

# BioC 2014

## Where Software and Biology Connect

Dana-Farber Cancer Research Institute  
Boston, MA

July 30 - August 1

# Welcome!

# Thanks

- ▶ Erica Feick, Shantel Owusu (DFCI)
- ▶ Melissa Alwendia (FHCRC)
- ▶ Sponsors: Rstudio, Revolution Analytics, Genentech, Novartis, Fred Hutchinson Cancer Research Institute, Dana-Farber Cancer Institute.
- ▶ *Bioconductor* team: Sonali Arora, Marc Carlson, Nate Hayden, Valerie Obenchain, Hervé Pagès, Paul Shannon, Dan Tenenbaum
- ▶ Speakers and workshop presenters
- ▶ Program committee: Vincent Carey, Sean Davis, Robert Gentleman, Kasper Hansen, Wolfgang Huber, Rafael Irizarry, Michael Lawrence, Levi Waldron
- ▶ NIH / NHGRI U41HG0004059
- ▶ ...and the *Bioconductor* community!

# Logistics



# Agenda

- ▶ <http://bioconductor.org/help/course-materials/2014/BioC2014/developer-day/>

## Morning

- ▶ Project updates
- ▶ Group activity
- ▶ Flashlight talks

## Afternoon

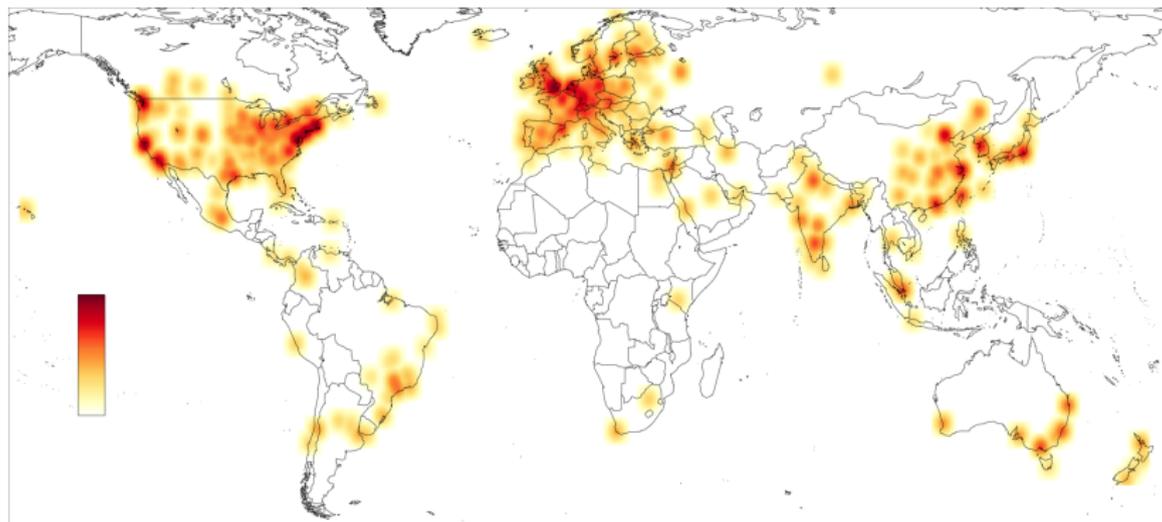
- ▶ Community needs
- ▶ Developer workshops
- ▶ Developer Day Address – George Ostrouchov

# Introductions

# Project Status

- ▶ Packages: 65 (2.12) + 84 (2.13) new (2-3 packages per week!); 854 in current 'devel'
- ▶ Citations: PubMedCentral full-text citations to 'Bioconductor': 9547 (1702 in the last year)
- ▶ Package downloads: 9.5M to 242,000 distinct IP addresses. *BiocInstaller*: 291,000 to 124,500; *DirichletMultinomial*: 3,190 to 1,636
- ▶ Web visits: 1M sessions from 409,000 visitors in the last year; 16 - 26% more than previous year

# Global user base



# What is *Bioconductor*?

Analysis and comprehension of high-throughput genomic data

- ▶ Statistics, annotation, visualization
- ▶ Domains: sequencing, microarrays, flow cytometry, proteomics, images, ...

And also...

- ▶ Programming with data
- ▶ Enrich for better programming practices – versioning, correctness, documentation
- ▶ Reproducibility

# Why is *Bioconductor*?

## Why S4?

- ▶ (user) Manage complex data, coordinated and error-free
- ▶ (user) Enable interoperability between packages
- ▶ (developer) Avoid re-inventing the wheel
- ▶ *Re-use existing classes*

## Why release / devel structure?

- ▶ (user) Stable experience
- ▶ (developer) Flexibility to innovate
- ▶ *As user or developer, NEVER mix release and devel branches*

## Why a centralized (svn) repository?

- ▶ (user) Definitive, long-lived location of packages
- ▶ (developer) Best practices for code management
- ▶ (project) Coordinate packages into stable, tested releases.
- ▶ *Keep your github / other repository in close sync with svn*