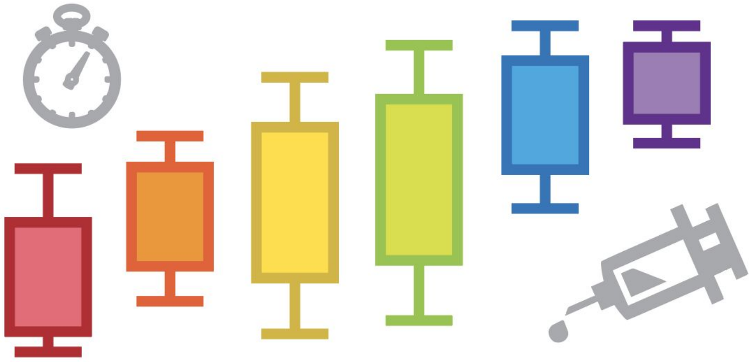
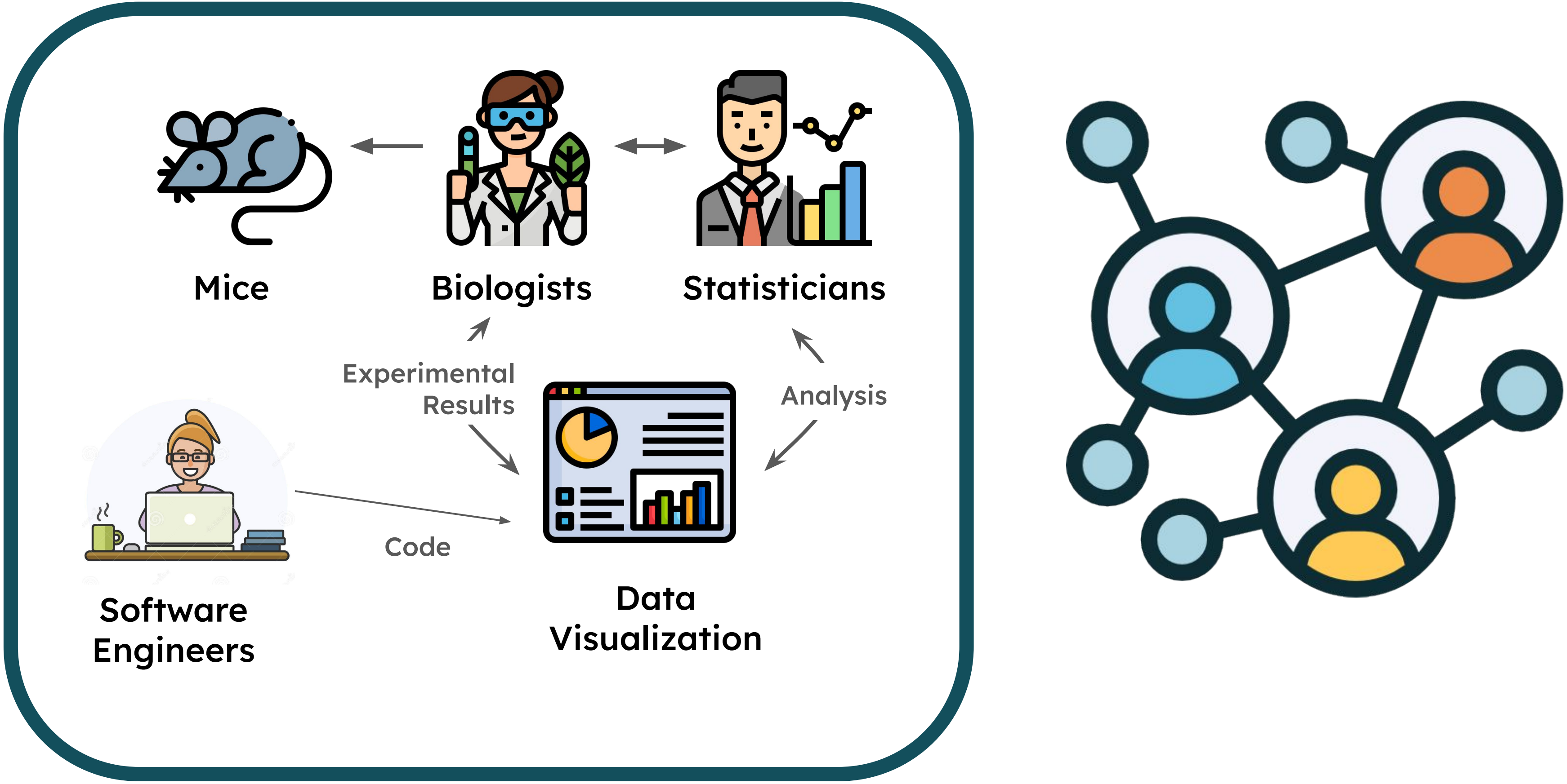
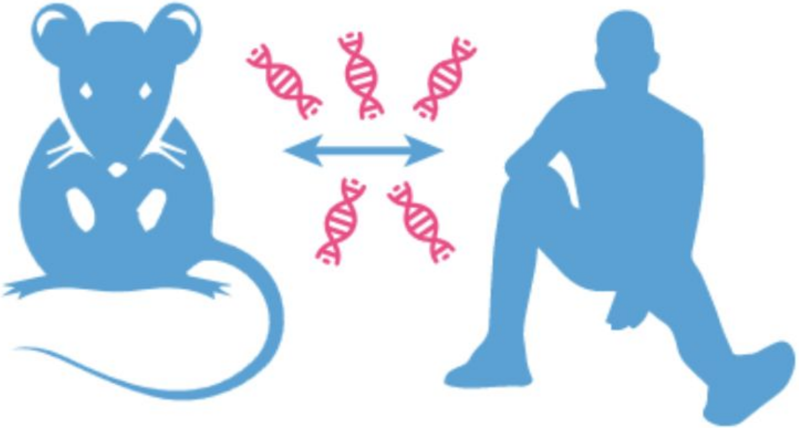


Systemic Challenges of Visualization Software Engineering in Genetics Research

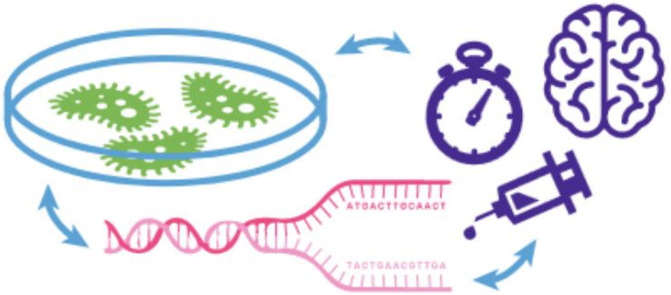
Jane Adams
3rd Year Computer Science
Khoury Data Visualization Lab


Mice
Biologists
Statisticians
Experimental
Results
Analysis
Software
Engineers
Code
Data
Visualization

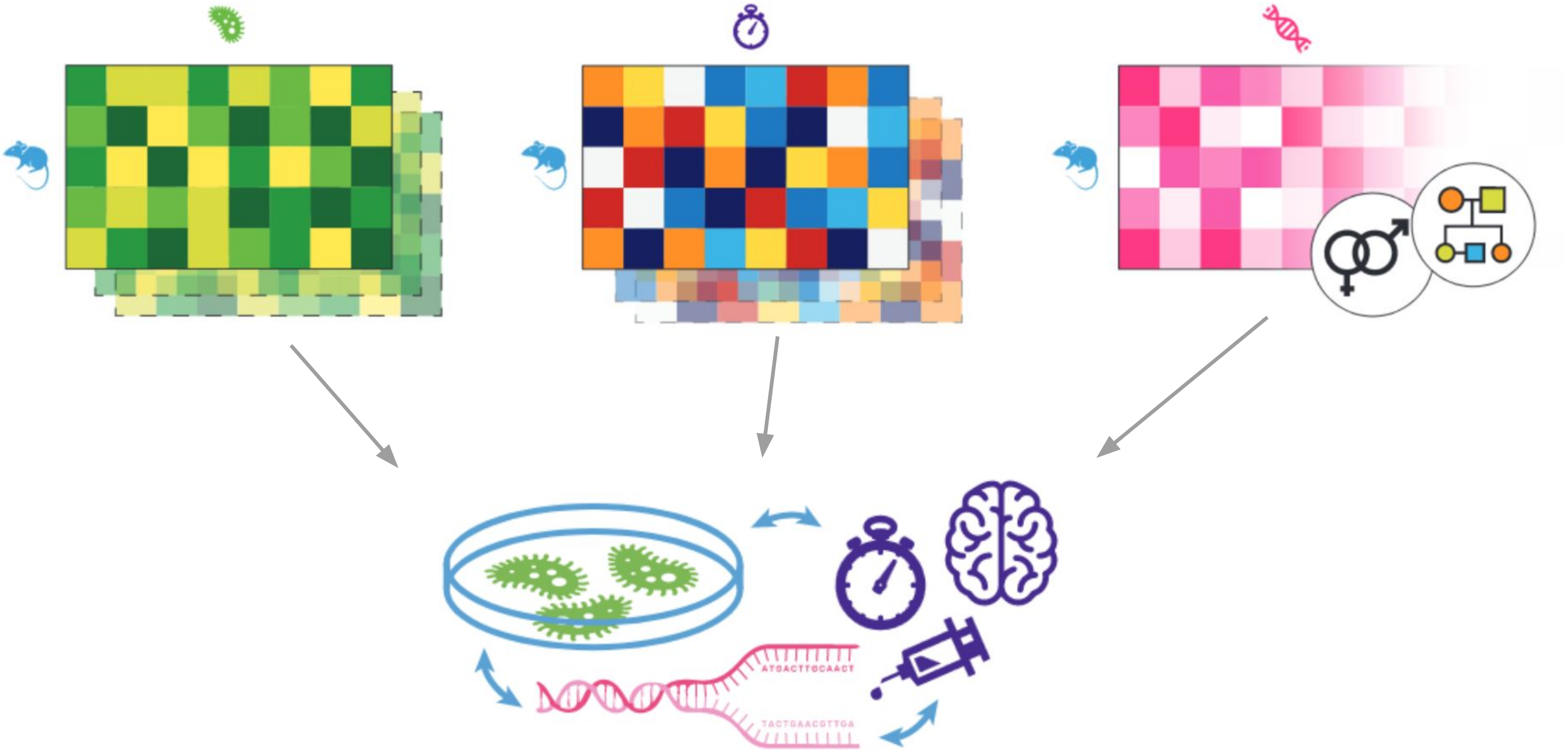


Studies in mice can help us make sense of human disease, due to genetic *orthologies* (overlap). In studying mice, we can formulate and test hypotheses quickly, and have experimental controls not afforded by human subjects research.

We are just beginning to understand the major role that genes and microbes play in determining traits, including behaviors -- in mice and in humans.

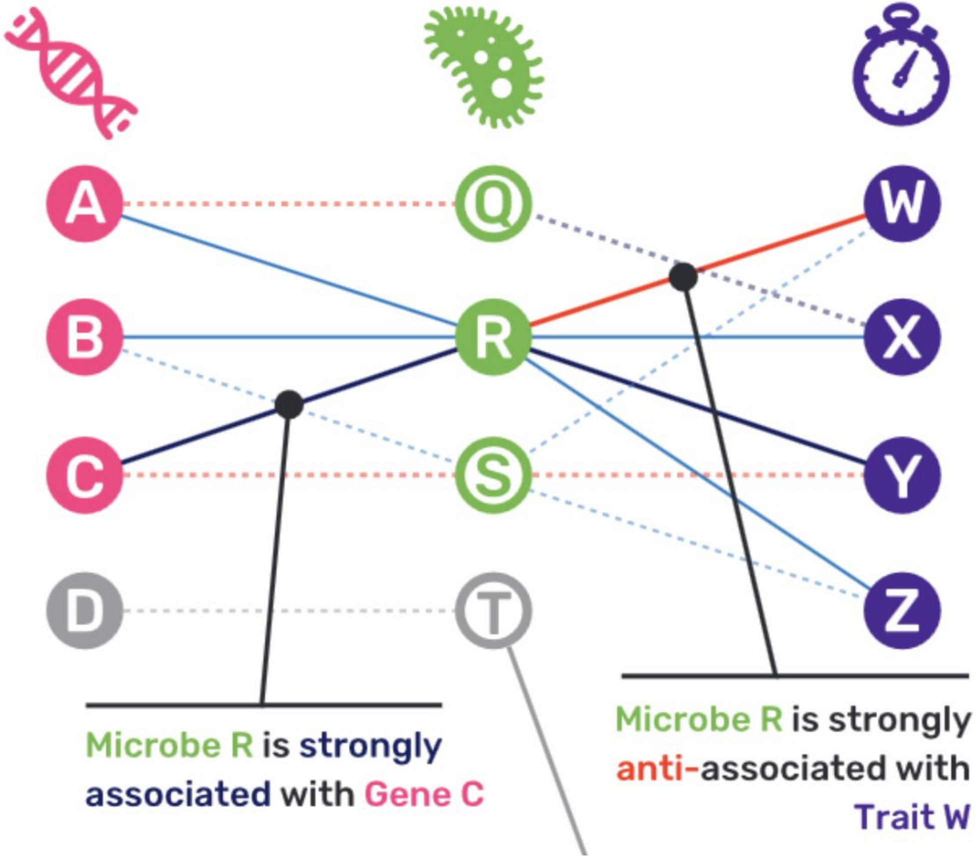


Did you know? The gut provides ~95% of humans' total body serotonin, and 50% of the body's dopamine is stored in the gut. That's why the gut can be known as your "second brain"!

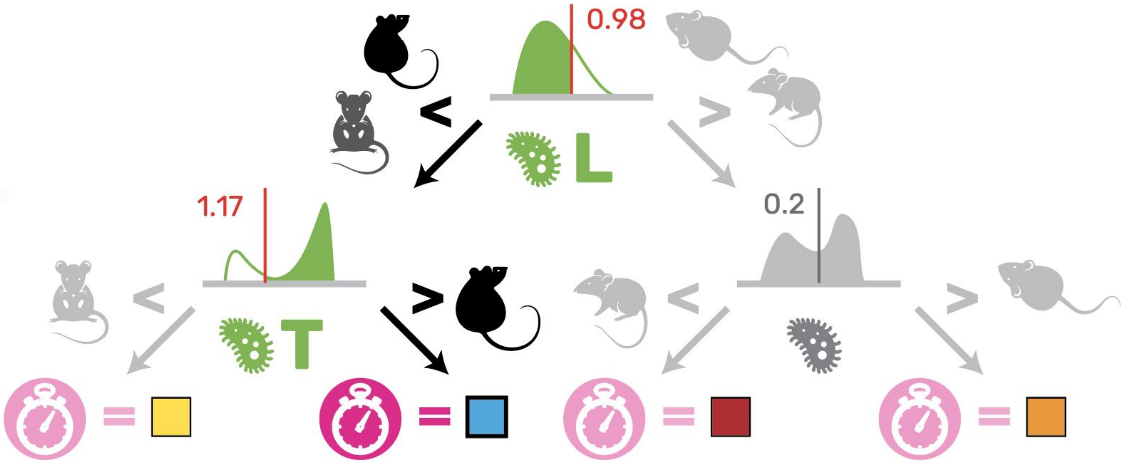


ATGACTTGCAACT
TACTGAACGTTGA

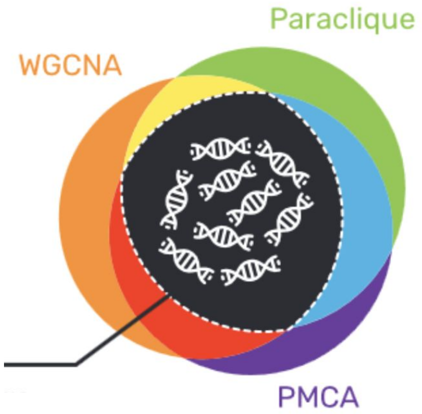
"How are genes and microbes working together to influence addiction-related traits?"

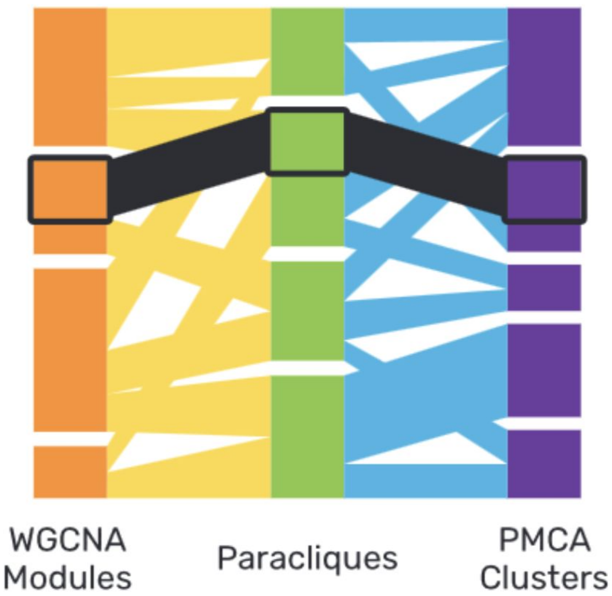


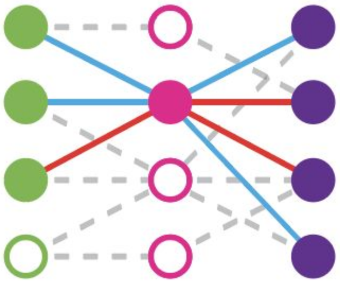
“How do these microbes work together to influence addiction-related traits?”



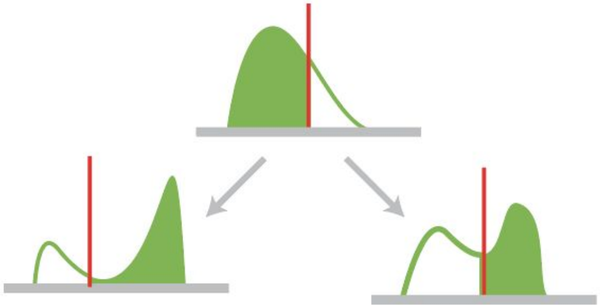
"Do the clustering methods agree on which genes should be in the same set?"



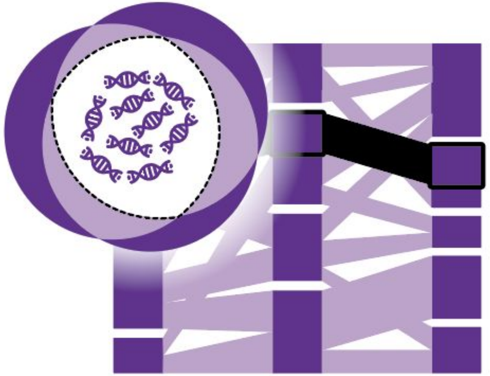




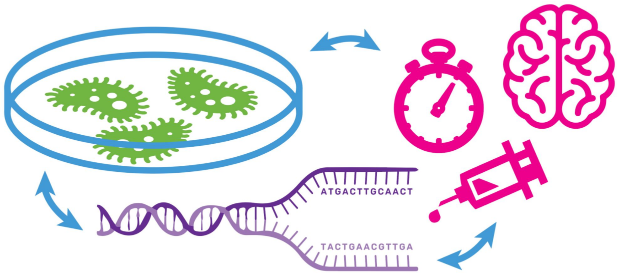
k-partite graph

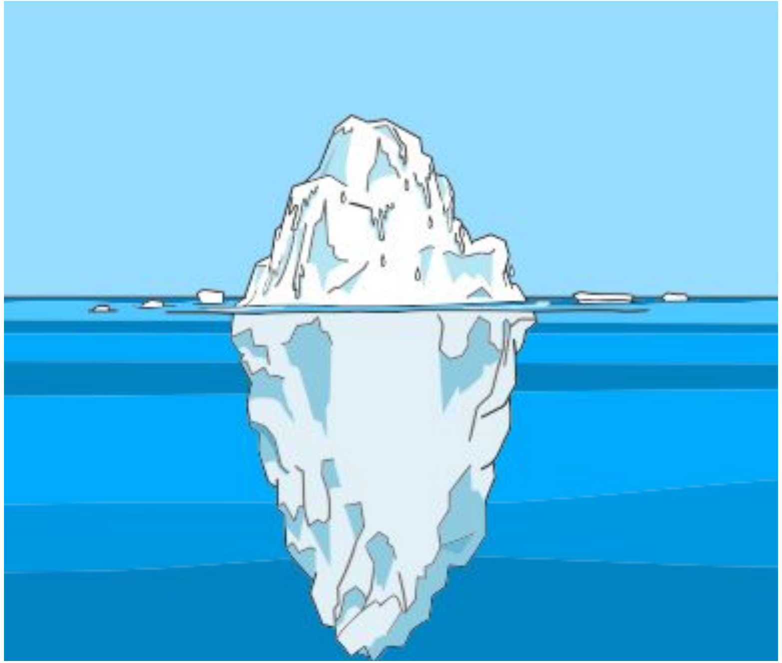


conditional inference tree



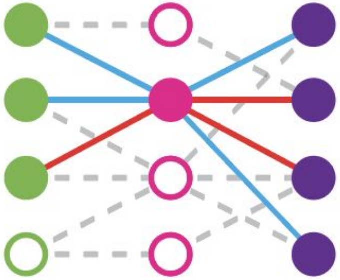
diff. co-expression concordance



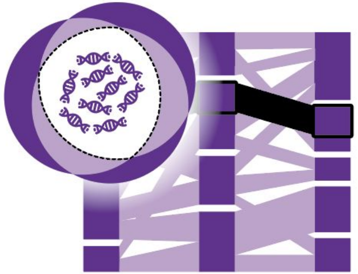


The visualization is just the tip of the iceberg…

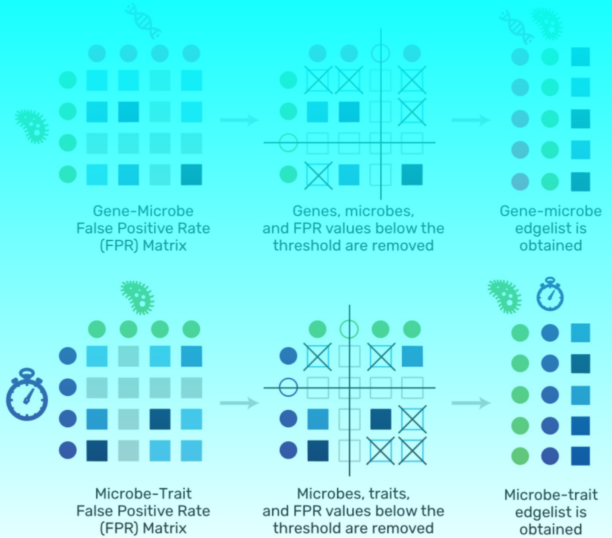
There’s a lot of code underneath that happens to transform the data

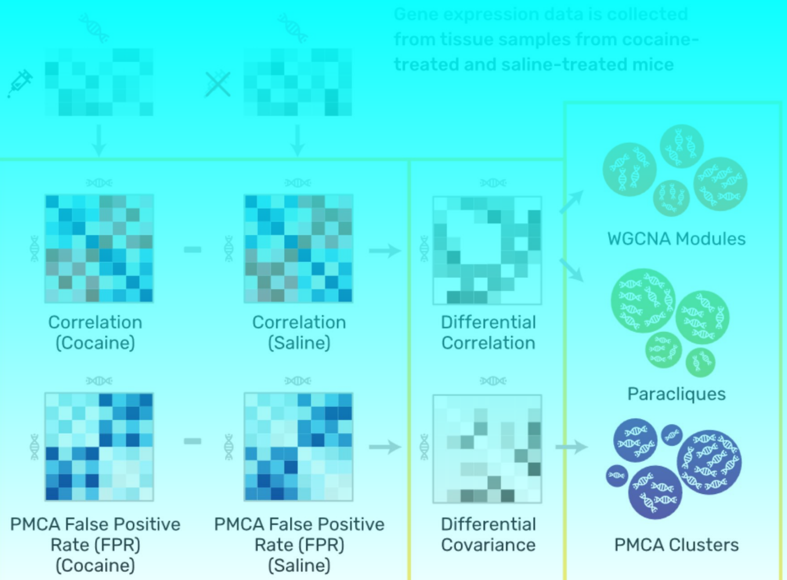


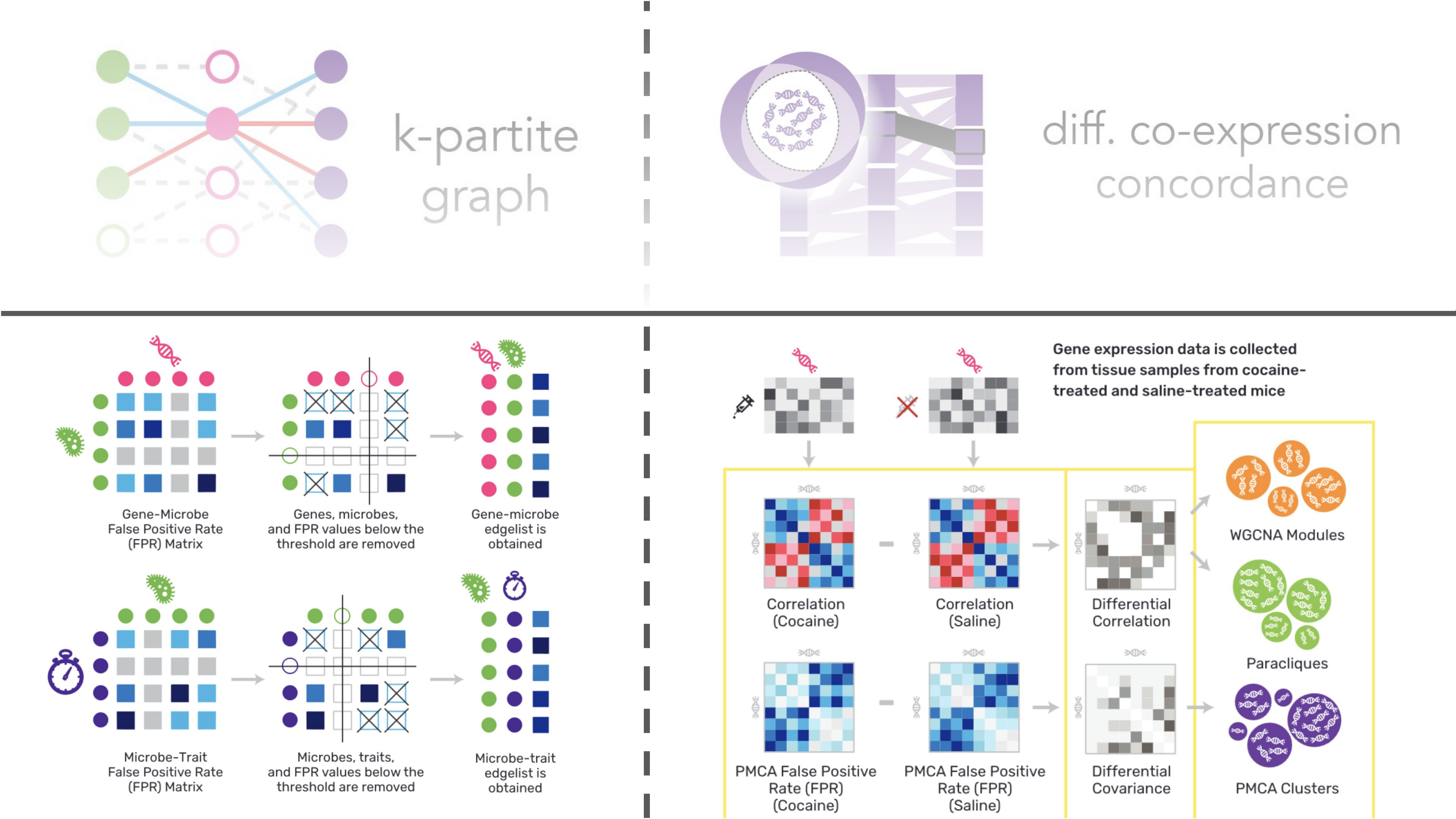
k-partite graph



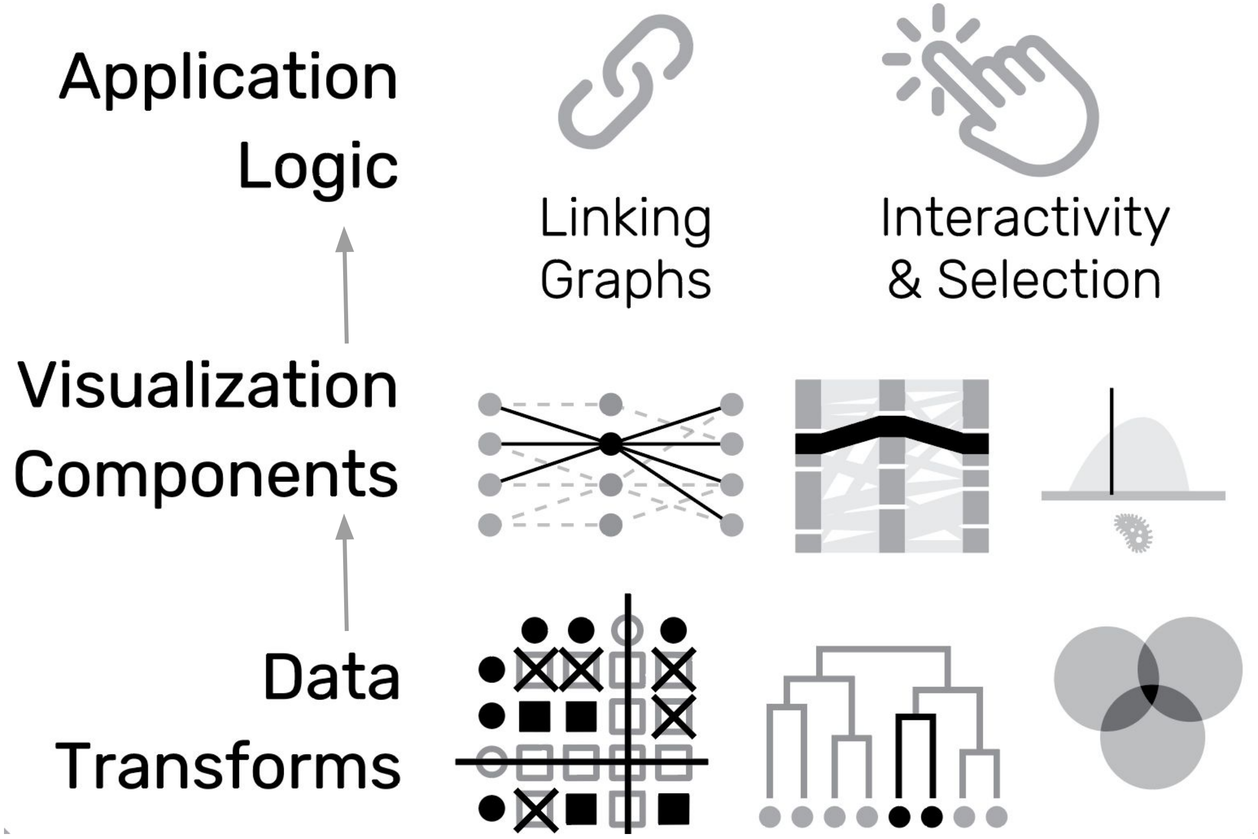
diff. co-expression concordance

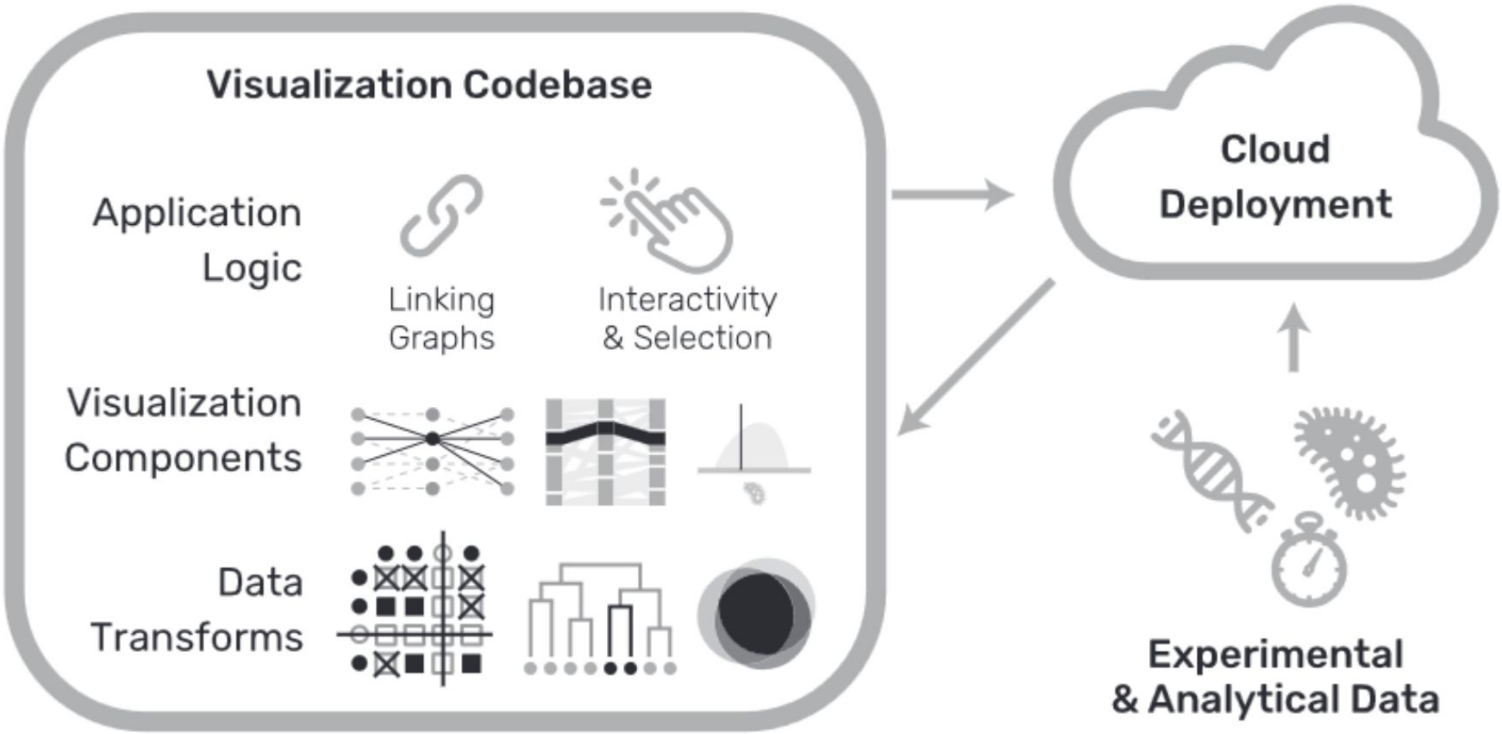






k-partite graph
diff. co-expression concordance
Gene-Microbe False Positive Rate (FPR) Matrix
Genes, microbes, and FPR values below the threshold are removed
Gene-microbe edgelist is obtained
Microbe-Trait False Positive Rate (FPR) Matrix
Microbes, traits, and FPR values below the threshold are removed
Microbe-trait edgelist is obtained
Gene expression data is collected from tissue samples from cocaine-treated and saline-treated mice
Correlation (Cocaine)
Correlation (Saline)
Differential Correlation
PMCA False Positive Rate (FPR) (Cocaine)
PMCA False Positive Rate (FPR) (Saline)
Differential Covariance
WGCNA Modules
Paracliques
PMCA Clusters


Application
Logic
Linking
Graphs
Interactivity
& Selection
Visualization
Components
Data
Transforms


Visualization Codebase
Application Logic
Linking Graphs
Interactivity & Selection
Visualization Components
Data Transforms
Cloud Deployment
Experimental & Analytical Data

diff. co-expression concordance

Parameters variably have to be supported by:

1) CLI args
2) .env
3) buttons

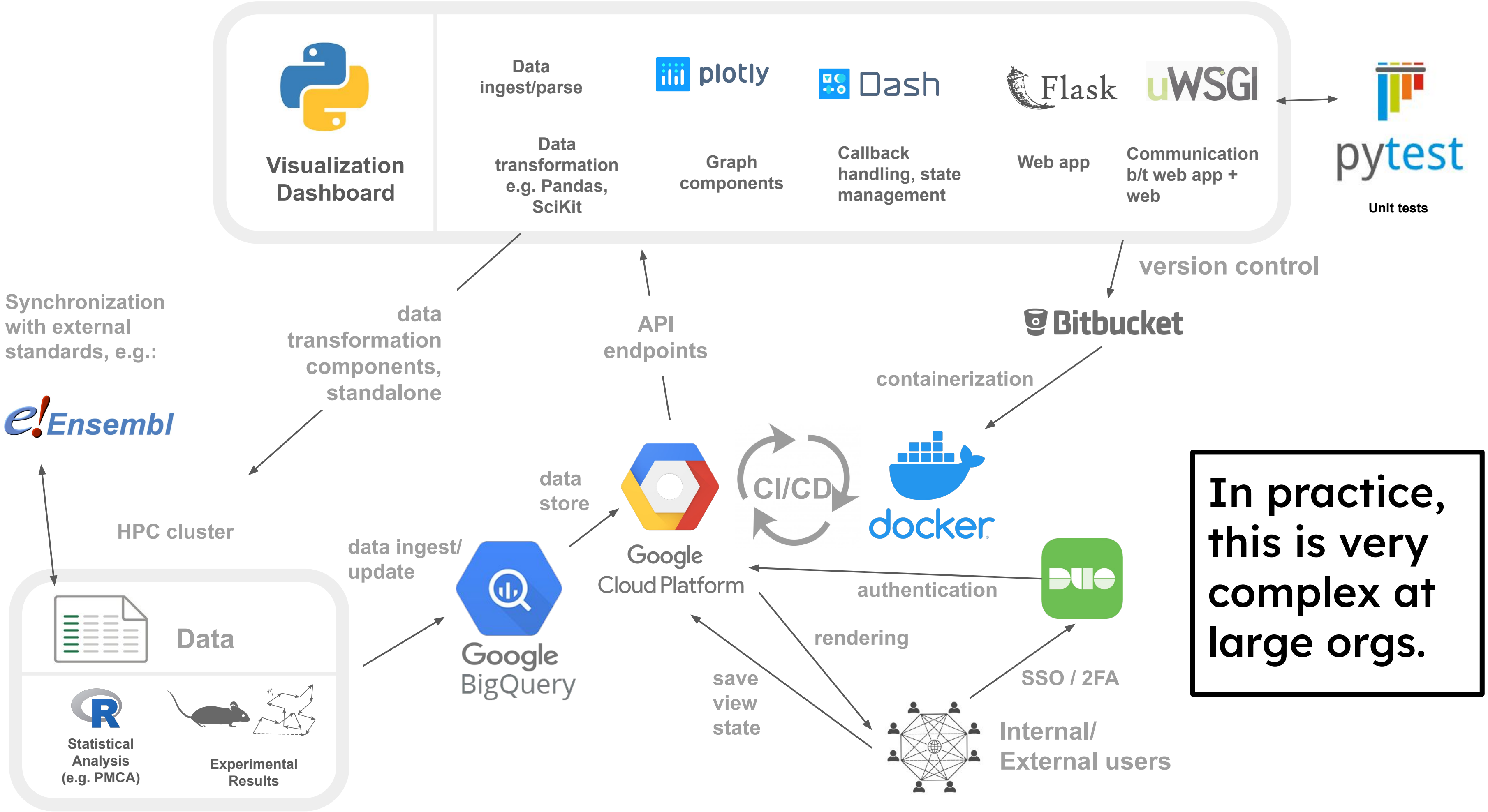
Setup parameters *(where does the data live?)*

ANNOTATION_NAME	my_genome_features	--annotation-name	str	None
GTF_PATH_OR_URL	data/Mus_musculus.GRCm38.102.gtf	--gtf-path	str	None
PARACLIQUE_PATH	data/paraclique.txt	--paraclique-path	str	None
PMCA_PATH	data/pmca.txt	--pmca-path	str	None
WGCNA_PATH	data/wgcna.txt	--wgcna-path	str	None
DEBUG	TRUE	--debug	bool	True
PORT	8888	--port	int	8888

Interaction parameters (what view is being displayed *this moment?*)

Parameters:

Parameter	Default	Description
`df`	N/A	The input DataFrame containing gene information.
`methods`	N/A	The names of the methods (column names) in the DataFrame.
`all_bool`	`True`	If set to `True`, it applies the 'threshold_all' filter.
`module_N`	`0`	Specifies the minimum number of genes required in each module.
`path_N`	`0`	Indicates the minimum number of genes required in each gene set union.
`debug`	`True`	If set to `True`, the function will print debug messages to help in troubleshooting.


Visualization Dashboard
Data ingest/parse
plotly
Dash
Flask
uWSGI
Data transformation e.g. Pandas, SciKit
Graph components
Callback handling, state management
Web app
Communication b/t web app + web
pytest
Unit tests
version control
Bitbucket
Synchronization with external standards, e.g.:
Ensembl
data transformation components, standalone
API endpoints
containerization
HPC cluster
data store
data ingest/ update
Google Cloud Platform
CI/CD
docker
In practice, this is very complex at large orgs.
authentication
Data
Google BigQuery
rendering
SSO / 2FA
save view state
Statistical Analysis (e.g. PMCA)
Experimental Results
Internal/ External users

The data visualization community is not prepared for this volume and complexity of data

DevOps for DataVis: A Survey and Provocation for Teaching Deployment of Data Visualizations

Jane L. Adams

Tooling Keyword Groups	Web	Database	Versioning	Dashboards	Notebooks	Server	Dependencies	Publishing	Coding	Pipeline	Testing	Scale	Community	Governance
Python	11	8	6	4	9	8	8	5	4	7	3	4	1	0
Web	13	6	8	4	7	8	6	5	5	3	3	2	0	0
Other	8	4	5	2	3	3	3	3	3	1	2	3	1	1
Tableau	7	6	3	9	1	2	2	3	1	3	2	2	1	0
R	5	4	7	4	4	1	1	2	4	3	2	1	2	0
Sheets	4	3	2	5	0	0	0	2	2	1	1	0	1	1

DevOps Keyword Groups

Fig. 1: Co-occurrence of tooling keyword groups and DevOps keyword groups within each syllabus from a survey of 65 data visualization college courses. Values represent the total number of syllabi that contained at least one mention from each keyword group. The most common DevOps keyword group, 'web', was mentioned in only 35.4% of syllabi.

Abstract—We present a provocation towards teaching development operations ("DevOps") and other infrastructure concepts in the course of collegiate data visualization instruction. We survey 65 syllabi from semester-long, college-level data visualization courses, with an eye toward languages and platforms used, as well as mentions of deployment related terms. Results convey significant variability in language and tooling used in curricula. We identify a distinct lack of discussions around 'DevOps for DataVis' scaffolding concepts such as version control, package management, server infrastructure, high-performance computing, and machine learning data pipelines. We acknowledge the challenges of adding supplemental information to already dense curricula, and the expectation that prior or concurrent classes should provide this computer science background. We propose a group community effort to create one free 'course' or 'wiki' as a living reference on the ways these broader DevOps concepts relate directly to data visualization specifically. A free copy of this paper and all supplemental materials are available at `https://osf.io/bxaqz/`.

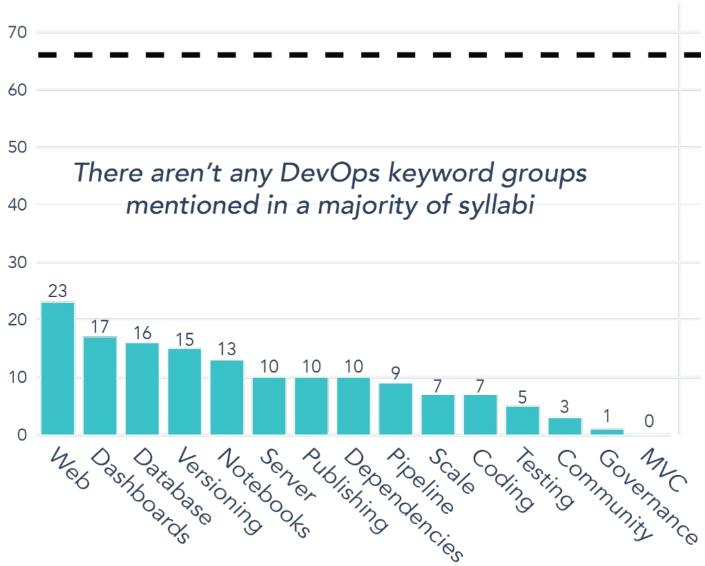
Index Terms—Computing, infrastructure, deployment, software engineering, education.

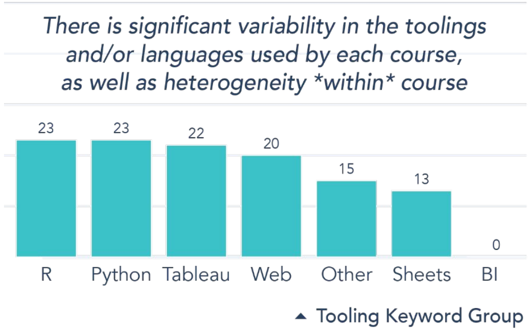
1 Introduction

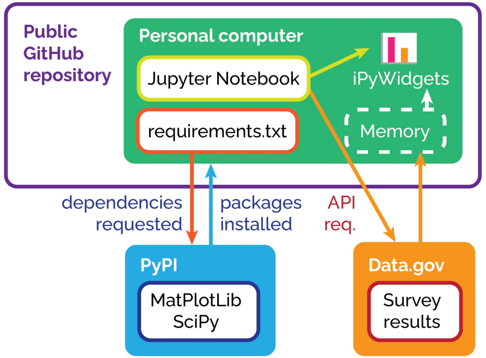
There exists significant heterogeneity in the content of collegiate data visualization curricula, both with regard to content and tooling. Some of these differences can be explained by the programs in which these courses are housed, which may range from social science to machine learning— the inherent symptoms of a highly interdisciplinary field of study. Likewise, there is tremendous variability in the existing familiarity students have with the technologies and languages used in these data visualization courses. The result of this diversity can be productive, as courses can theoretically cater more narrowly to the direct needs of students in a particular program; but they can also create problems. Students may complete a course feeling confident in their ability to code interactive visualizations, only to face confusing and complex battles in deploying these visualizations for use in a portfolio or in the context of building a dashboard for an employer. In these latter cases, it may have been beneficial for the student to have encountered educational scaffolding related to deployment and infrastructure – development operations, or "DevOps" – during their coursework.

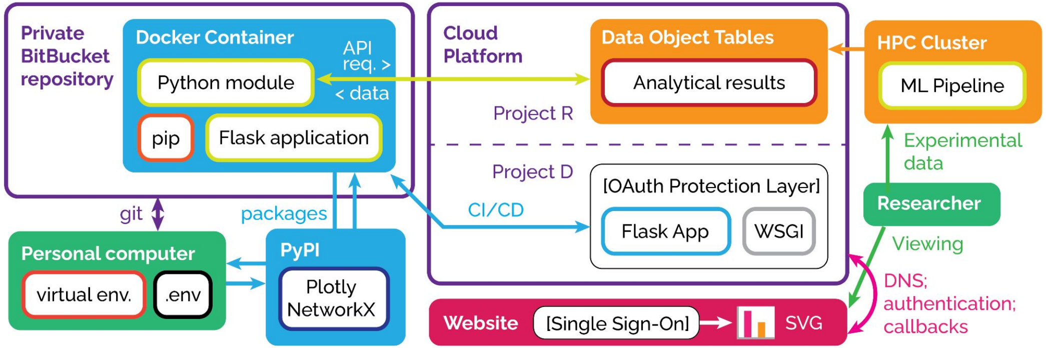
This is a symptom also of the 'gap' between academic research and industry practice, as described by Velt et al. [20], investigated by Parsons through interviews with practitioners [17], and discussed in the VisGap workshops of 2021-2023 [5, 7, 11]. As the proportion of PhD graduates heading to industry surpassed academia for the first time in 2020, and continues to rise, educational aims necessarily should consider the needs of industry positions [13]. Concurrently, as visualization researchers increasingly encourage one another to consider the long term reusability of research prototypes, the value of lessons in these concepts extends beyond the classroom [11]. A search of "data

• *Jane Adams is with Northeastern University. E-mail: adams.jan@northeastern.edu*
• *Conflict of Interest (COI) Disclosure: Jane Adams is on the steering committee of alt.VIS, and was an organizer in 2021 and 2022.*

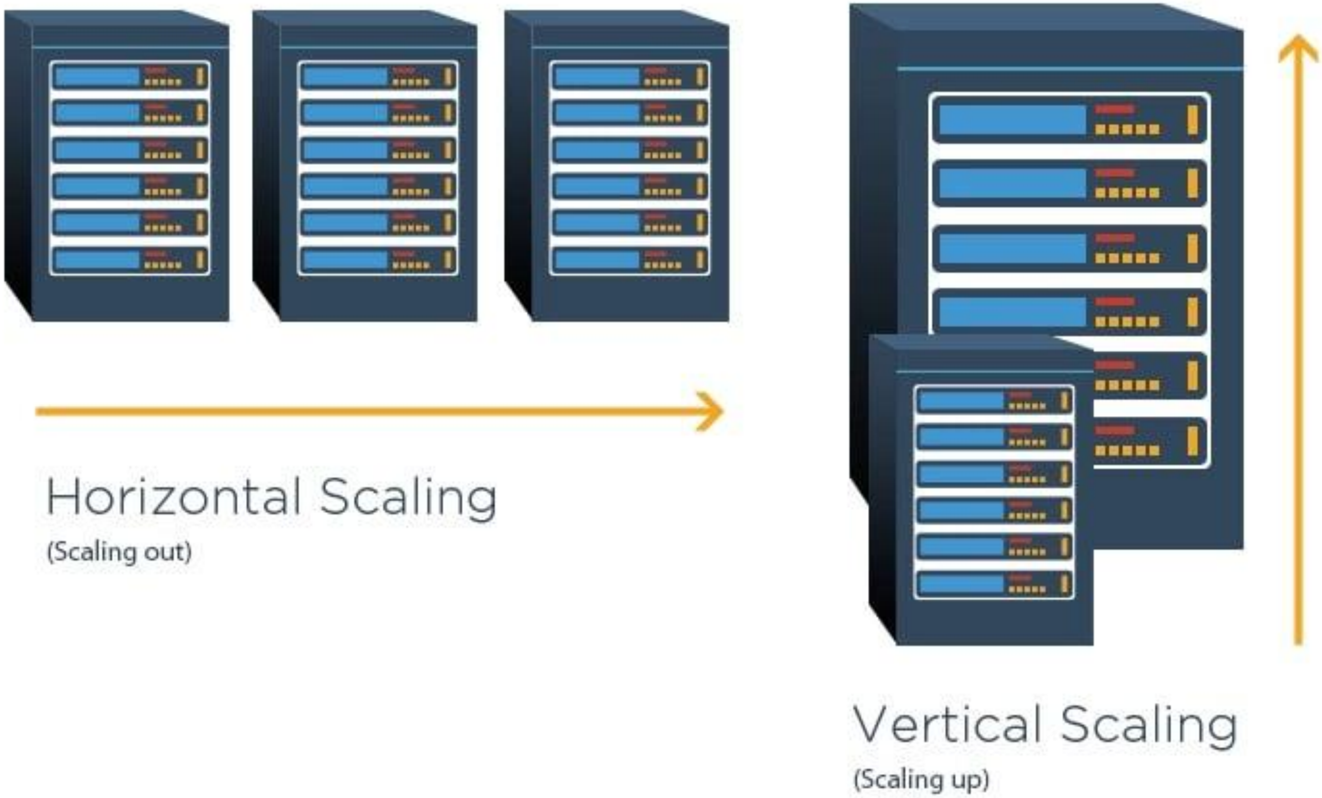




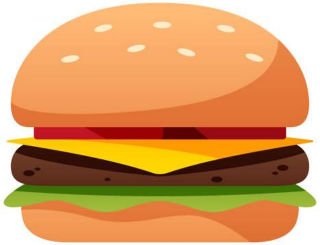

Public GitHub repository
Personal computer
Jupyter Notebook
iPyWidgets
requirements.txt
Memory
dependencies requested
packages installed
API req.
PyPI
MatPlotLib
SciPy
Data.gov
Survey results

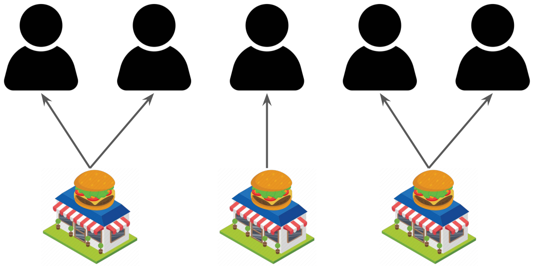

Private BitBucket repository
Docker Container
API req. >
Python module
< data
pip
Flask application
Cloud Platform
Data Object Tables
Analytical results
Project R
Project D
[OAuth Protection Layer]
CI/CD
Flask App
WSGI
HPC Cluster
ML Pipeline
Experimental data
Researcher
Viewing
git
packages
Personal computer
virtual env.
.env
PyPI
Plotly
NetworkX
Website
[Single Sign-On]
SVG
DNS; authentication; callbacks

Research software will increasingly run into the problem that startup infra has known for years:



Research software will increasingly run into the problem that startup infra has known for years:





It's easy to grow wide…

(horizontal scaling)

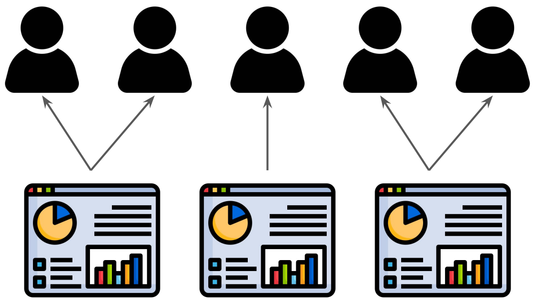
Lots of research code is organized like this:

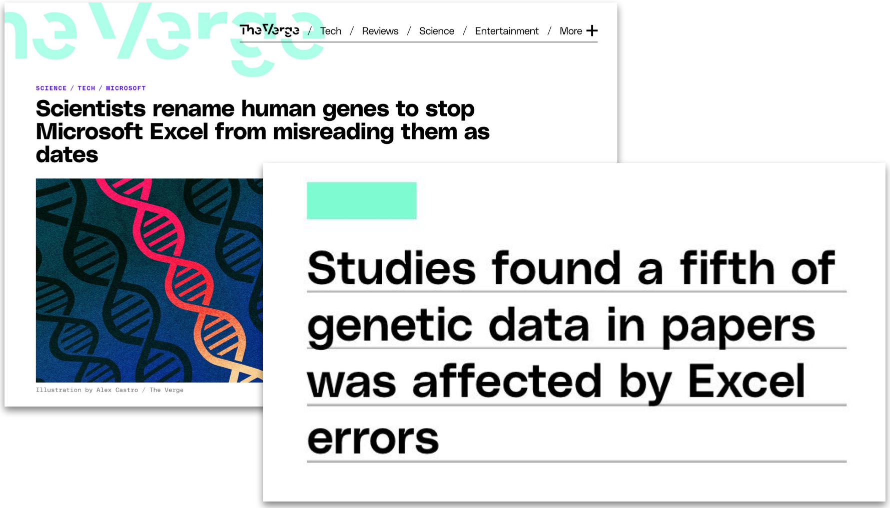
A small team (lab) or IC (single author) creates a codebase…

…if the [data, functions] appear in multiple apps,

the [data, functions] exist in multiple places

Research software will increasingly run into the problem that startup infra has known for years:




The Verge / Tech / Reviews / Science / Entertainment / More +

SCIENCE / TECH / MICROSOFT

Scientists rename human genes to stop Microsoft Excel from misreading them as dates

Illustration by Alex Castro / The Verge

Studies found a fifth of genetic data in papers was affected by Excel errors

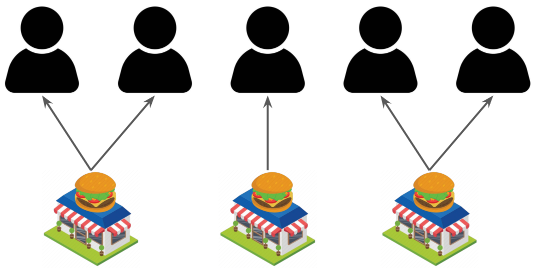
Lots of research code is organized like this:

A small team (lab) or IC (single author) creates a codebase…

…if the [data, functions] appear in multiple apps,

the [data, functions] exist in multiple places

Research software will increasingly run into the problem that startup infra has known for years:

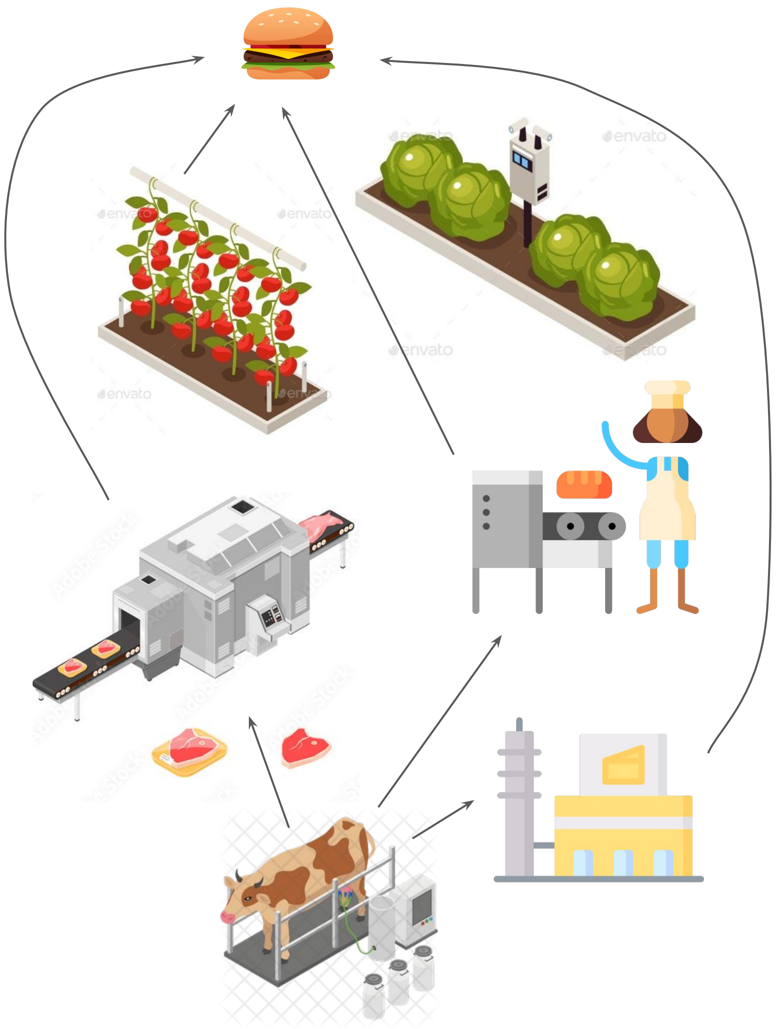


It's easy to grow wide…

(horizontal scaling)

…it's hard to grow tall

(vertical scaling)

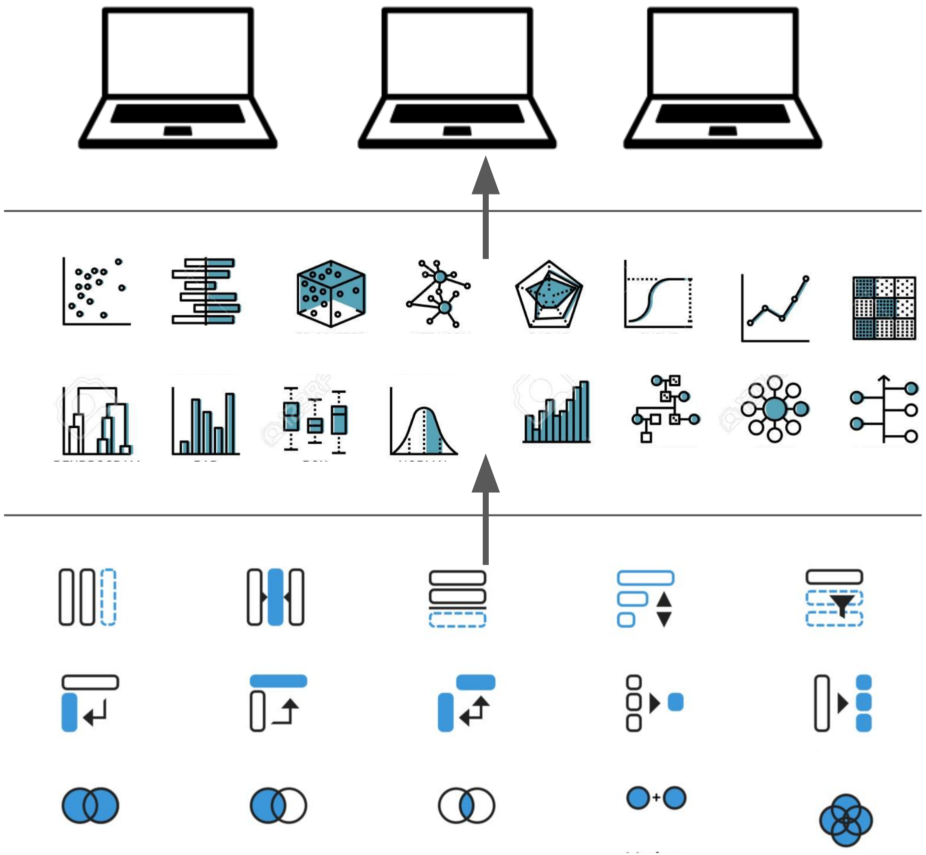


Compartmentalizing requires interdependence
(documentation, communication, etc etc etc…)

Dashboards

Visualization components

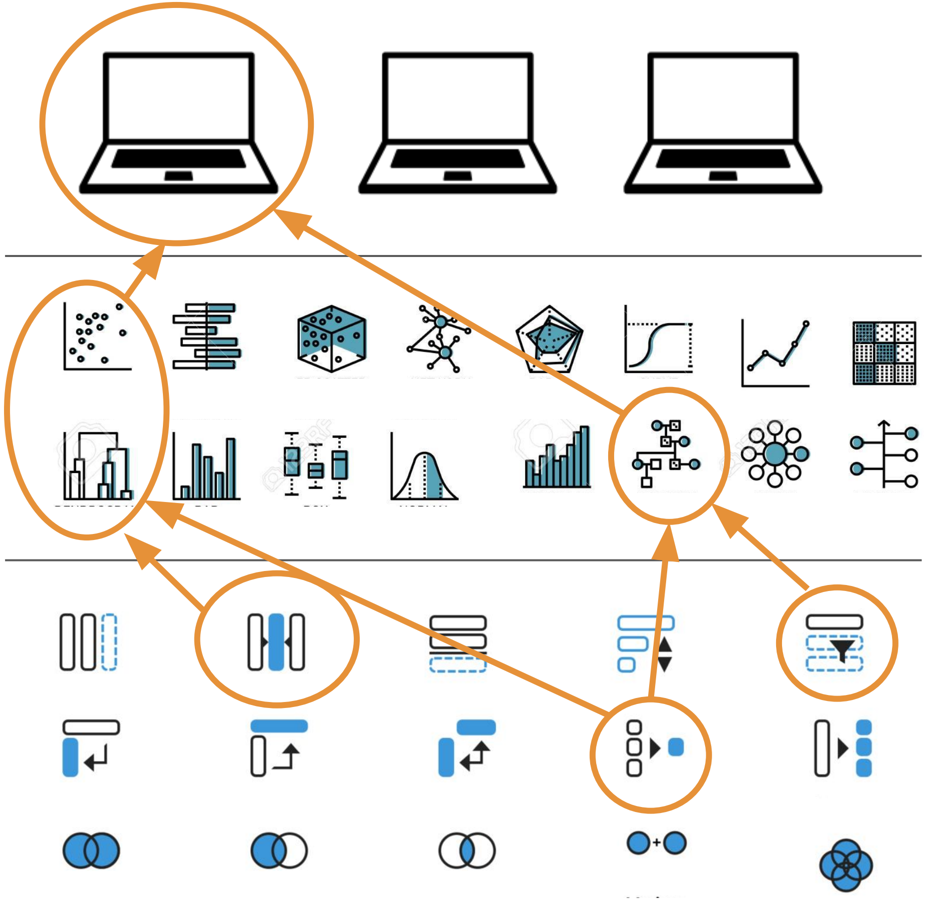
Data transformations



Dashboards

Visualization components

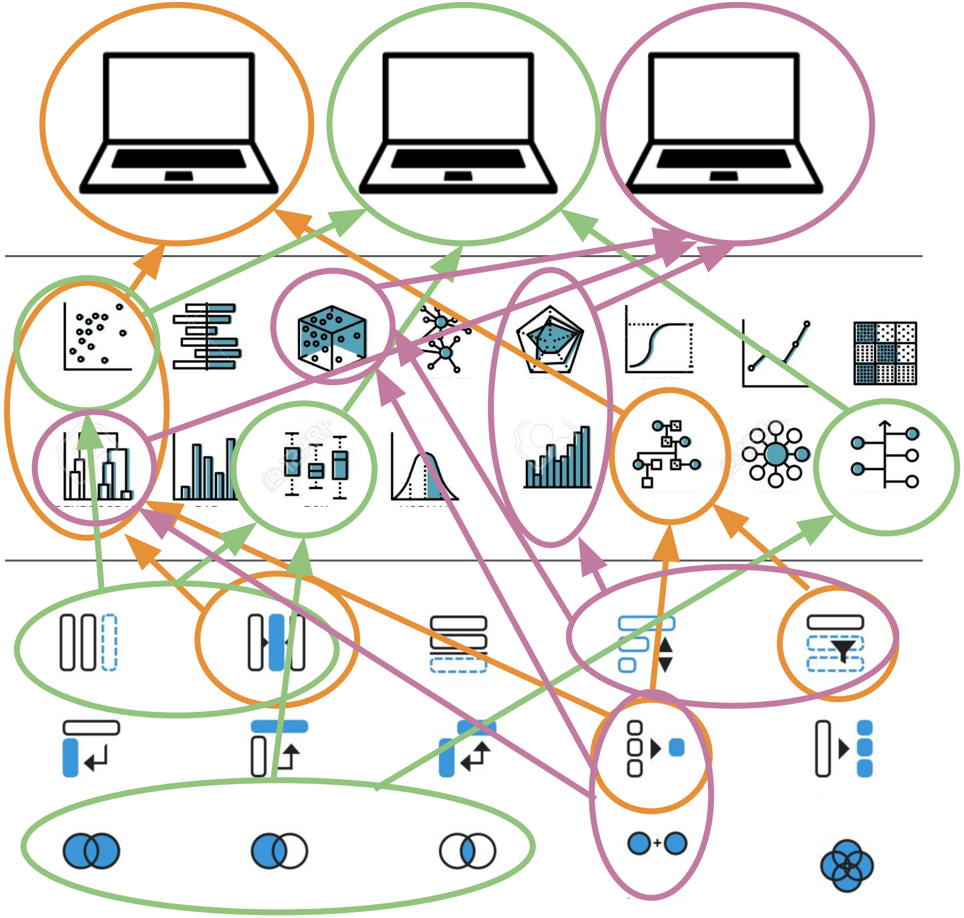
Data transformations



Dashboards

Visualization components

Data transformations

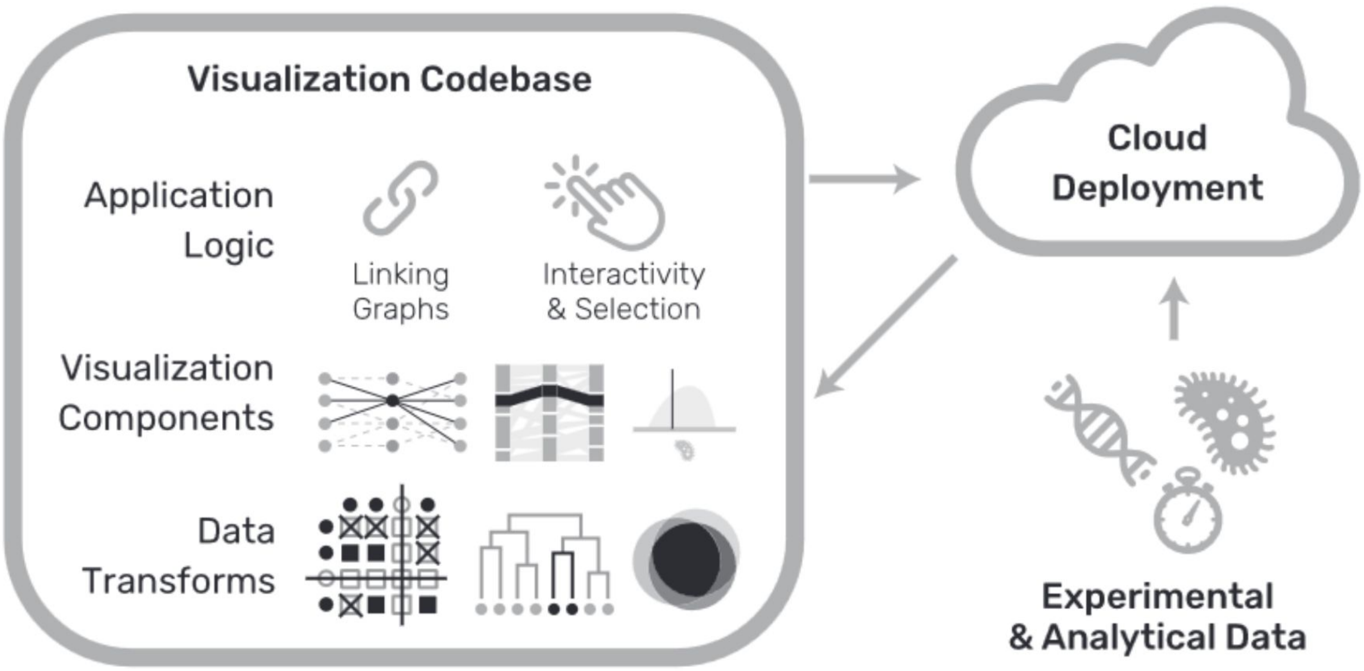


Users expressed data management concerns due to complexities of institutional embedding, volume

Reduced Redundancy	Data objects should only live in one location, with version control
Governance	Storage objects and visualization projects need dynamic permissions scoping that align with research release cycle
Cross-institutional syncing	If visualizations rely on external authorities e.g. for nomenclature and ontologies, they should update in sync with that authority. For example, Ensembl gene IDs change with new research
Egress	Any transformation that can be made using the UI should be exportable and workflow recorded. Imagine a 'graphical API'
Multimodality	Web deployment but also paper publication, scientific notebooks (Python, R)
Longevity	Long-term support via reduced technical debt, unit tests, and platform support

There are systemic challenges to meeting these objectives

Funding	There is limited funding either for person-power or compute resources to set up workflows
Time	‘Publish or perish’ and grant obligations mean limited time for processes like unit testing
Siloed ownership	When teams are organized by biological research question, there is redundancy due to reduced communication
Intellectual property	Open sourcing code can be challenging when data has already been open sourced and analysis is the primary novel contribution
Comfort	Don’t tell R users they have to learn Python… …especially not statisticians
Mental model	Modularity of code elements is incongruous with organizing projects into distinct compartments



Thanks! Jane Adams (WVH 306)