FINAL PRESENTATION
MOG Project
EVE-MOG-5

Adam, Austin, Fahmi, John, Kamsi

MetaOmGraph
a workbench for interactive exploratory data analysis of large expression datasets
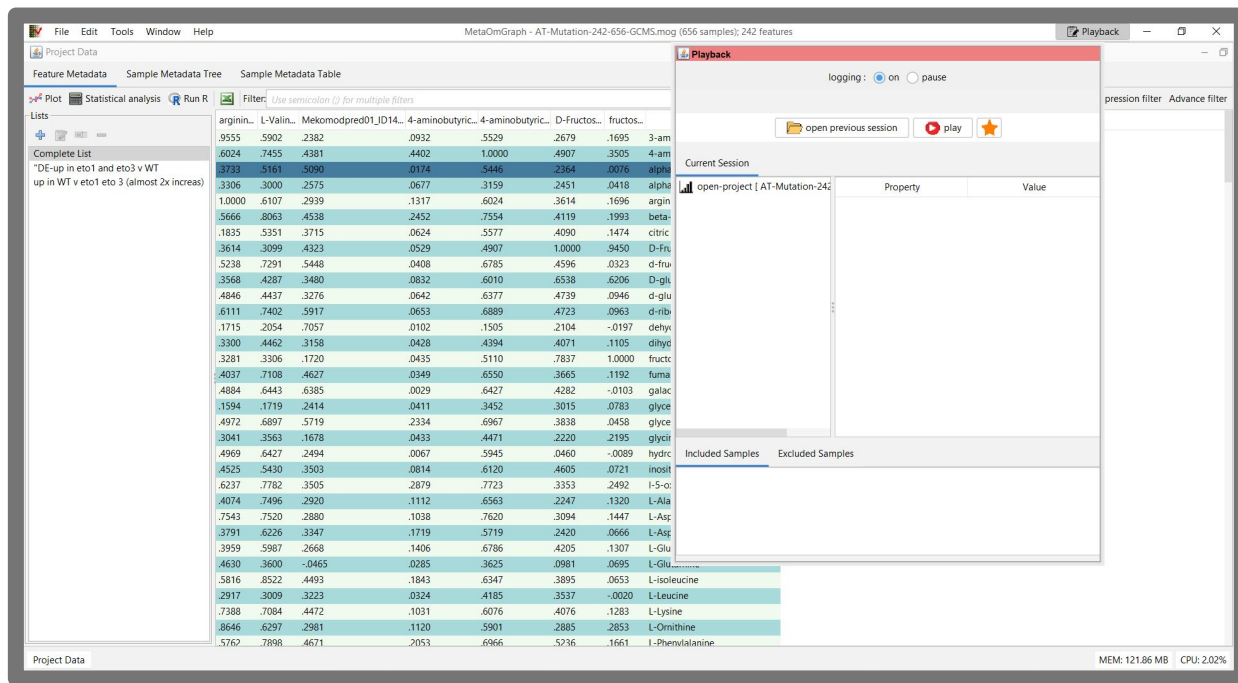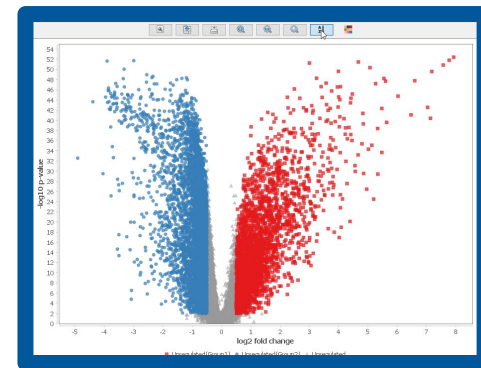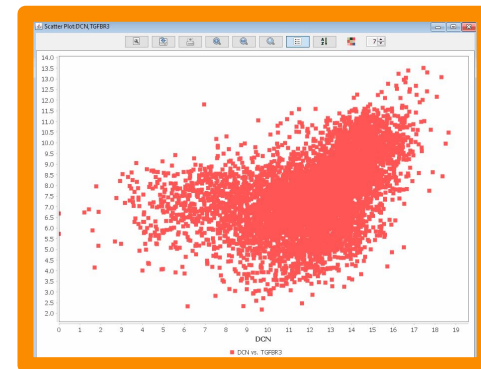

Figure 1

Overview

Updating and Improving MetaOmGraph (MOG)

Main goals

- Java Upgrade

- Limma Analysis

- Filter Improvements

- Logging and Playback

- Quality of Life Improvements

- Performance Improvements

- Increasing Test Coverage

Technologies Used

Development

- Java 11
- Maven Framework
- Stan4j; Log4j
- Javascript
- Swing
- Renjin (R in JVM)

Project Management

- Notion
- When2Meet
- GitLab Issues
- Discord
- Webex
- Google Sheets/Docs

Requirements tracking

#	Task	WorkSize	Client's Rank	Status	Assignments	Comments
3	Limma Analysis	8		PARTIALLY COMPLETE	Fahmi, Austin	
11	Enhancements to HeatMap	3		PARTIALLY COMPLETE	Adam	
9	Identifying and Fixing Perf bottlenecks	3		PARTIALLY COMPLETE	Adam	
18	Launcher (set diff memory profiles)	4		5 Done	Adam	
1	MOG latest version of Java	3		5 Done	Austin, Fahmi	
2	Filter Genes	7		5 Done	Austin, Kamsi	
6	Complex AND and OR queries	5		5 Done	Austin	
7	Extend Logging.Playback	5		5 Done	Austin	
10	Add Violin Plot to MOG	5		5 Done	Kamsi	
15	Display MOG cpu meory usage in taskbar	1		5 Done	Kamsi, Austin	
16	Add an option to add mean/median of values (Gene filter)	3		5 Done		
19	Setting up CI/CD	1		5 Done		
20	Docker build process	3		5 Done		
21	Package/bundle app frontend into standard OS executable format	3		5 Done		
13	Add help using animations, gifs	1		INCOMP		
12	Extend MOG as general plotting	0		INCOMP		
4	Add unit testing	8		INCOMP		
8	Enable hypothesis testing in boxplot charts	0		INCOMP		
22	Web Framework	8		INCOMP		
14	Auto fetch standard gene metadata for organisms	0		INCOMP		
17	Add set/unset plots to MOG	0		INCOMP		

Figure 2

Status dropdown options:
- 5 Done
- 0 UNASSIGNED
- 1 TODO
- 2 Next Up
- 3 Just Started
- 4 In Progress
- 0 PAUSED
- 5 Done

Assignments (right panel):
Adam
Adam
Austin, Fahmi
Austin, Kamsi
Austin
Austin
Kamsi
Kamsi, Austin
Austin
John
John

Development Practices

- Agile workflow

- Assigning time estimates to tasks

- Tracking known issues from discovery to resolution

- Pairing and Mobbing Sessions

- Cross-Platform Testing

Development Challenges

Problems

1. Difficult to reach client for the first few weeks

2. Confusing project goals

   a. Web or desktop?

   b. Unclear specifications

Solutions

★ Scheduled weekly meetings early in the semester

★ Meet frequently to discuss thoughts with client, professor, and team

Development Challenges

Other Slowdowns

1. Large existing codebase

2. Technical debt

3. Existing technology

How we handled them

★ Stan4j/Code Coverage Tools

★ Learning how to use their technology

Development Challenges

Bugs

- Started appearing as we used MOG more
- Found in our code, and theirs
- Troubleshooting was sometimes difficult

How we handled them

★ Working with our client who knew the code better

★ Debugging in IntelliJ

Current Status

Progress

Advanced Filter

- ❖ Searching data using a complex query comprised of AND and OR operations in CNF form

- ❖ Dynamically display query
- ❖ Refactor code to be much more modular

Gene Expression Filter

- ❖ Searching metadata for minimum expression value

Mean Expression Filter

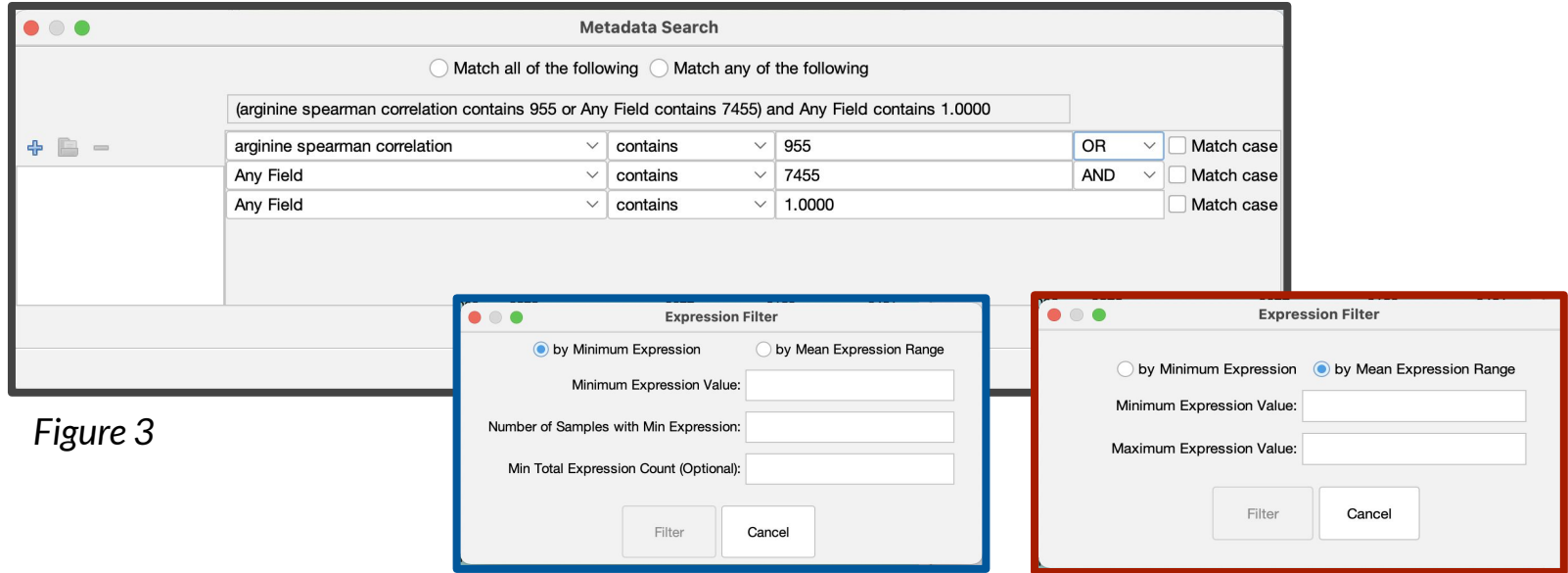- ❖ Searching metadata for a mean value within a range

Filter Changes and Additions

Figure 3

Progress

Extending MOG Logging

❖ MOG logs every action a user takes

❖ As we added features, we also needed to log those

❖ Certain pre-existing features were not being logged at all

Extending MOG Playback

❖ Using MOGs logging users can recreate a graph using the original data

❖ Mostly expanding to cover different kinds of charts

Logging and Playback

Figure 4

Progress

Java Upgrade

❖ Client wanted to move from Java 8 to newer versions

❖ Removing MacOS specific code and updating dependencies

Test Suite

❖ Generated reports using structural analysis tools and code coverage tools

❖ Created a testing plan based on analysis and coverage

Progress

Heatmap Improvements

❖ Implemented zoom options for Heatmap

❖ Added dependencies which improve performance

Launcher Abstraction

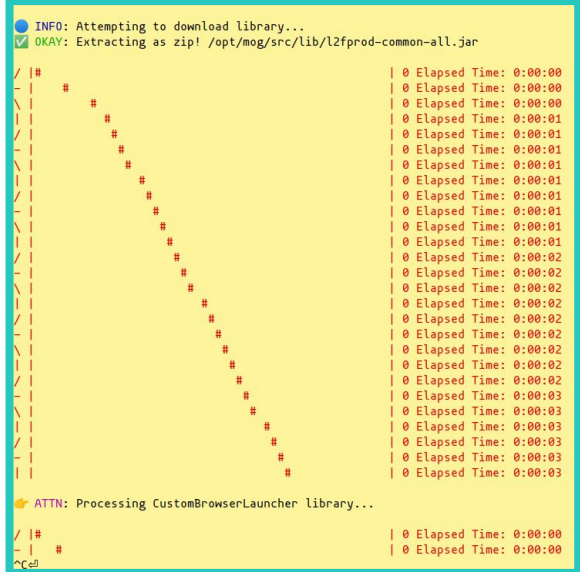❖ Seperate Java program which wraps MOG jar

❖ Use a façade as a normal executable versus Java applet

❖ Checks Java version and prompts user to update

Progress

Docker Build Pipeline

❖ Bundle MOG with Java 11 in Docker image with R

❖ User does not have to worry about requirements other than running Docker

❖ R needs to have EdgeR installed too, can be easily done in Docker
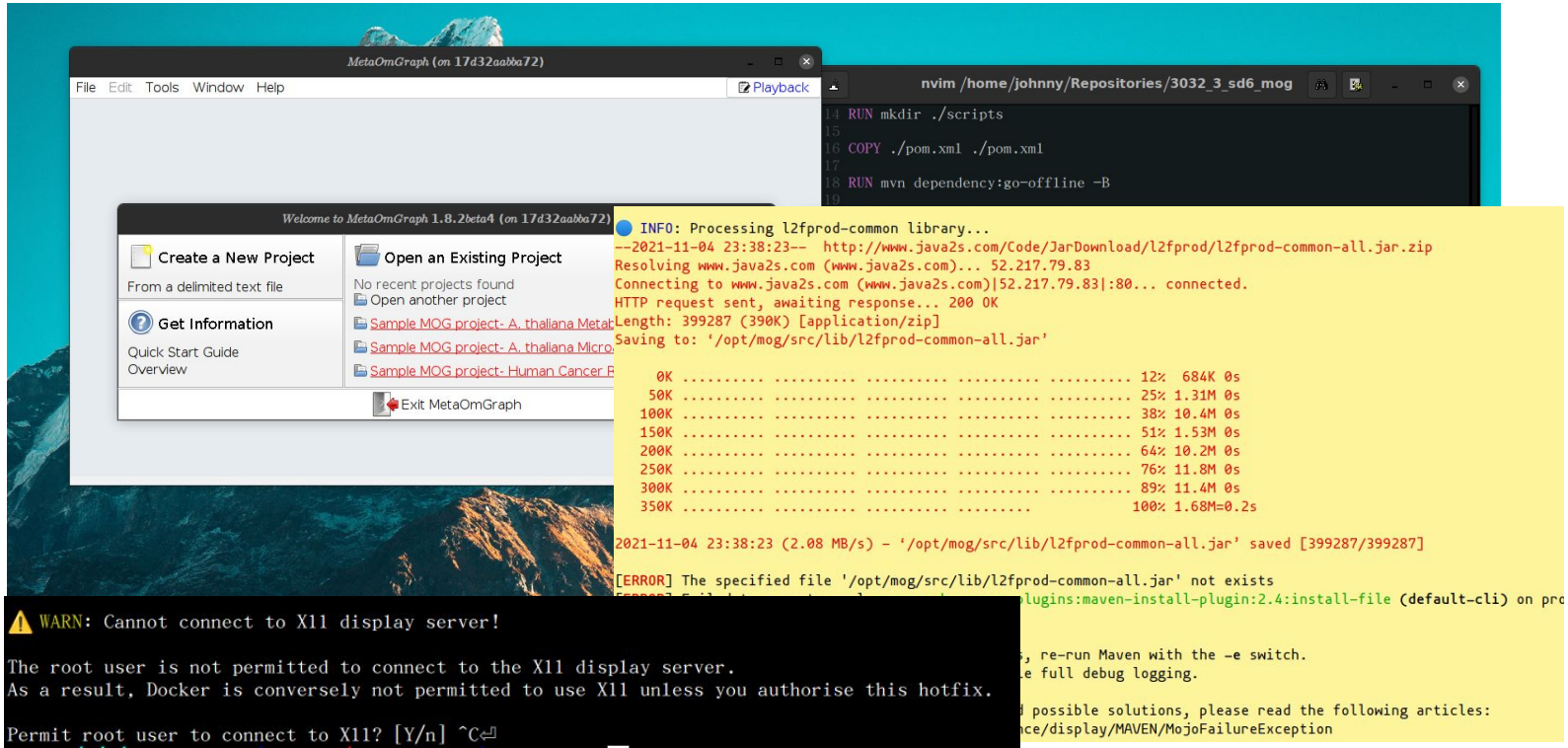

Figure 5

Figure 6

```
⚠ WARN: Cannot connect to X11 display server!

The root user is not permitted to connect to the X11 display server.
As a result, Docker is conversely not permitted to use X11 unless you authorise this hotfix.

Permit root user to connect to X11? [Y/n] ^C↵
```



```
● INFO: Processing l2fprod-common library...
--2021-11-04 23:38:23--  http://www.java2s.com/Code/JarDownload/l2fprod/l2fprod-common-all.jar.zip
Resolving www.java2s.com (www.java2s.com)... 52.217.79.83
Connecting to www.java2s.com (www.java2s.com)|52.217.79.83|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 399287 (390K) [application/zip]
Saving to: '/opt/mog/src/lib/l2fprod-common-all.jar'

    0K .......... .......... .......... .......... .......... 12%  684K 0s
   50K .......... .......... .......... .......... .......... 25% 1.31M 0s
  100K .......... .......... .......... .......... .......... 38% 10.4M 0s
  150K .......... .......... .......... .......... .......... 51% 1.53M 0s
  200K .......... .......... .......... .......... .......... 64% 10.2M 0s
  250K .......... .......... .......... .......... .......... 76% 11.8M 0s
  300K .......... .......... .......... .......... .......... 89% 11.4M 0s
  350K .......... .......... .......... .......... ......      100% 1.68M=0.2s

2021-11-04 23:38:23 (2.08 MB/s) - '/opt/mog/src/lib/l2fprod-common-all.jar' saved [399287/399287]

[ERROR] The specified file '/opt/mog/src/lib/l2fprod-common-all.jar' not exists
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-install-plugin:2.4:install-file (default-cli) on pro
lp 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```

Figure 7

Progress

CI/CD Target Builds

❖ Automate building and testing after every push.

❖ Bundle, build, and push to upstream CI to generate Debian packages, MSI installer, etc.

❖ Easy future work for features like code-signing and content delivery.


```
[I] johnny@velocity ~/R/3/b/debian (main)> dpkg-deb --build metaomgraph4/
dpkg-deb: building package 'metaomgraph4' in 'metaomgraph4.deb'.
[I] johnny@velocity ~/R/3/b/debian (main)> ls
metaomgraph4   metaomgraph4.deb
[I] johnny@velocity ~/R/3/b/debian (main)> sudo dpkg -i   metaomgraph4.deb
[sudo] password for johnny:
Selecting previously unselected package metaomgraph4.
(Reading database ... 255080 files and directories currently installed.)
Preparing to unpack metaomgraph4.deb ...
Unpacking metaomgraph4 (1.0) ...
Setting up metaomgraph4 (1.0) ...
[I] johnny@velocity ~/R/3/b/debian (main)> ./metaomgraph4^C
[I] johnny@velocity ~/R/3/b/debian (main)> metaomgraph4
Hello World!
[N] johnny@velocity ~/R/3/b/debian (main)> sudo dpkg -i   metaomgraph4.deb
```

Figure 8
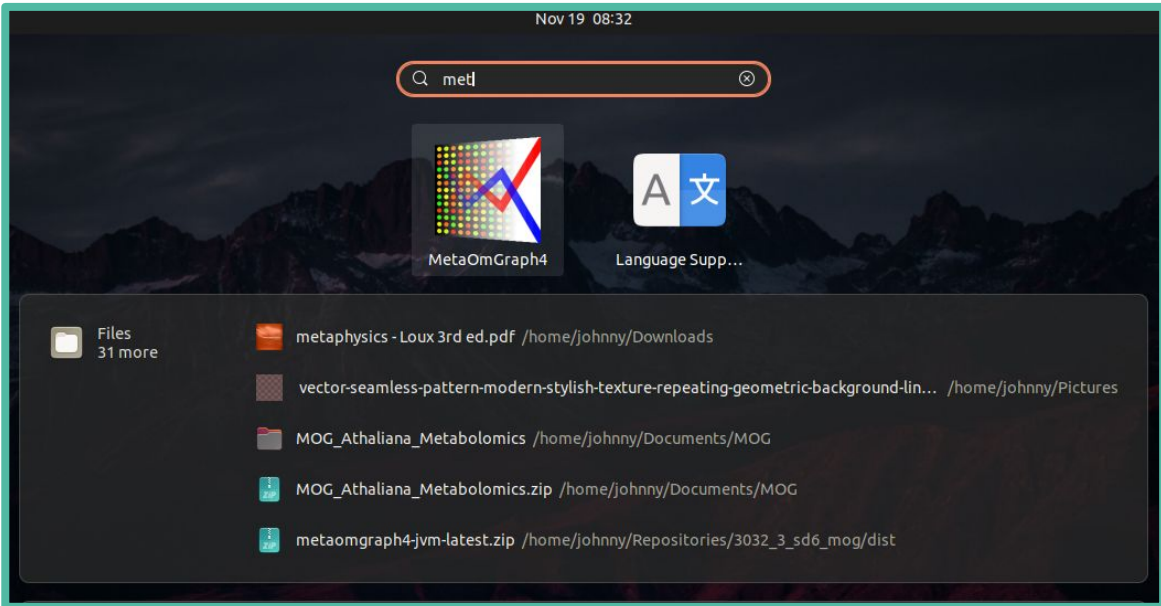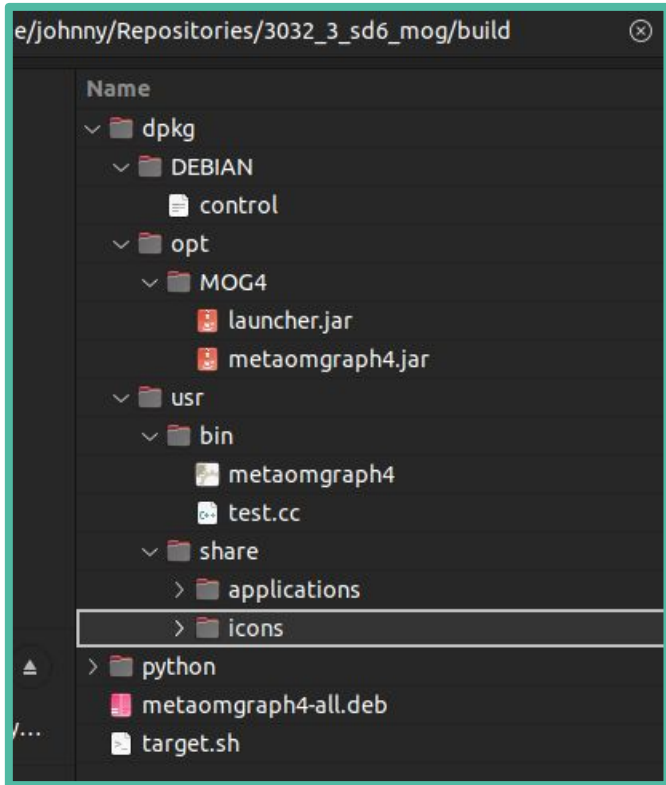
Figure 9

Progress

Website planning

- ❖ Apply UX ideation model to formulate initial product reqs.

- ❖ Utilise the double-waterfall agile-inspired workflow to design website.

- ❖ Document UX research on Notion for high dev availability.
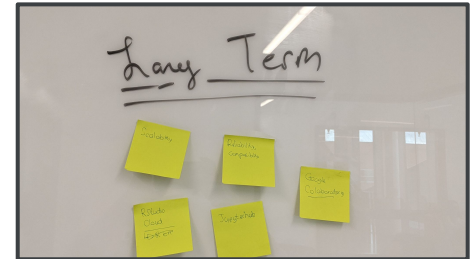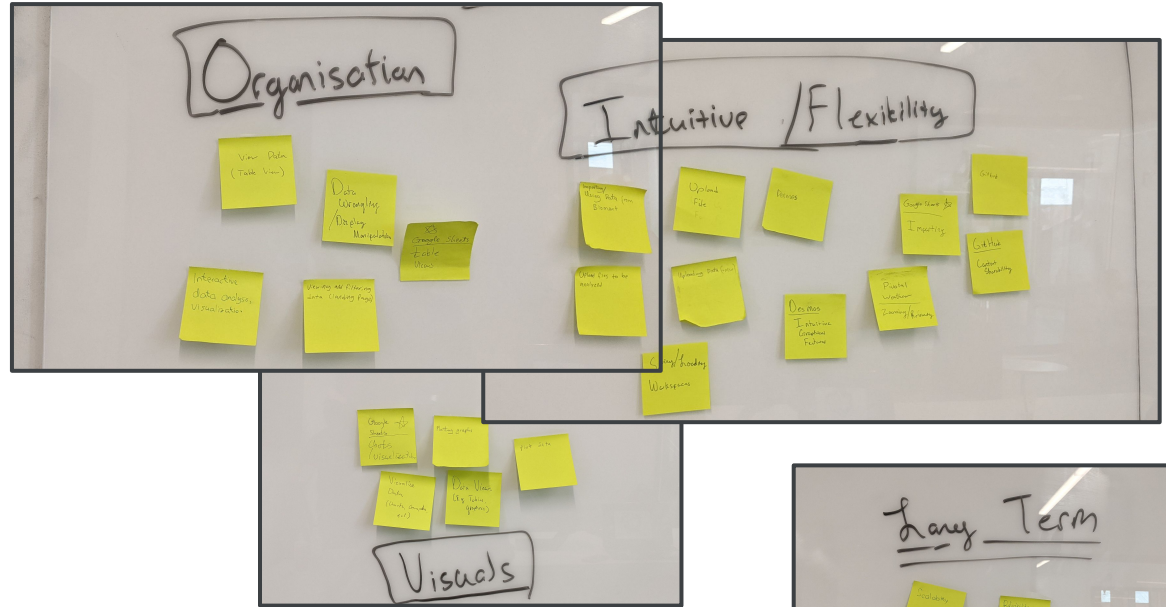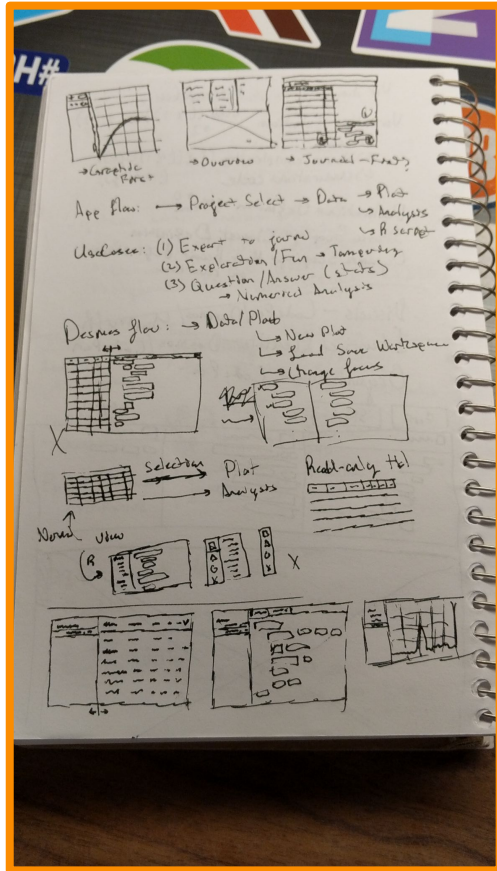
Figure 10

Figure 11

```
launcher-*.jar.part: found 1 matching files and directories
build/*.deb: found 1 matching files and directories
ERROR: Uploading artifacts as "archive" to coordinator... too large archive  id=320958 responseStatus=413 Request Entity Too Large status=413 token=Ty6is-G
FATAL: too large
```

JDK version

Version ≥ 17?

✗ no

✓ yes

continue

UNSUPPORTED JAVA VERSION

You using JDK 1, but JDK 17+ is the oldest Java framework we support.

We can download and setup *JDK 17* for you automatically or you can choose to do so manually. If you choose to allow us to manually setup JDK 17, the binaries will be stored relative to MetaOmGraph.

MANUAL

INSTALL JDK17

Installation Dialogue

JDK 11 INSTALL

Setting up Java JDK 17...   Step: Download content

Step 2/3 (066%)

Downloading content...   download.java.net

openjdk-11.0.1_windows-x64_bin.zip

12,325,232 Kb (010%)

OKAY

Figure 12

Progress

Limma Analysis
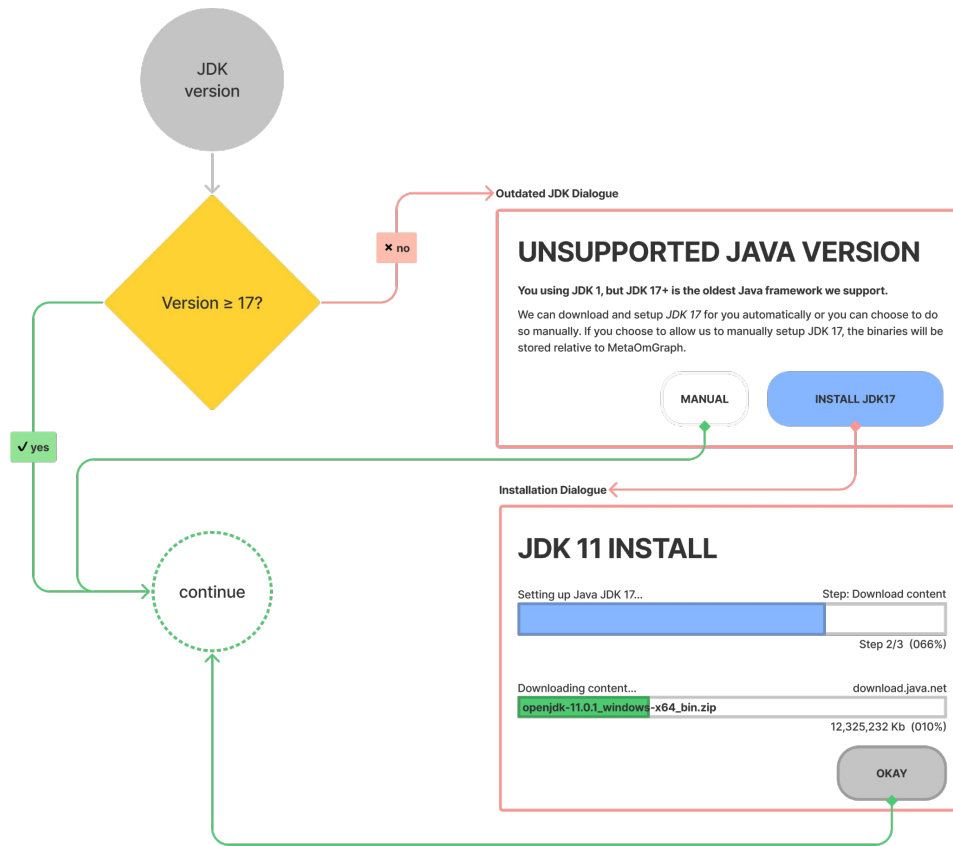
- ❖ Differential analysis but for microarrays (< 100 elements)

- ❖ Calls R program within Java while after processing user inputs using Renjin

- ❖ Generates MDS plot, Voom plot, & differentially expressed genes table

- ❖ Complete GUI to analyze up to 10 groups

- ❖ Modify R source code to handle MDS plot issue

- ❖ Future work: display results & charts in MOG properly
- ❖ Figma workflow

	Gene	logFC	AveExpr	t	P.Value	adj.P.Val	B
2	Mekomodpred01_ID138	-1.300075	12.282562	-4.807979	0.000025	0.006123	2.176492
3	Mekomodpred01_ID007	0.303134	7.604631	3.064756	0.004045	0.158023	-2.117615
4	Mekomodpred01_ID043	0.276023	7.51868	2.905288	0.00615	0.158023	-2.503283
5	glyceric acid	0.367746	7.564463	2.638134	0.012108	0.158023	-3.113563
6	Mekomodpred01_ID266	0.270117	7.328405	2.622274	0.012592	0.158023	-3.160971
7	sucrose	0.236172	7.29403	2.543072	0.015285	0.158023	-3.335895
8	Mekomodpred01_ID147	0.204897	7.227874	2.396125	0.02172	0.158023	-3.648772
9	Mekomodpred01_ID151	0.245834	7.673111	2.36868	0.023165	0.158023	-3.651361
10	Mekomodpred01_ID068	0.220227	7.33652	2.321414	0.025857	0.158023	-3.788582
11	Mekomodpred01_ID076	0.300147	7.298063	2.306507	0.026763	0.158023	-3.824771
12	inositol-1-phosphate	0.254028	8.421879	2.220329	0.03258	0.158023	-3.829225
13	Mekomodpred01_ID107	0.320991	8.712013	2.212583	0.033154	0.158023	-3.787752
14	Mekomodpred01_ID187	0.209863	7.205668	2.188241	0.035018	0.158023	-4.065341
15	Mekomodpred01_ID111	0.336664	7.503875	2.171139	0.036383	0.158023	-4.064261
16	Mekomodpred01_ID220	0.236328	7.030504	2.155932	0.037637	0.158023	-4.135956
17	Mekomodpred01_ID209	0.235642	7.036406	2.153413	0.037848	0.158023	-4.139667
18	citric acid	-0.524726	9.739031	-2.152762	0.037902	0.158023	-3.771739
19	Mekomodpred01_ID093	0.290211	7.226378	2.147953	0.038309	0.158023	-4.138574
20	linoleic acid	0.203854	7.433569	2.14414	0.038634	0.158023	-4.124903
21	L-Serine	-0.419567	17.18935	-2.13681	0.039266	0.158023	-3.714617
22	Mekomodpred01_ID019	0.201536	7.016848	2.135721	0.039361	0.158023	-4.174582
23	Mekomodpred01_ID084	0.195926	7.042558	2.13168	0.039714	0.158023	-4.18079
24	Mekomodpred01_ID201	0.207221	7.124338	2.121991	0.040572	0.158023	-4.194635
25	Mekomodpred01_ID011	0.211685	7.329006	2.108306	0.041812	0.158023	-4.203974
26	palmitic acid	0.183127	7.164231	2.091722	0.043359	0.158023	-4.246882
27	Mekomodpred01_ID223	0.194534	7.032653	2.086157	0.043889	0.158023	-4.266357
28	Mekomodpred01_ID227	0.196973	7.004797	2.081711	0.044316	0.158023	-4.275955
29	Mekomodpred01_ID228	0.201534	7.035411	2.074079	0.045059	0.158023	-4.288546
30	Mekomodpred01_ID029	0.162282	8.475539	2.071313	0.045331	0.158023	-4.091446

Figure 13

Figure 14

Top-left table fragment:

.6024	.7455	.4381	.4402	1.0000
.3733	.5161	.5090	.0174	.5446
.3306	.3000	.2575	.0677	.3159
1.0000	.6107	.2939	.1317	.6024
.5666	.8063	.4538	.2452	.7554
.1835	.5351	.3715	.0624	.5577
.3614	.3099	.4323	.0529	.4907
.5238	.7291	.5448	.0408	.6785

WT
ost 2x increas)

voom.png — 100%

voom: Mean-variance trend

(scatter plot: x-axis "log2(count size + 0.5)", y-axis "Sqrt(standard deviation)")
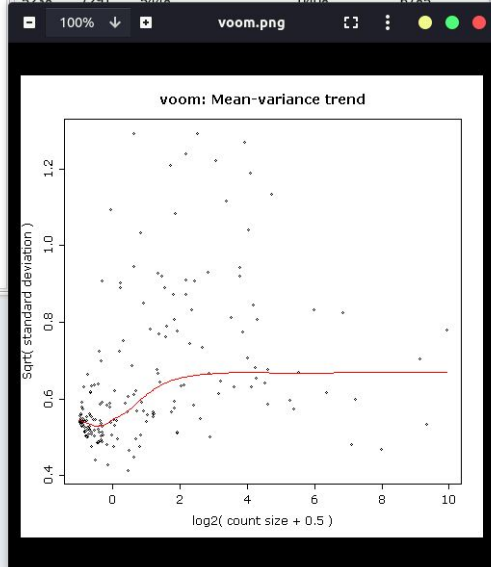
Limma Analysis

Select feature list: Complete List

Group name: Group1 n=24   Move Selected Rows   Group name: Group2 n=26   Move Selected Rows

run	run
CS297-MeKO_02_03_04 replicate 1	CS297-MeKO_02_03_04 replicate 3
CS297-MeKO_02_03_04 replicate 2	CS297-MeKO_02_03_04 replicate 4
Wild type (JP-MeKO)_01_09-2 replicate 9	CS297-MeKO_02_03_04 replicate 5
Wild type (JP-MeKO)_01_09-3 replicate 1	CS297-MeKO_02_03_04 replicate 6
Wild type (JP-MeKO)_01_09-3 replicate 10	CS297-MeKO_02_03_04 replicate 7
Wild type (JP-MeKO)_01_09-3 replicate 11	CS297-MeKO_02_03_04 replicate 8
Wild type (JP-MeKO)_01_09-3 replicate 12	Wild type (JP-MeKO)_01_09-1 replicate 1
Wild type (JP-MeKO)_01_09-3 replicate 13	Wild type (JP-MeKO)_01_09-1 replicate 2
Wild type (JP-MeKO)_01_09-3 replicate 14	Wild type (JP-MeKO)_01_09-1 replicate 3
Wild type (JP-MeKO)_01_09-3 replicate 15	Wild type (JP-MeKO)_01_09-1 replicate 4
Wild type (JP-MeKO)_01_09-3 replicate 16	Wild type (JP-MeKO)_01_09-1 replicate 5
Wild type (JP-MeKO)_01_09-3 replicate 2	Wild type (JP-MeKO)_01_09-1 replicate 6
Wild type (JP-MeKO)_01_09-3 replicate 3	Wild type (JP-MeKO)_01_09-2 replicate 1
Wild type (JP-MeKO)_01_09-3 replicate 4	Wild type (JP-MeKO)_01_09-2 replicate 10
Wild type (JP-MeKO)_01_09-3 replicate 5	Wild type (JP-MeKO)_01_09-2 replicate 11
Wild type (JP-MeKO)_01_09-3 replicate 6	Wild type (JP-MeKO)_01_09-2 replicate 12
Wild type (JP-MeKO)_01_09-3 replicate 7	Wild type (JP-MeKO)_01_09-2 replicate 13
Wild type (JP-MeKO)_01_09-3 replicate 8	Wild type (JP-MeKO)_01_09-2 replicate 14
Wild type (JP-MeKO)_01_09-3 replicate 9	Wild type (JP-MeKO)_01_09-2 replicate 15
Wild type (JP-MeKO)_01_09-4 replicate 1	Wild type (JP-MeKO)_01_09-2 replicate 2
Wild type (JP-MeKO)_01_09-4 replicate 10	Wild type (JP-MeKO)_01_09-2 replicate 3
Wild type (JP-MeKO)_01_09-4 replicate 11	Wild type (JP-MeKO)_01_09-2 replicate 4
Wild type (JP-MeKO)_01_09-4 replicate 12	Wild type (JP-MeKO)_01_09-2 replicate 5
Wild type (JP-MeKO)_01_09-4 replicate 13	Wild type (JP-MeKO)_01_09-2 replicate 6
	Wild type (JP-MeKO)_01_09-2 replicate 7

Add   Remove   Search   Add   Remove   Search
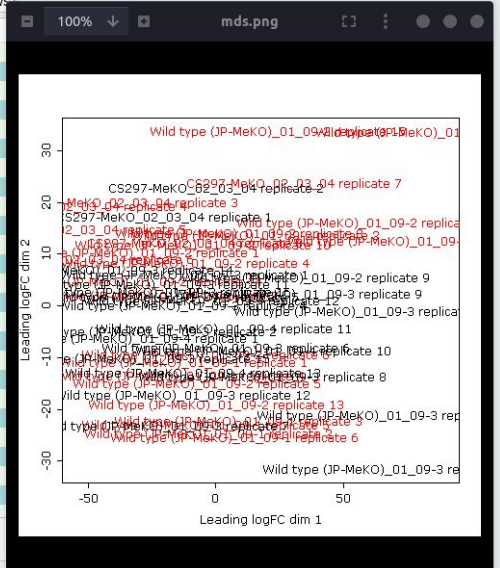
Ok   More   Less

mds.png — 100%

(MDS plot: x-axis "Leading logFC dim 1", y-axis "Leading logFC dim 2", with overlapping red and black sample labels)

Figure 15

```
                    fahmi@rankine:~/3032_3_sd6_mog

> cd 3032_3_sd6_mog
> tokei
=====================================================================
 Language            Files        Lines         Code     Comments      Blanks
=====================================================================
 Autoconf               33        13341        11532          447        1362
 Automake               33         1138          972            0         166
 CSS                     1          574          533           31          10
 Dockerfile              1           53           23           16          14
 Java                  314       109253        83261        10921       15071
 JavaScript              2           78           74            0           4
 Python                  3          201          159            5          37
 R                       1           56           30           10          16
 Shell                  11          797          566           64         167
 Plain Text             56       999130            0       999121           9
 XML                     3          620          617            0           3
---------------------------------------------------------------------
 HTML                  666       183218       167141        16074           3
 |- JavaScript         651        19233        17287         1946           0
 (Total)                         202451       184428        18020           3
---------------------------------------------------------------------
 Markdown                3          231            0          147          84
 |- BASH                 1            2            2            0           0
 (Total)                            233            2          147          84
=====================================================================
 Total                1127      1308690       264908      1026836       16946
=====================================================================
~/3032_3_sd6_mog main >
```

Figure 16

Progress

Java & dependencies upgrades, bug fixes

- ❖ Removed CustomBrowserLauncher dependency

- ❖ Added remote repo for l2fprod-common-all.jar

- ❖ Log4j security fix

- ❖ Upgraded MOG from Java 8 to later versions
  - ➢ Java 9+ no longer supports com.apple.eawt
  - ➢ Replaced with java.awt.Desktop
- ❖ Minor performance upgrades
  - ➢ Replaced unnecessary wrapped data types with primitives (Integer with int, etc)

Progress

Violin Plot

- ❖ Violin plot option now available for MOG users.

- ❖ The Graph is interactive and is generated on the users browser

Memory & CPU Usage

- ❖ System usage can now be viewed on MOG
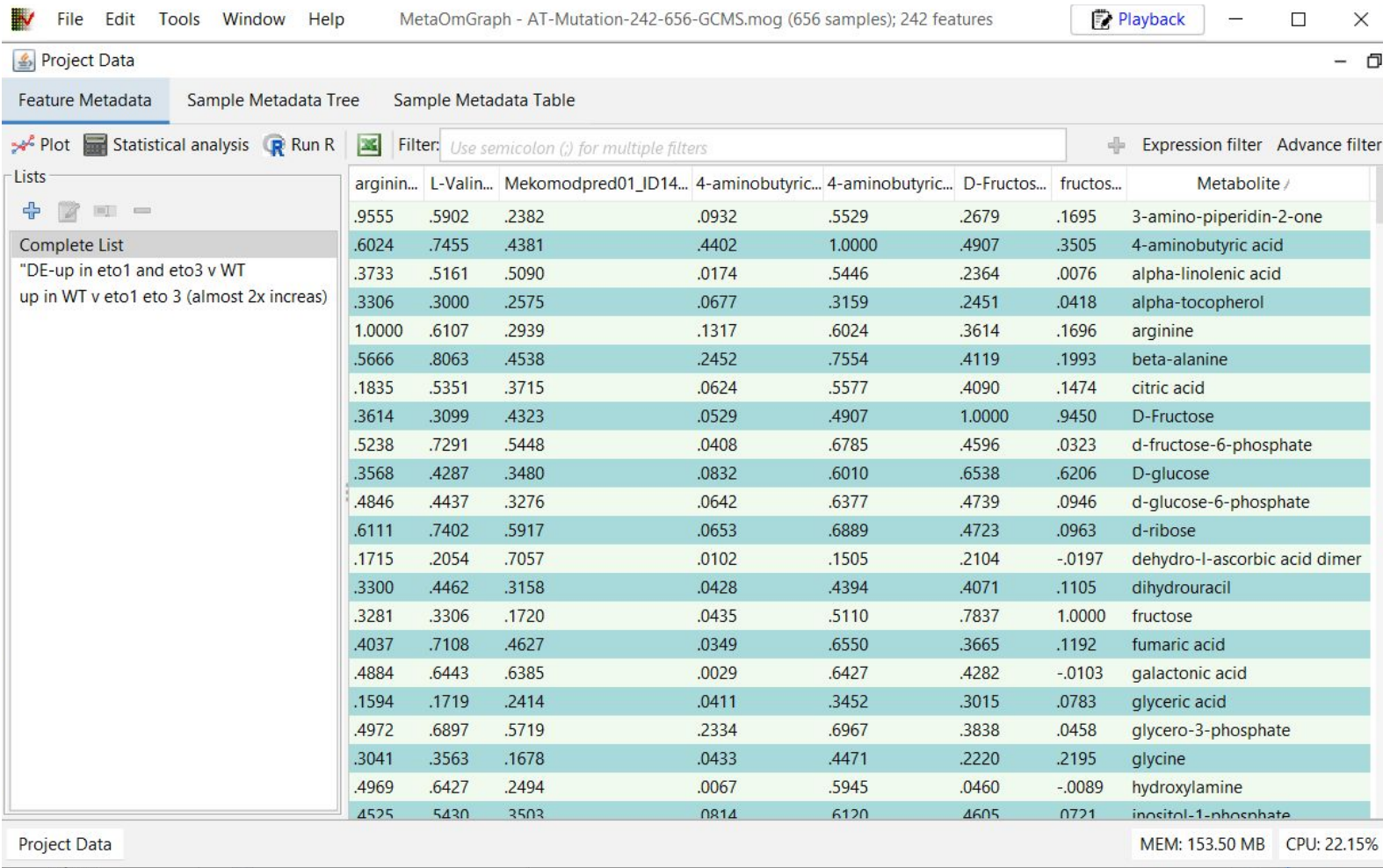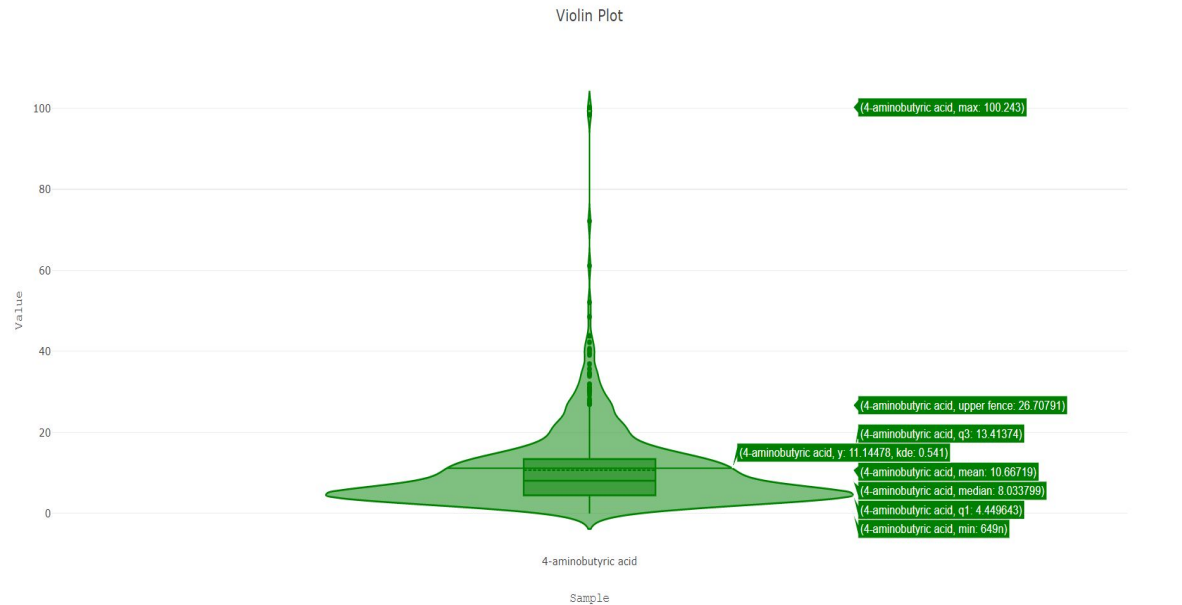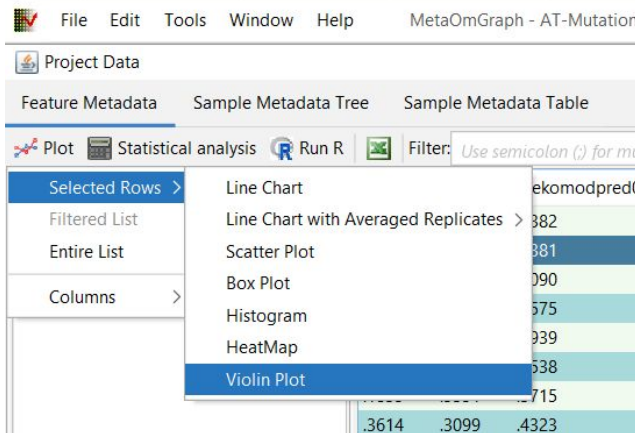
- ❖ MOG memory usage now available on MOG

Figure 17

Figure 18

Progress

Testing Suite

❖ Testing Plan has been created showing analysis using structural analysis from Stan4j

Final Thoughts

What Went Right

- ❖ A lot of tasks were completed

- ❖ We got more comfortable with the codebase

What Went Wrong

- ❖ Communication Issues towards the end

- ❖ Underestimated workload for certain task

Design Limitations

- ❖ Java Swing is bulky
- ❖ Test suite was unavailable
- ❖ Heatmap package didn't have all the features needed
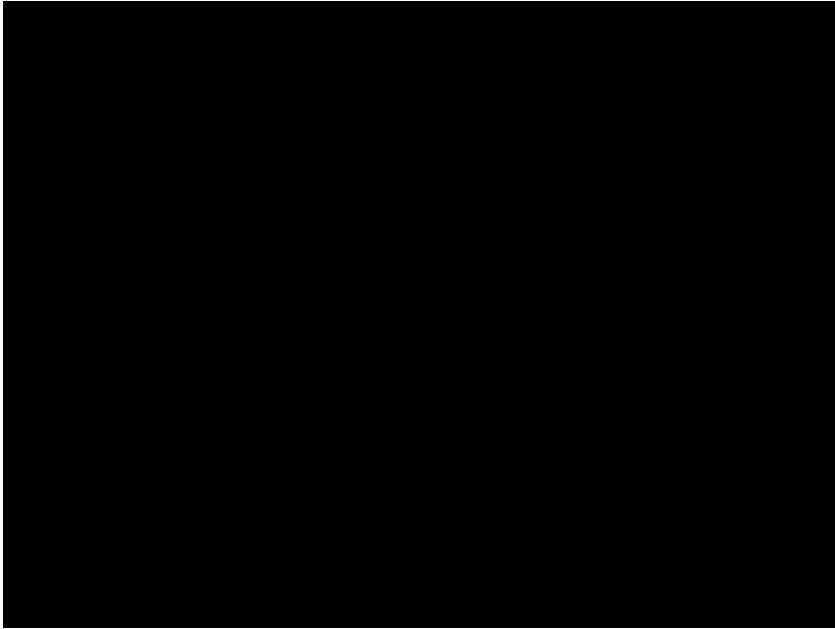- ❖ **TODO:** Docker didn't match our development pipeline

Takeaways

❖ Improve communication
❖ Learn when to place items in a backlog
❖ Match development priority with client's priority
❖ Always make progress

Demo

Launcher

Plotting/Logging/Filtering

Thank you!

Questions?

Adam

Heatmap optimization (Major)

Heatmap improvements (Medium)

Application launcher (Medium)

Maven packaging/POM (Minor)

Java upgrade (Minor)

———

Austin

Java upgrade - medium

Gene expression and mean filter - major

Complex queries - major

Limma UI - medium

Extending Logging and Playback - major

CPU and MEM usage in taskbar - minor

Testing releases on macOS - minor

Helping with Violin and Launcher - minor

Fahmi

- Limma Analysis (Major)
- Dependencies upgrade and compatibility checks, pom cleanup (Medium)
- JDK 11 upgrade (Minor)
- Misc minor bug fixes (Minor)

Kamsi

- Violin Plot (Major)
- Viewing Memory Usage (Major)
- Adding Filter (Medium)
- Test Suite (Minor)

John

- Docker build pipeline (Primary)
- CI/CD scripting (Secondary)
- GUI design (Major)
- UX bug fixes (Minor)