

Systems Genetics Tools

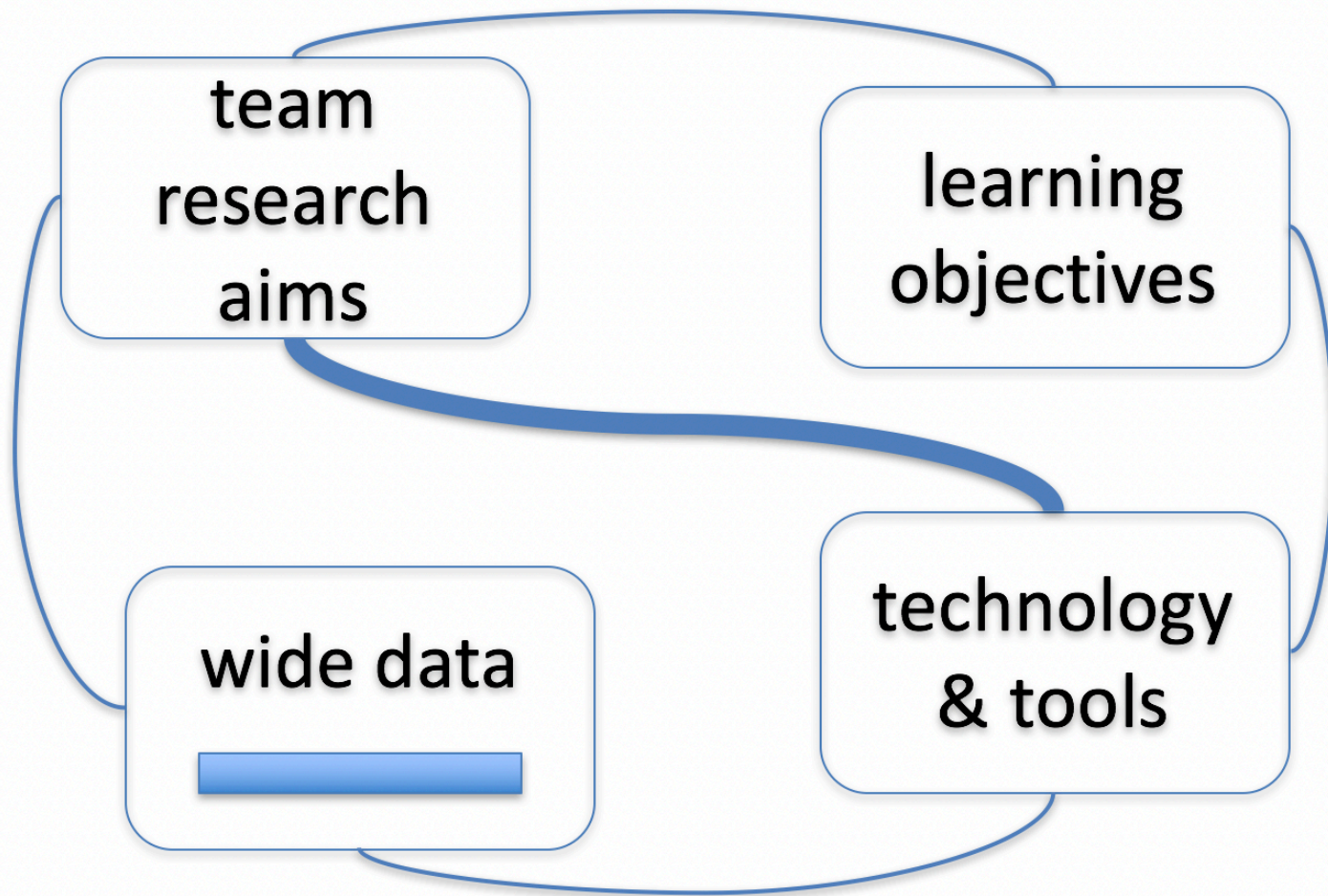
Brian S. Yandell, UW-Madison

January 2017

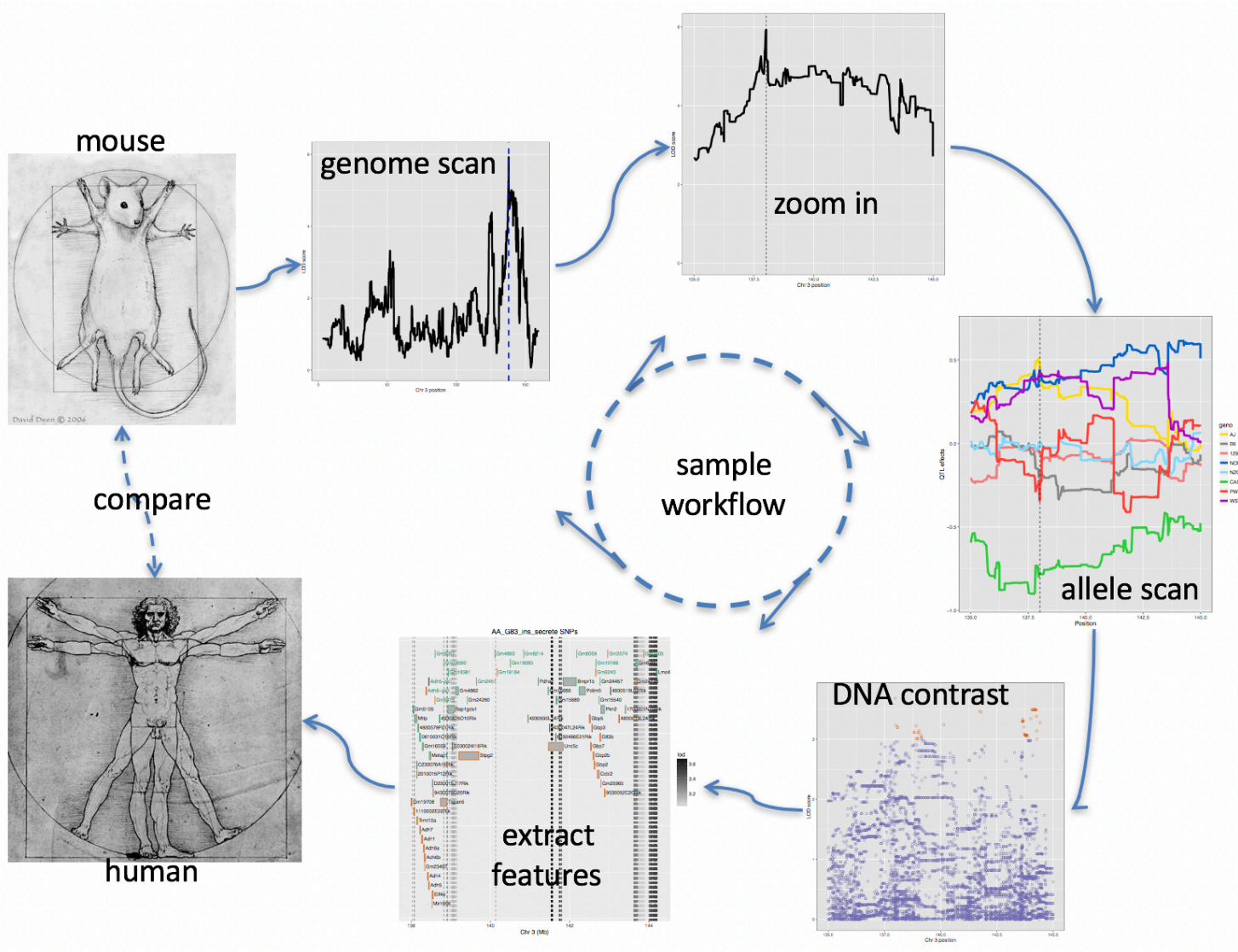
appeal of new technology

- Better, faster, smarter, ...?
 - but does more yield better insight?
- How to manage data volume
 - keep track of it all
 - decide what is important
- How (or why) to use more complicated models?
 - need to develop new analysis tools
 - want to find new ways to visualize
- Be involved in process of data inquiry
 - share with colleagues (in real time)
 - reach beyond domain boundaries

tools & workflow: big idea



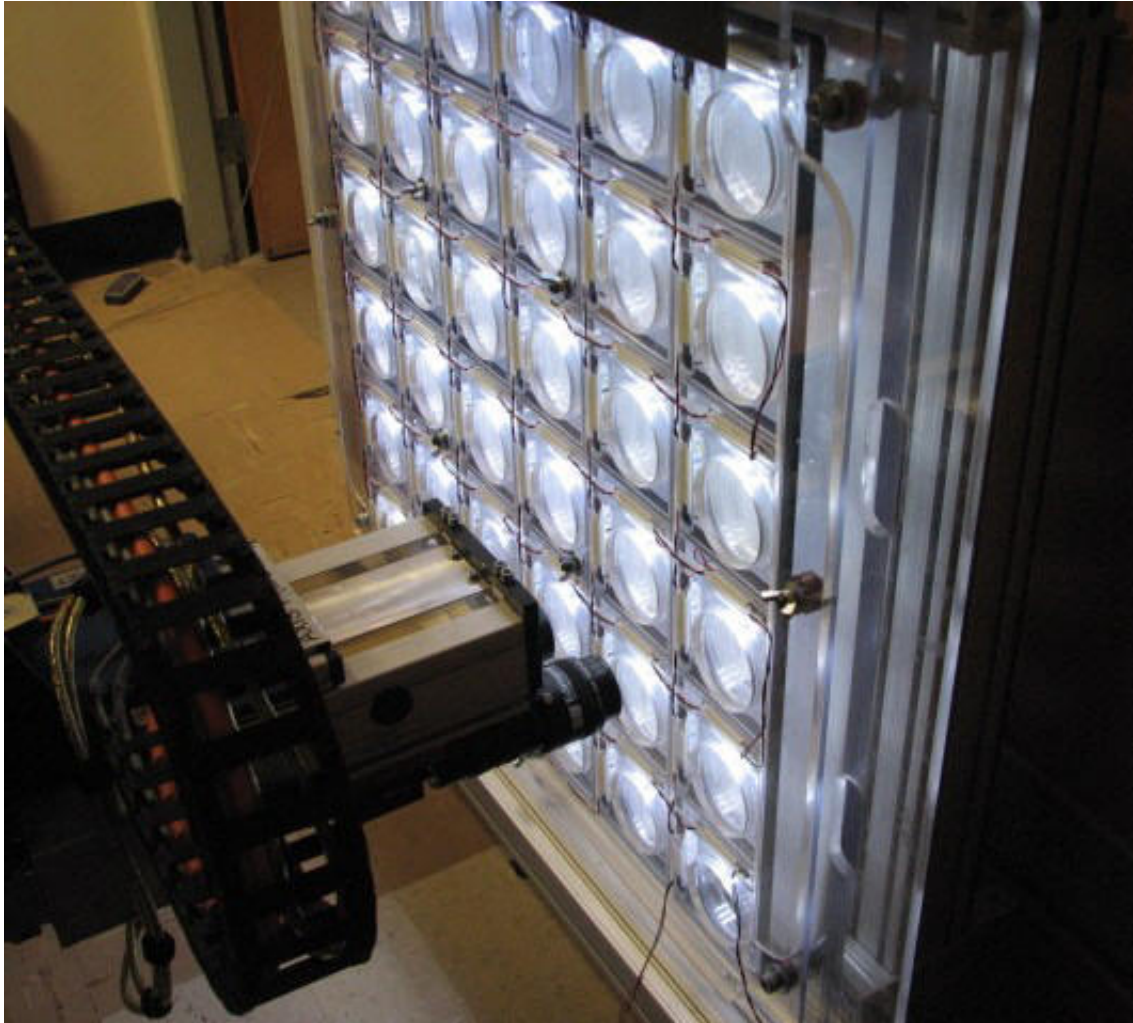
team research aims



technology tools and platforms

- types of tools
 - algorithms: software tools
 - machines: hardware tools
- Your laptop
 - powerful for mid- to large-size projects
 - powerful communication tool
- Scaling up
 - Massive data: storage, management, access, sharing
 - Workflow steps: engine/CPU, organizing, scheduling

data recording hardware



<http://phytomorph.wisc.edu> (Edgar Spalding Lab)

sharing with authentication

- Tools: Box, Google, GitHub
 - data, methods, ideas, results
- Research community
 - [Arabidopsis](#)
 - [Switchgrass](#)
 - [Maize](#)
- Cross-species systems
 - DNA sequence (genomics)
 - Phe-gen relationship (QTL, systems genetics)

technology considerations

- Your time
 - familiarity with tools
 - familiarity with data
- Comparing results
 - model selection on one dataset
 - several methods on one data set
 - multiple data sites

data visualization

- genotype diagnostics
- Distribution at locus
- Scatterplots with symbols (QTL, env)
- Genome scans
 - LOD profile
 - Allele scans
 - SNP scans (GWA Manhattan plots)
- multiple traits
 - over time or space (Moore)
 - networks (small, not hair balls)
 - box- or dot-plots over conditions

evolution of computational tools

Advances in measurement, design and analysis would be academic without advances in computational technology.

- faster machines -> faster throughput of more stuff
- methods translated into algorithms
 - open source: freely distributed, easy to study
- standalone programs
- packages in language systems (R or Python or Matlab)
- interconnectivity of algorithms and data resources

collaboration systems

dangers of email-based collaboration

- trading large files back and forth (slow, not secure)
- nearly impossible to keep track of versions
- minor updates require repeat sending

modern approach: use email to notify collaborators only

- [GitHub](#) to share code & ideas with version control
- Box/DropBox & Google Drive to share documents
- iPlant to improve data access & processing efficiency

emerging collaboration systems

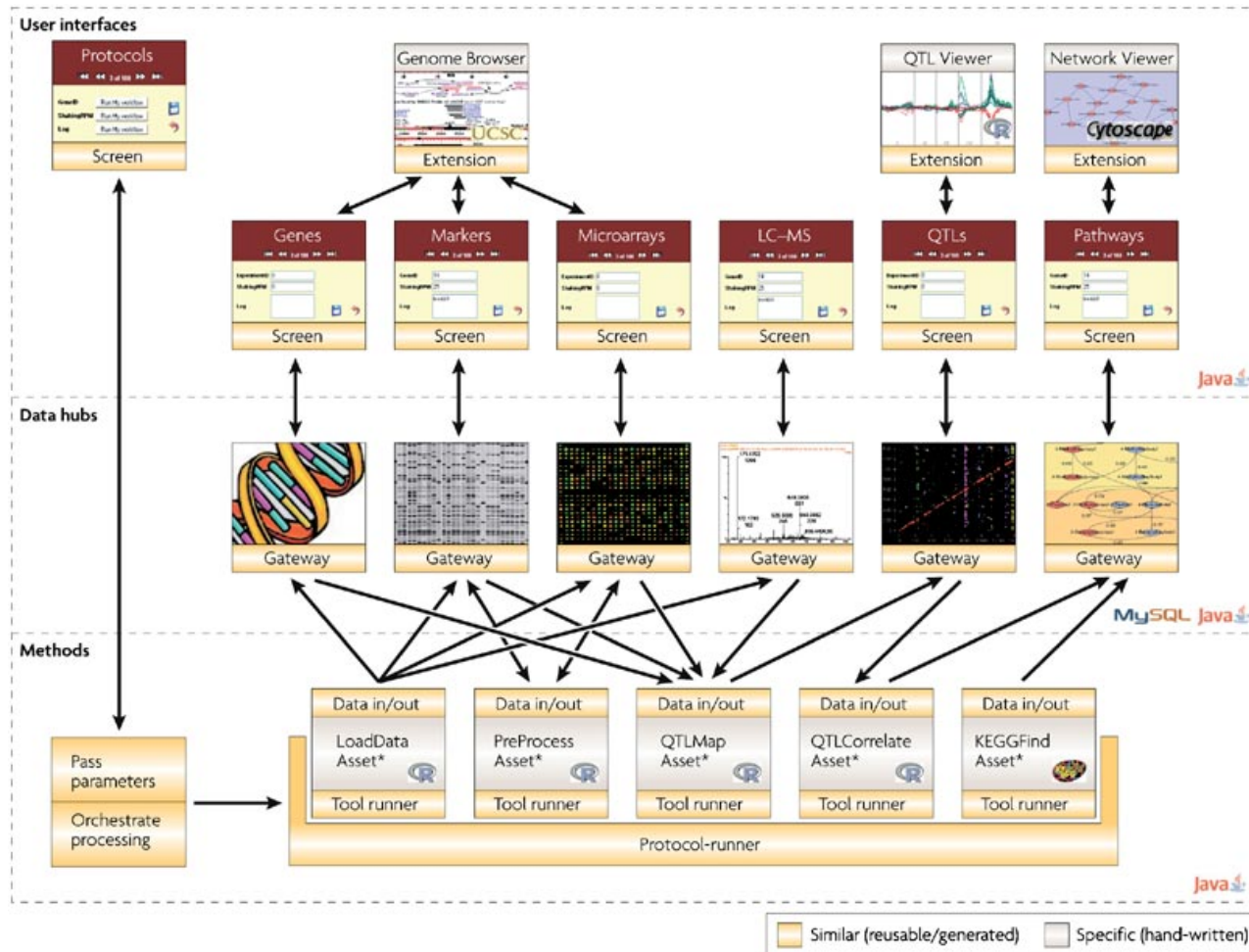
modular philosophy of layers to separate

- back-end: data and compute processing layer
- middle-ware: analysis methodology layer
- front-end: human interaction and data visualization layer

will enable overlapping communities to

- customize local use
- share data, methods & results with other communities
- off-load data handling & compute headaches

software infrastructure



Nature Reviews | Genetics

tools workflow

