AIRR data analysis with immunarch

Bioinformatics software problems and how to make immune data analysis effortless

AIRR-seq Tool Demonstration | Genoa, Italy | May 2019

Vadim I. Nazarov

National Research University "Higher School of Economics", Russia
ImmunoMind, USA

Table of contents

Two types of software

Living software

Software for People

- thoroughly developed and tested software that solves a real-world task
- addresses the needs of people beyond the laboratory
- regularly updated and supported
- ideal case does not exist, but we all seek for it

Undead software

Software for Software

- in-house scripts re-branded as a trendy new software package
- yet they are often but a combination of lazy workarounds
- in majority of cases the support and updates cease immediately after publication
- due to these facts the software rarely works as expected if works at all

We aimed for the living

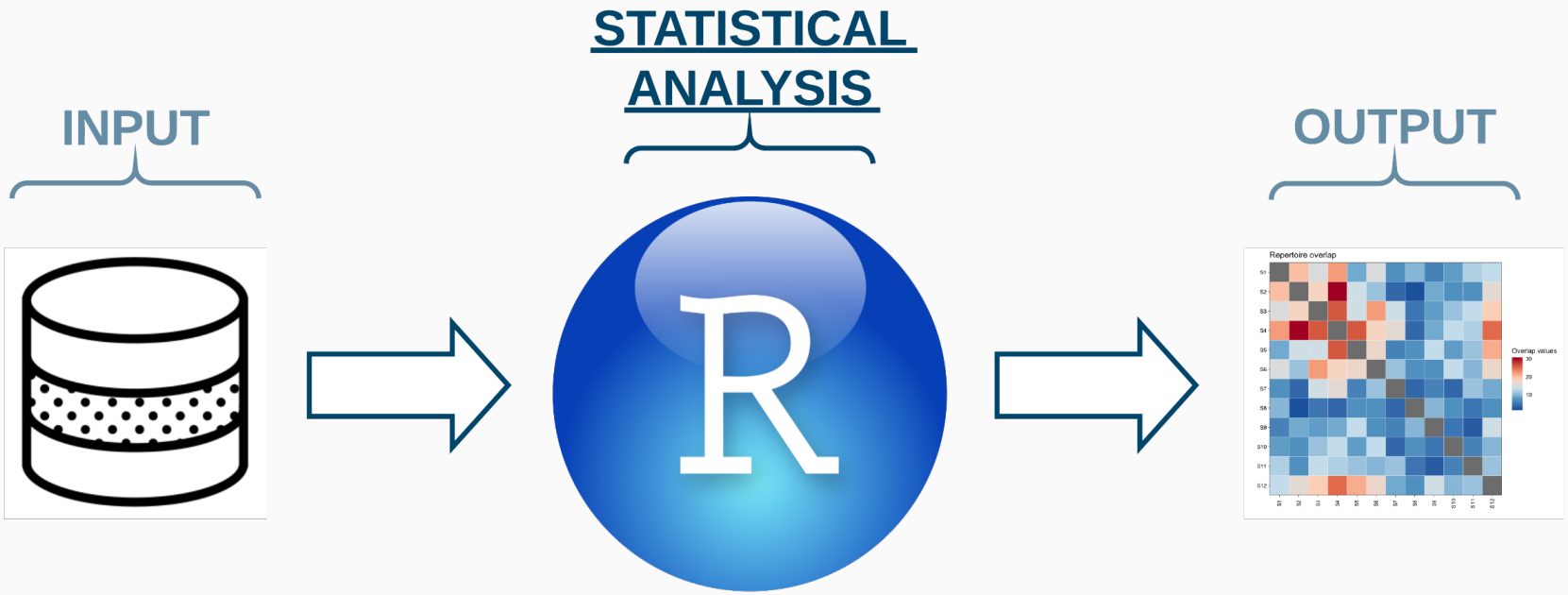
2015 — **tcR**: an R package for AIRR data analysis, **18,000 downloads** over 4 years

2017 — we started to build **immunarch** focusing on **painless AIRR data analysis**

2018 — start of interviews and surveys with researchers from different laboratories

2018—2019 — we revealed a couple of our **over-sights...**

What is "painless AIRR analysis"?



Initial assumption: painless analysis is a fast and effortless computation of common statistics

AIRR data analysis as an one-step process I

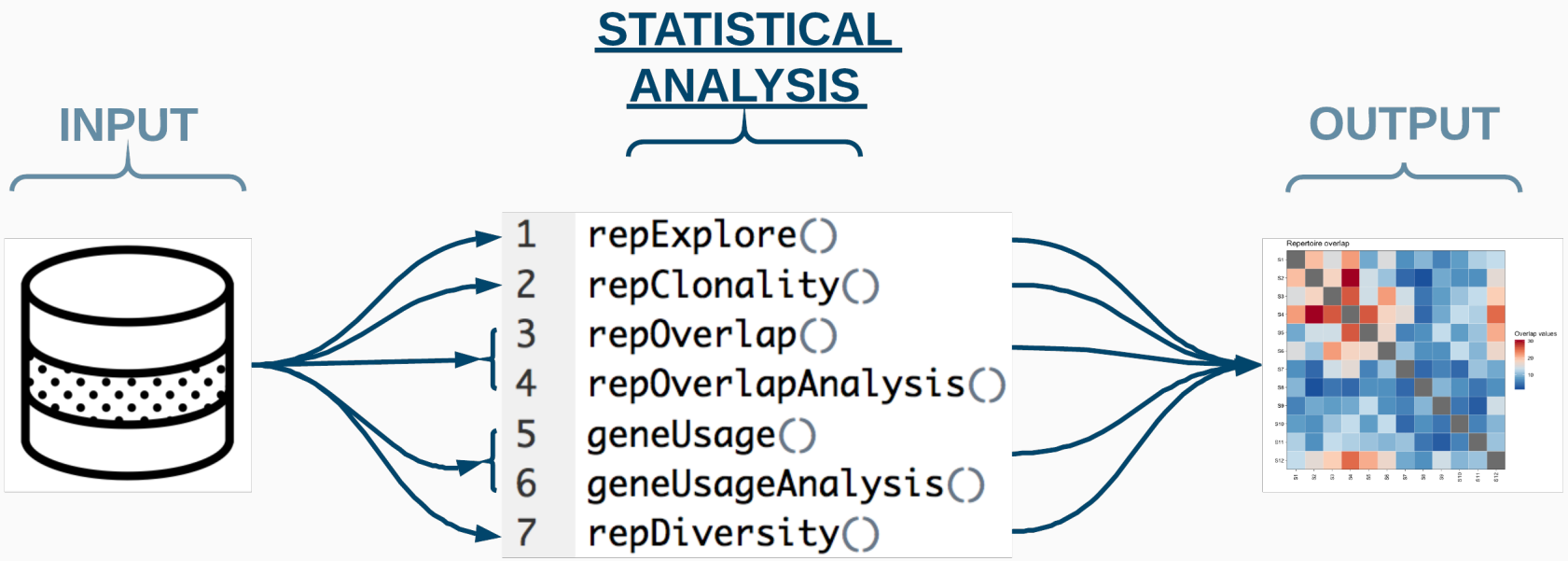
We focused on the statistical analysis of AIRR data first and made it as as effortless as possible:

```
# Overlap of repertoires using Jaccard and Morisita-Horn's indices
result = repOverlap(DATA, "jaccard")
result = repOverlap(DATA, "morisita")

# Jensen-Shannon divergence applied to gene usage
result  = geneUsage(DATA, "HomoSapiens.TRBV")
result2 = geneUsageAnalysis(result, "js")

# Diversity estimation using D50 and Chao1 indices
result = repDiversity(DATA, "d50")
result = repDiversity(DATA, "chao1")
```

AIRR data analysis as an one-step process II



The most important question

Why do scientists suffer when they analyse AIRR data?

AIRR data analysis - parsing I

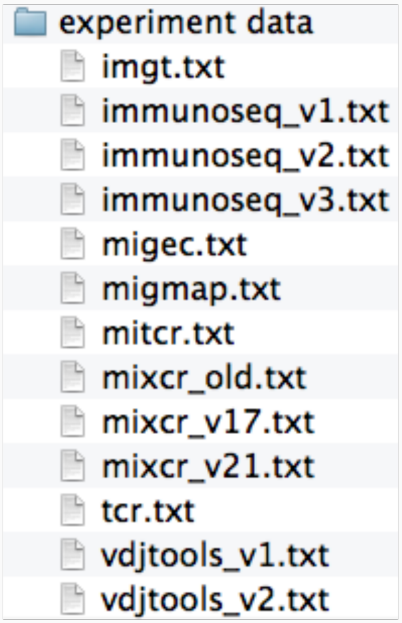
Problem I: Researchers struggle to **load the multiple data formats** into R

And struggle greatly

AIRR data analysis - parsing II

repLoad() - one function to parse them all. It automatically detects the input format and parses it without any specification from the user

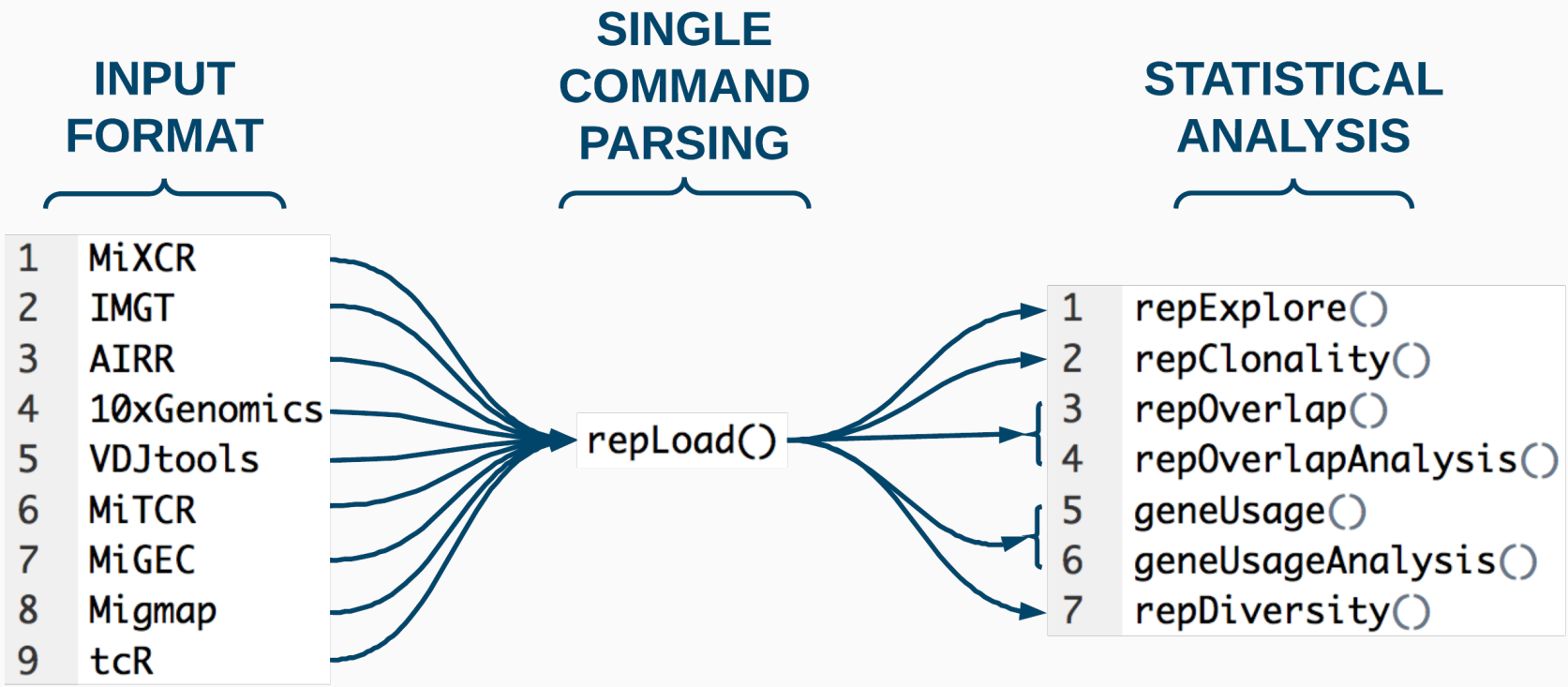
Folder with AIRR data in different formats



R console: repLoad() in action

```
> experiment_data = repLoad("experiment data/")
Parsing experiment data/ ...
Parsing experiment data//imgt.txt -- imgt
Parsing experiment data//immunoseq_v1.txt -- immunoseq
Parsing experiment data//immunoseq_v2.txt -- immunoseq
Parsing experiment data//immunoseq_v3.txt -- immunoseq
Parsing experiment data//migec.txt -- migec
Parsing experiment data//migmap.txt -- migmap
Parsing experiment data//mitcr.txt -- mitcr
Parsing experiment data//mixcr_old.txt -- mixcr
Parsing experiment data//mixcr_v17.txt -- mixcr
Parsing experiment data//mixcr_v21.txt -- mixcr
Parsing experiment data//tcr.txt -- tcr
Parsing experiment data//vdjtools_v1.txt -- vdjtools
Parsing experiment data//vdjtools_v2.txt -- vdjtools
```

AIRR data analysis - the parsing pipeline



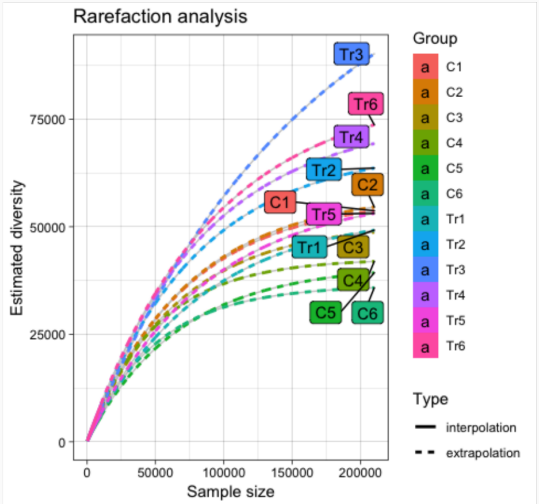
Visualisation of AIRR data analysis I

Problem II: Researchers **constantly need to manipulate visualisations** to make them publication-ready, so we designed tools for easy visualisation

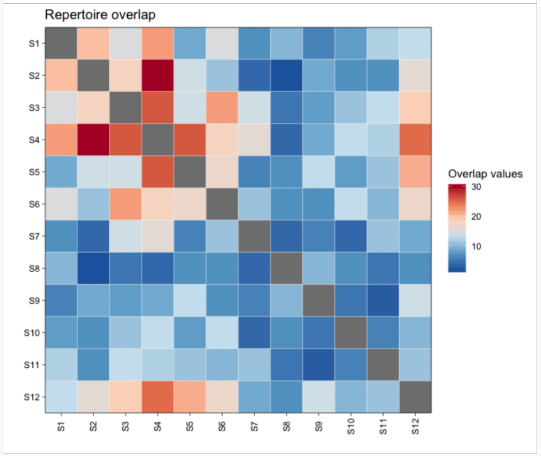
Visualisation of AIRR data analysis II

vis() - one function to plot them all

```
repDiversity(DATA$data, "raref") %>% vis()
```




```
repOverlap(DATA$data) %>% vis()
```



Visualisation of AIRR data analysis III

fixVis() is a tool to manipulate plots painlessly - change font sizes, titles, angles, etc.

FixVis: make your plots publication-ready already!

Save | Plot to R console

General | Title & subtitle | Legends | X axis | Y axis

Title | Subtitle

Subtitle text:

Gene usage calculated after filtering out non

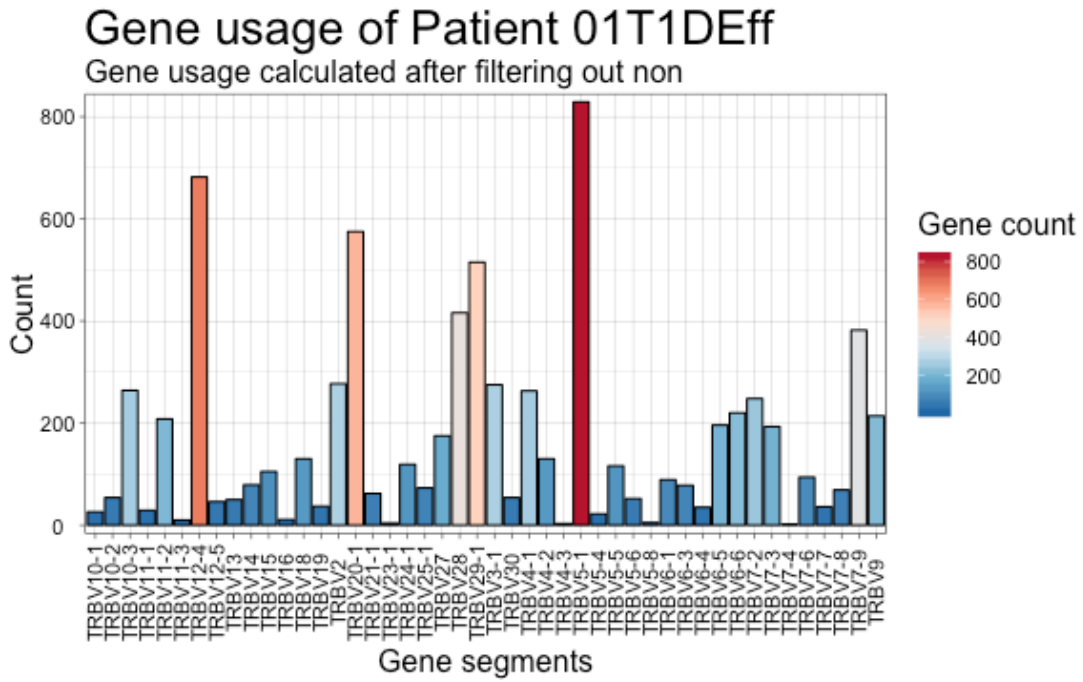
Subtitle text size:

1 16 40

1 5 9 13 17 21 25 29 33 37 40

Subtitle text horizontal adjustment:

0 1



AIRR data analysis - visualisation pipeline

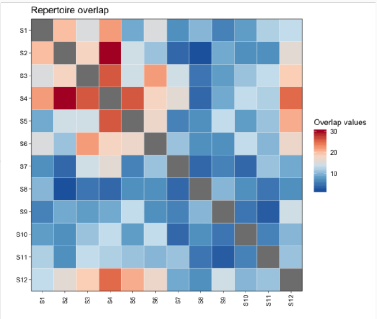
STATISTICAL ANALYSIS

```
1  repExplore()
2  repClonality()
3  repOverlap()
4  repOverlapAnalysis()
5  geneUsage()
6  geneUsageAnalysis()
7  repDiversity()
```

ONE-COMMAND VISUALISATION

```
1  vis()
2  fixVis()
```

PUBLICATION-READY PLOTS



Overall immunarch pipeline

ONE-COMMAND PARSING

STATISTICAL ANALYSIS

ONE-COMMAND VISUALISATION

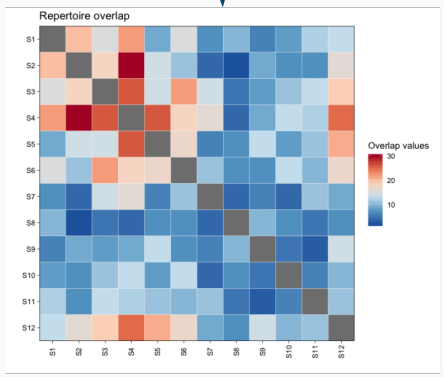
```
repLoad()
```

```
1  repExplore()
2  repClonality()
3  repOverlap()
4  repOverlapAnalysis()
5  geneUsage()
6  geneUsageAnalysis()
7  repDiversity()
```

```
1  vis()
2  fixVis()
```

```
1  MiXCR
2  IMGT
3  AIRR
4  10xGenomics
5  VDJtools
6  MiTCR
7  MiGEC
8  Migmap
9  tcR
```

	Clones	Percentage	Clonal.count.prop
Sample1	18	10.1	0.00238347457627119
Sample2	28	10.1	0.003690523263477
Sample3	9	10.2	0.00121605188488042
Sample4	115	10	0.0145680263491259
Sample5	4	11.7	0.000674081563869228
Sample6	8	10.6	0.00119313944817301
Sample7	3	12.2	0.000477554918815664
Sample8	69	10	0.00831024930747922
Sample9	2	10.8	0.000266311584553928



VARIOUS FORMAT FILES

AIRR DATA STATISTICS

PUBLICATION-READY PLOTS

Pipeline example I

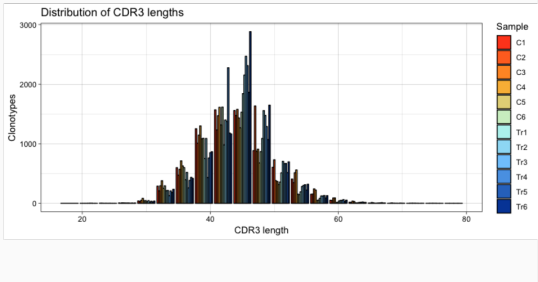
What does it take to perform a typical pipeline in **immunarch**?

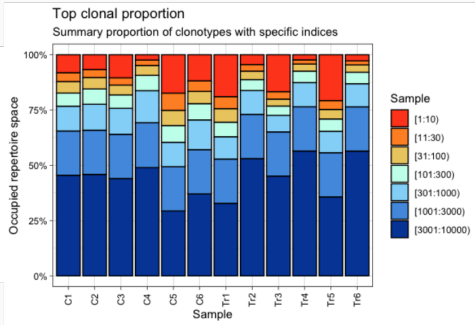
1. Load all the data into R
2. Compute and visualise:
    - 2.1 distribution of CDR3 lengths
    - 2.2 relative abundances of the most frequent clonotypes
    - 2.3 the number of public clonotypes between samples
    - 2.4 Morisita-Horn index values between samples
    - 2.5 V gene usage and group the visualisation by treatment status
    - 2.6 Chao1 diversity metric and group the visualisation by treatment status
    - 2.7 rarefaction curves

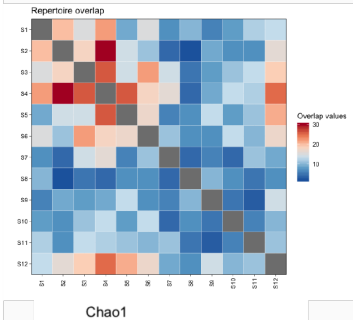
Pipeline example II

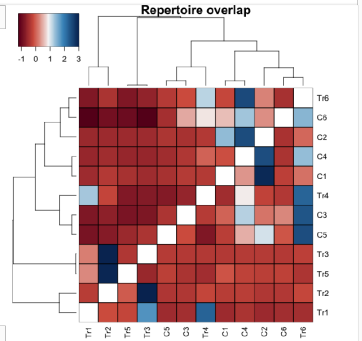
Answer: 8 lines of code and 2 minutes of coding

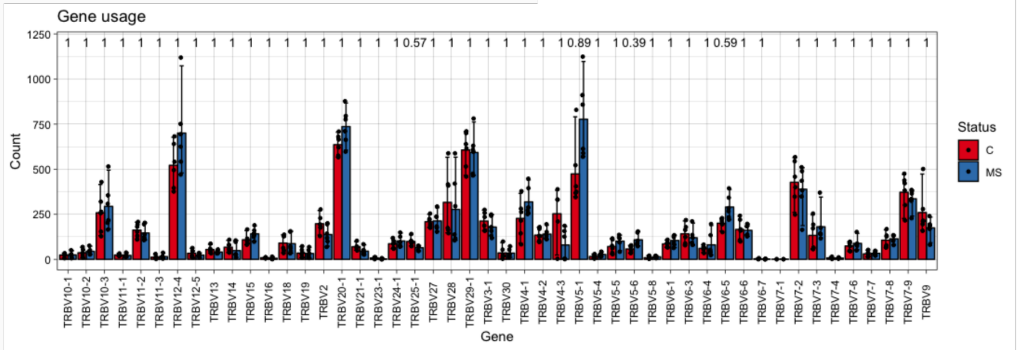
```
DATA = repLoad("path/to/your/folder")
repExplore(DATA$data, "len") %>% vis()
repClonality(DATA$data, "top") %>% vis()
repOverlap(DATA$data) %>% vis()
repOverlap(DATA$data, "morisita") %>% vis(.plot="heatmap2")
geneUsage(DATA$data) %>% vis(.by="Status", .meta=DATA$meta, .grid=F)
repDiversity(DATA$data, "chao1") %>% vis(.by="Status", .meta=DATA$meta)
repDiversity(DATA$data, "rarefaction") %>% vis()
```

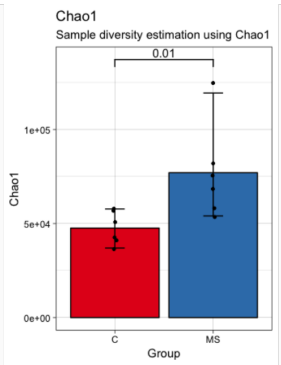


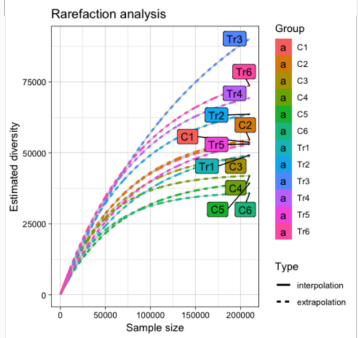












Thank you!

ImmunoMind

- Evgenii Ofitserov
- Sergey Fedyushchenko
- Vadim I. Nazarov

Collaborators

- Vasily Tsvetkov
- Daniel J. Moore, PhD
- Victor Greiff, PhD

Interviewees and surveyees

- Jean-Philippe Bürckert
- Anne Eugster
- Wyatt J. McDonnell
- Anna Lorenc
- Nils-Petter Rudqvist
- Anna Vyborova
- Alexander G. Watson
- Cindy L. Zuleger
- ... and many more

Conclusion & contacts

Want to focus on research and not coding?

Pre-release version is on **https://immunarch.com**

Presentation and news:

Twitter **@immunomind**

Interested in autoimmunity or oncology?

We are looking for partners on biomarker discovery—CDR3 motifs

Vadim I. Nazarov

HSE (RU), ImmunoMind (US)

vadim@immunomind.io

... or in Genoa until **May 19**