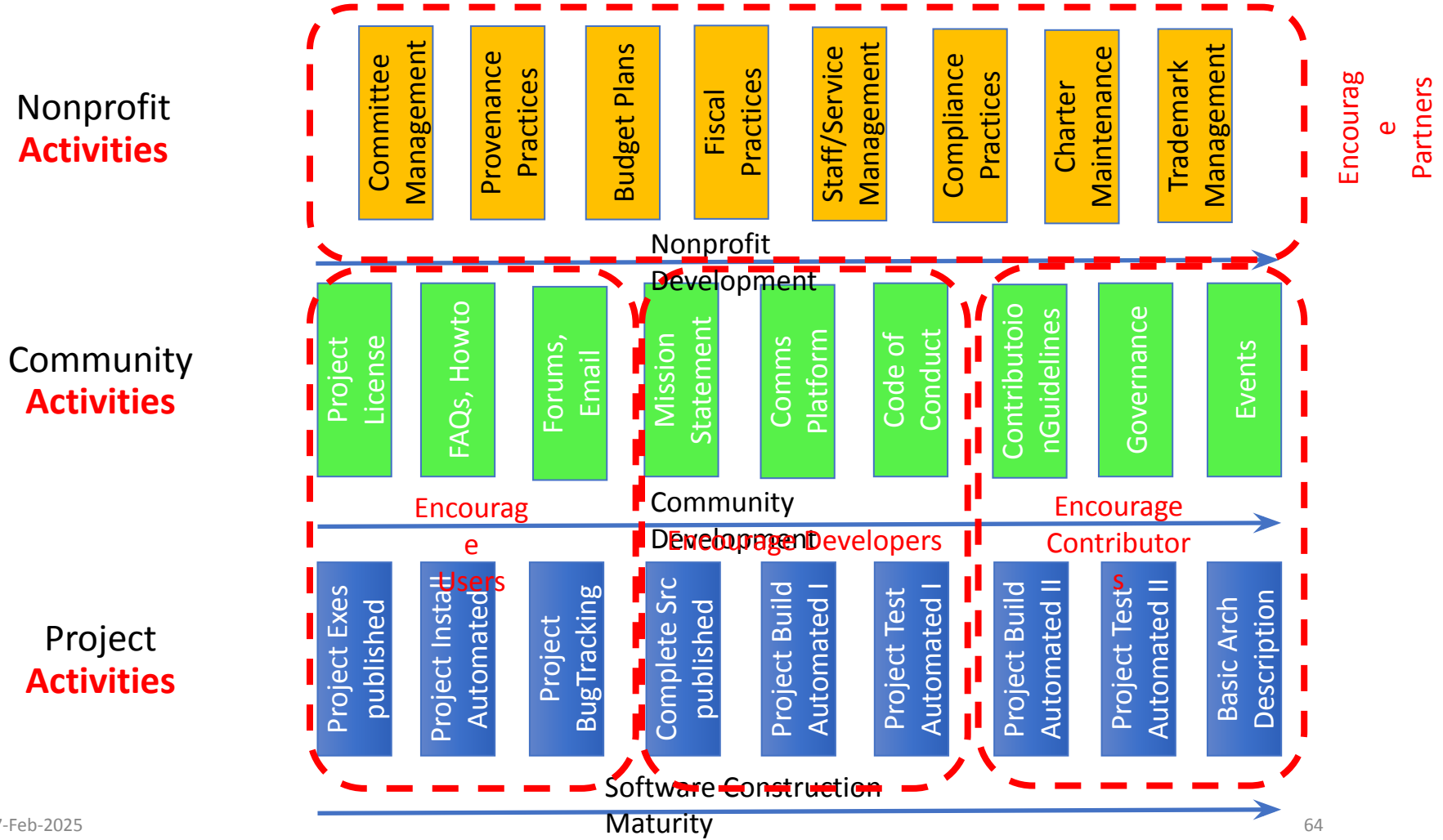
Maintainers
+
Working
Code

Building the Software
(Sharing Innovation Outbound)



Building the Community
(Capturing Innovation Inbound)

Building the Non-profit
(Remove Risk, Hold Assets, Collect and Distribute Funds, Anchor the message)


Nonprofit Activities
Community Activities
Project Activities
Committee Management
Provenance Practices
Budget Plans
Fiscal Practices
Staff/Service Management
Compliance Practices
Charter Maintenance
Trademark Management
Encourage Partners
Nonprofit Development
Project License
FAQs, Howto
Forums, Email
Mission Statement
Comms Platform
Code of Conduct
Contributoio nGuidelines
Governance
Events
Encourage Users
Community Development
Encourage Developers
Encourage Contributors
Project Exes published
Project Install Automated
Project BugTracking
Complete Src published
Project Build Automated I
Project Test Automated I
Project Build Automated II
Project Test Automated II
Basic Arch Description
Software Construction Maturity

Continuing the Evolution

- Continuing the university experiments
- Software Engineering Bootcamps – Stepping outside tradition university settings (CMU-Africa, Co-Develop, Open Source Community Africa, etc.)
- Bringing the Summer Internship in-house – secure software practice!

Q&A