R-evolutionary insights

Deep dive into SNP-based population genomics

ANU Guest Conference WiFi
Username: PopGenR
Password: Kioloa_PopGen_24

SESSION 1

Introduction

- Olly Berry – Welcome and Global Perspective [10 mins]
- Andrzej Kilian – Introduction to Diversity Arrays Technology [15 mins]
- Bernd Gruber – The dartRverse [15 mins]
- Arthur Georges – dartR fundamentals -- HANDS ON [55 mins]
- Renee Catullo – More on filtering and the need for a flexible experimental approach to analysis [15 mins]
- Arthur Georges -- Discussion [10 mins]

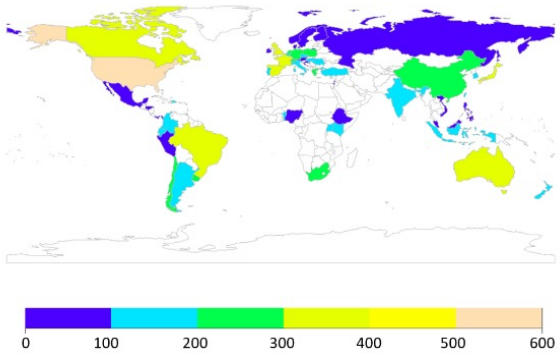
Welcome!

Workshop on population genomics with the dartR platform

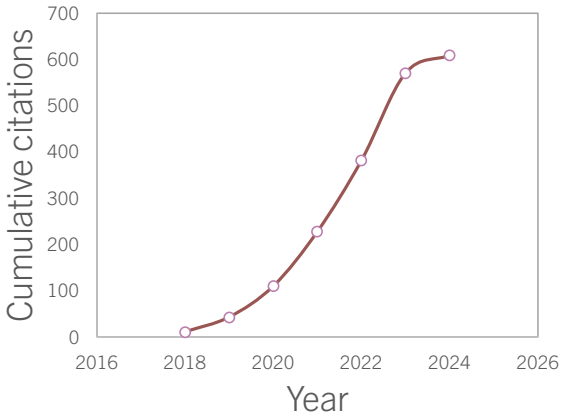
Kioloa, March 11-15, 2024

Oliver Berry
CSIRO Environomics Future Science Platform

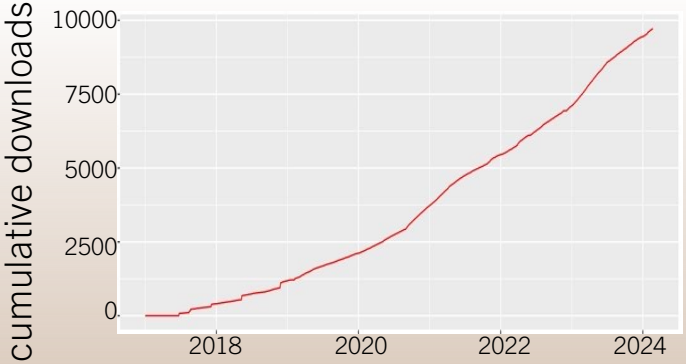
dartR downloads (geographic)



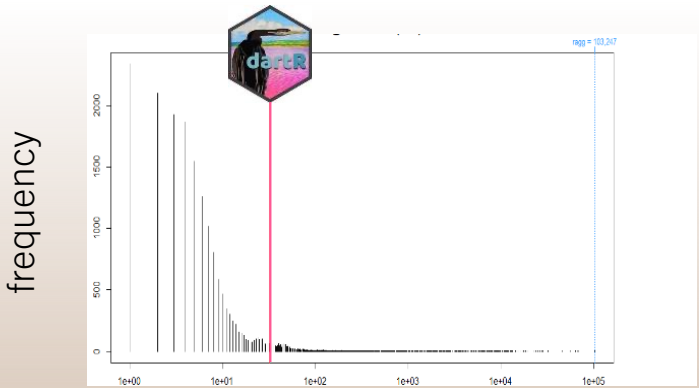
dartR Citations



dartR downloads numbers



dartR in context




dartR
Popgen in R Workshop | 2019 | Hobart
Environomics Future Science Platform
CSIRO
Population Genomics in R Workshop
1st – 5th April, 2019
Hobart, Tasmania
ENVIRONOMICS
Future Science Platform
Admixture model
CSIRO

Main developers

Luis

DArT

Arthur

UNIVERSITY OF CANBERRA

Diana

MONASH University

Bernd

UNIVERSITY OF CANBERRA

Carlo

Arthur Rylah Institute for environmental research

Workshop contributors

UNIVERSITY OF CANBERRA

UNSW SYDNEY

Flinders University

The University of Western Australia

Flinders University

Australian National University

Flinders University

ZSL LONDON ZOO

CSIRO

THE UNIVERSITY OF SYDNEY

THE UNIVERSITY OF SYDNEY

UNIVERSITY OF CANBERRA

DArT

Arthur Rylah Institute for environmental research

DArT

MACQUARIE University

Australian National University

Welcome!

Workshop on population genomics with the dartR platform

Kioloa, March 11-15, 2024

R-evolutionary Insights: Deep Dive into SNP-based Population Genomics

dartR

Analyse DNA like a PRO

UNIVERSITY OF CANBERRA · CSIRO · DArT · Arthur Rylah Institute · MONASH University · THE UNIVERSITY OF WESTERN AUSTRALIA · Flinders University · UNSW SYDNEY · THE UNIVERSITY OF SYDNEY · MACQUARIE University · Australian National University · Centre for Biodiversity Analysis · ZSL LONDON ZOO

Introduction to DArT

Andrzej Kilian
Kioloa PopGen
11 March 2024


Since 2001
Our Mission
Actively support more effective and sustainable use of natural resources in equitable manner.
A social enterprise operating as a successful business

About Us

Genome profiling services and IT support for agriculture and ecology.

→ Cost-effective solutions

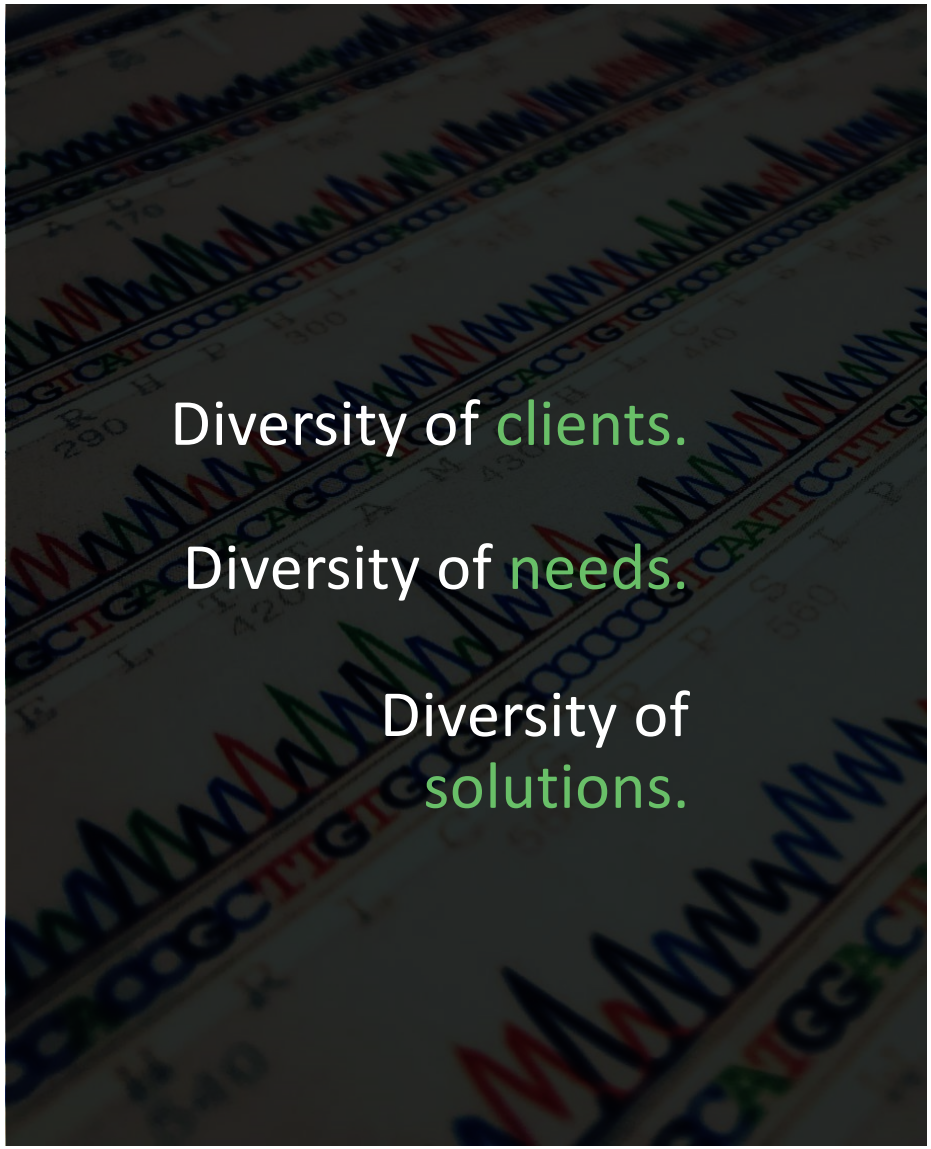
We are the first affordable whole genome profiling service based on DArT, providing end-to-end IT and service management throughout the entire breeding lifecycle.

→ Scalable & independent

We offer high throughput solution for any organism due to a high level of automation, sequence information & platform independence.

→ Integrated and Big Data ready

OneDArT integrates all data types relevant for crop improvement, ecology and agricultural practice



Our Team

A diverse, international team with a mixture of molecular biology, chemistry, electronics, mathematics, stats and IT skills.

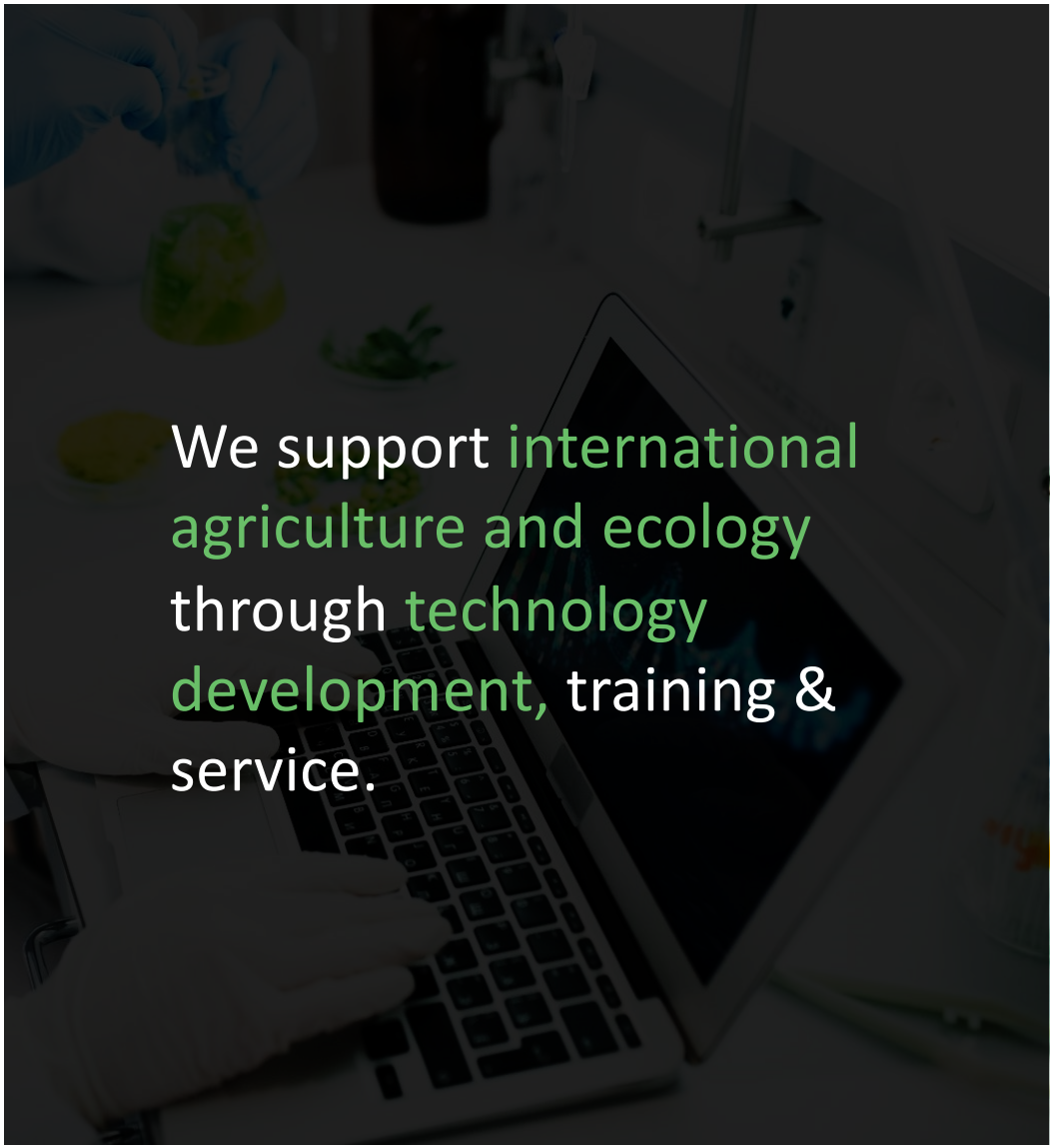
65
staff & growing

Corporate Structure

Organised into three main domains of operations with significant overlap of staff across domains

Our culture

Operating on "chaordic" principles and family-like atmosphere



3.7 million +
genome profiles

1000+
organisms
(thousands of species)

2500+
clients

Global Reach

SAGA, Mexico

Our international footprint spans every continent and over 80 countries.

Genomic analysis Landscape

Started with DNA array-based methods but moved to using Next Generation **Sequencing (NGS)**

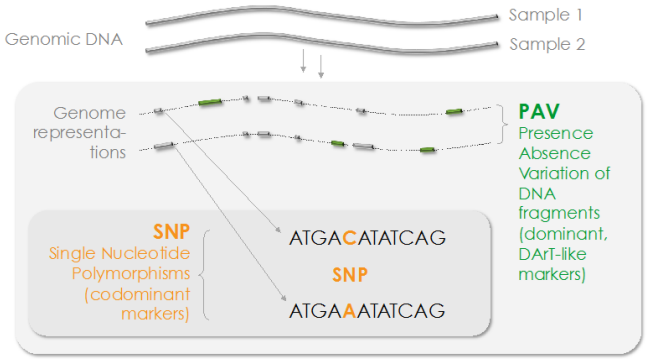
DArTseq - **leading genotyping by sequencing technology**

- Sequencing random genome fragments creating “genome representations”
- Highly scalable – adjust marker number
- High data quality
- Ability to resolve closely related material

Targeted Genotyping - amplicon sequencing

- **DArTag** – 300- 10,000 selected SNPs
  - Predominantly breeding tool, increasingly adopted in ecology
- **DArTcap**– 100- 10,000 selected SNPs used heavily in agriculture and in ecology
- **DArTmp** – up to 300 amplicons sequenced, used for genetic identification and paternity testing

DArTseqMet - DNA methylation analysis both at specific loci and genome-wide

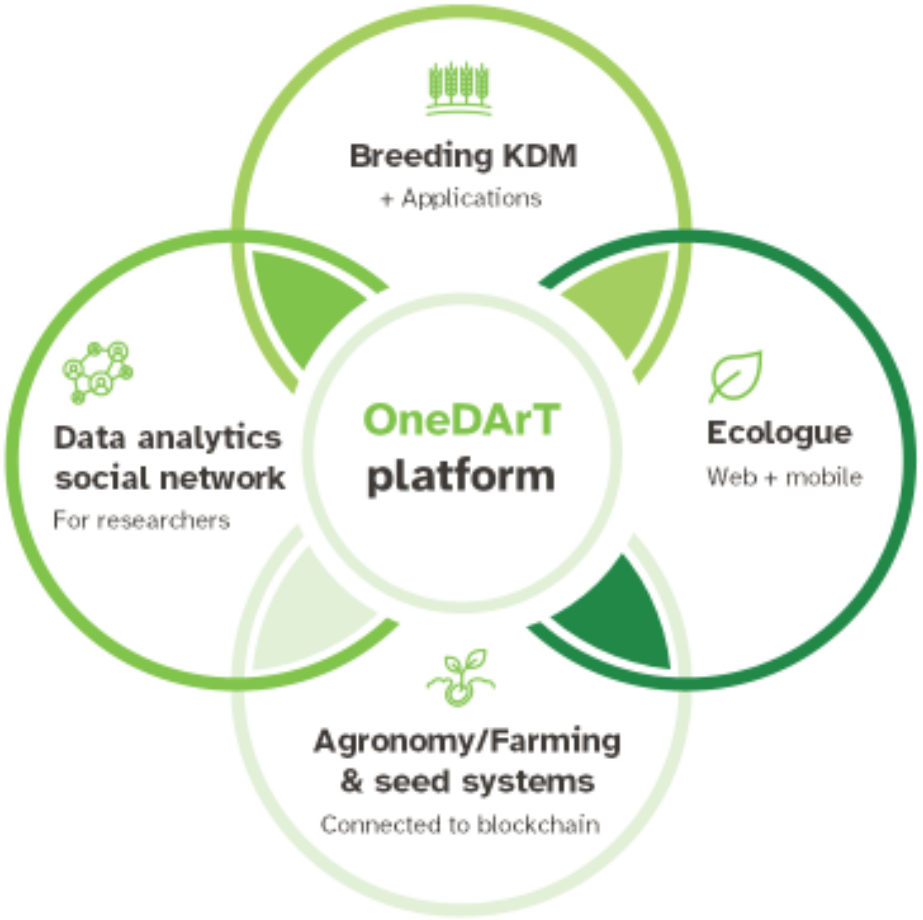


OneDArT platform hub

Meet the client-facing, **super-applications** of the OneDArT platform.

- Genotyping & sequencing services
- Ecommerce platform for **all services**
- **Shared** data sources, schema elements & APIs (DNA, environment, performance)

*Some domain specific



Architecture | Data | UI | Analytics | Configuration

Ecologue
Overview

Target Customer Fields

- Ecology and popgen
- Biosecurity
- Biodiversity conservation
- Ecological Compliance
- Environmental Impact Asessment
- Captive breeding/genetic rescue

Correlate ecological data with genomic data

Ecologue Mobile App

- Collect Ecological Data, including Geographic data, time and place of sample collection
- Run App offline in remote locations, upload data when online later

Ecologue Web App

- Manage, curate & export data
- View data collection location information on maps
- Data analysis using integrated **dartR package**

DArT diversity arrays technology

Stakeholder Engagment

Started as a project for Chevron – platform for biodiversity monitoring on Barrrow Island in Western Australia

Last three years supported by goverment with $3M Priority Investment Program grant

Envisioned as a component of OneDArT platform: from sample and its metadata collection to analytics!

Initial testing and engagement in our target customer fields in preparation for release in second half 2024

External persona

Steven

Ecologist

> Steven is an ecologist, at an ecological & biodiversity consultancy company in Canberra. He manages a team of ecologists in conducting field surveys, assessments and impact analysis of the biodiversity and ecological impact of construction & infrastructure projects in extractive industries to help clients comply with Australian legislation and mitigate negative impact to the environment.

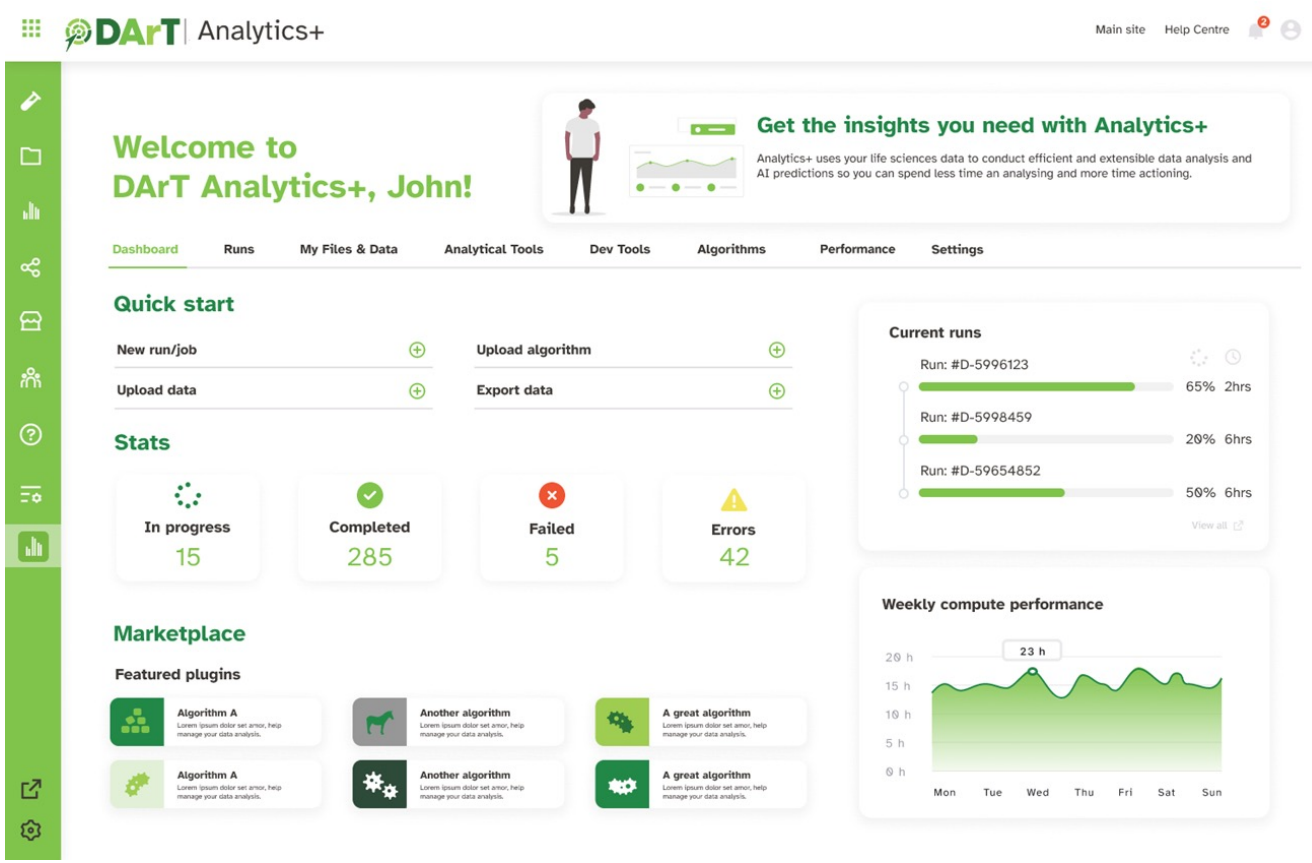
Goals

- To have a streamlined, central, cloud based data management tool where all field survey data is collated and can be exported and analysed
- To provide accurate assessments of development sites to provide advice to clients through a unified analysis framework
- Reduce data loss and increase data re-usability and reproducibility
- Being able to link genotypic information and DNA barcodes to ecological survey data

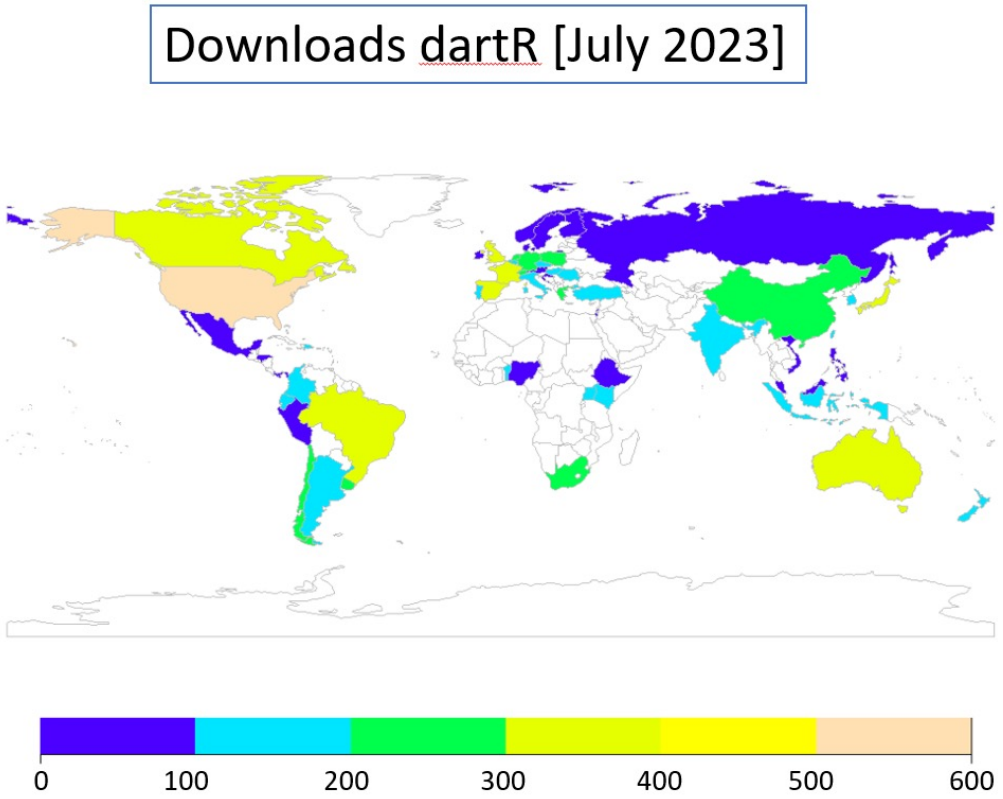
Pain Points

- Not being able to integrate multiple data sources including geographic data e.g. GEOME
- Using multiple, disparate platforms
- Challenges using and managing 'big data' and having the computational power to manage large data sets from different sources e.g. optical sensors, satellite sensors, apps, etc.
- Being able to extract usable information from complex and diverse data sources
- Connecting patterns in the data to ecological processes
- Reliability of traditional genomic tools e.g. phylogenetic trees
- The re-usability of different datasets and survey processes

- **Analytics+ features**
- Users can take data directly from files/database storage
- **dartR** – one of the key applications, especially for ecologists
- Replication of all dartRverse functionalities – complementing and expanding dartR reach
- Reporting Hub – access to reports and format conversion
- DS14 application for marker data extraction from DArTseq libraries + reference genome-based tools
- Universal Test Bench (UTB): ML benchmarking platform for GWAS and genomic prediction tools



The dartRverse



The dartRverse

From 1 to 7 (8)
Why?

- Easier maintenance.
- Faster development.
- Less dependencies on other packages.
- Easier to become a contributor.
- Have your own package developed that can be branded as part of the **dartRverse**.
- [How to install is in the manual]

Overview

+ Input/output
+ filter and report
+ Basic analysis
+ Simple visualisation

Overview

+ Sample data for examples

Overview

+ Simple time forward simulations
+ Mutations, offspring
+ Simple dispersal scenarios
+ Complex processes (overdominance)
+ integrates with slimR

Overview

+ Comprehensive id and filter of sexlinked markers

Overview

+ Functions to study kin-ship
+ Support captive breeding

Overview

+ Studying diversity and population structure
+ Run Structure, Faststructure, Newhybrids, blast
+ Create an SFS
+ Study LD
+ Estimate Ne (Neestimator2)

Overview

+ IBD
+ spatial autocorrelation
+ Resistance surfaces
+ EEMS (estimating effective migration surfaces)

+ To come:
+ Estimate historic population sizes (Epos, Stairways2, Gone)
+ More “genomic” features

How become a contributor?

+ Simply send a script to us

  [via email]

How become a contributor?

+ Simply send a script to us
+ Develop a fully-fledged gl function
    - some requirements
    - good help page
    - tutorial http://georges.biomatix.org/dartR
    - coding evening
+ Get cited and become a package contributor

```
gl.sfs <- function(x,
                   minbinsize = 0,
                   folded = TRUE,
                   singlepop = FALSE,
                   plot.out = TRUE,
                   plot.file = NULL,
                   plot.dir = NULL,
                   verbose = NULL) {
  # SET VERBOSITY
  verbose <- gl.check.verbosity(verbose)

  # SET WORKING DIRECTORY
  plot.dir <- gl.check.wd(plot.dir, verbose = 0)

  # CHECK DATATYPE
  datatype <- utils.check.datatype(x, verbose = verbose)

  # FUNCTION SPECIFIC ERROR CHECKING

  # DO THE JOB


  # Optionally save the plot ---------------------

  # FLAG SCRIPT END

  return(sfs)
  }
  # + a good help page
```

How become a contributor?

+ Simply send a script to us
+ Develop a fully-fledged gl function

  some requirements,

  use github

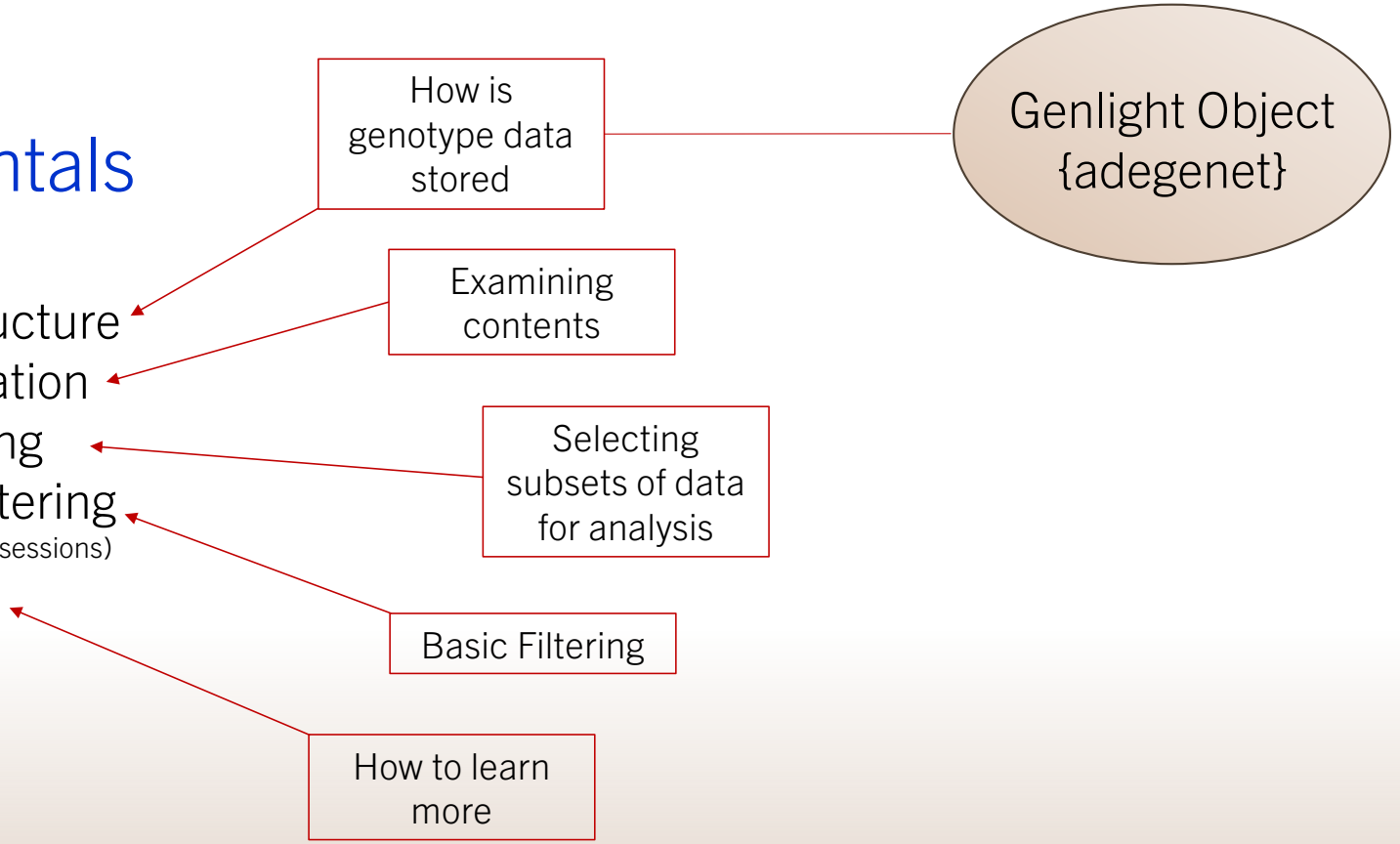
How become a contributor?

+ Simply send a script to us
+ Develop a fully-fledged gl function
+ Create your own package in the dartRverse and become a permanent member of the developer team

```
➢ citation("dartR.sexlinked")
➢ To cite dartR in publications use:
Robledo-Ruiz, D. A., Austin, L., Amos, J. N., Castrejón-Figueroa, J.,
Harley, D. K. P., Magrath, M. J. L., Sunnucks, P., & Pavlova, A. (2023).
Easy-to-use R functions to separate reduced-representation genomic
datasets into sex-linked and autosomal loci, and conduct sex assignment.
Molecular Ecology Resources, 00, 1–21. https://doi.org/10.1111/1755-
0998.13844
```

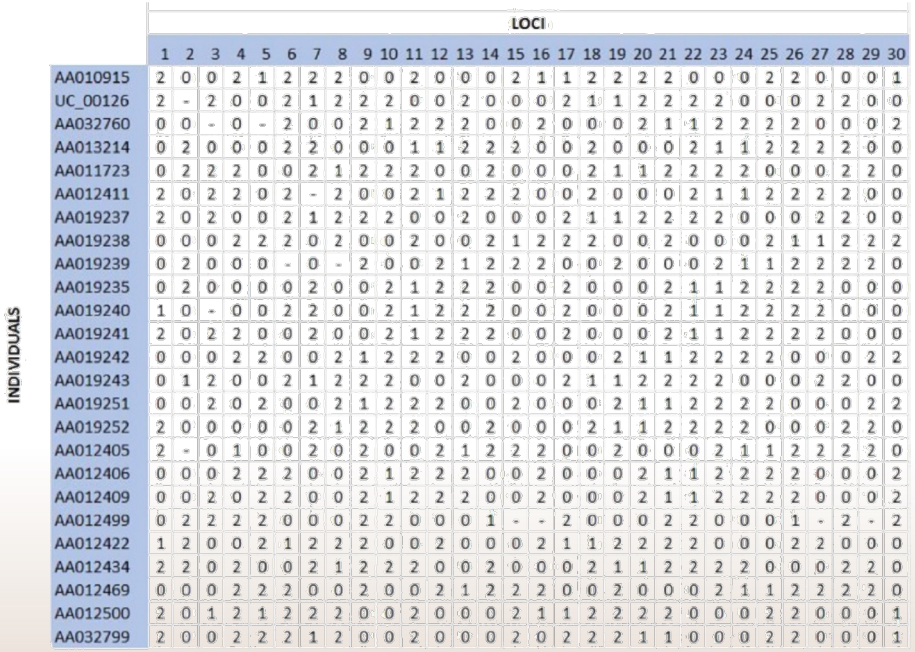
dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials



dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials



INDIVIDUALS	LOCI																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
AA010915	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0	1
UC_00126	2	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA032760	0	0	-	0	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA013214	0	2	0	0	0	2	2	0	0	0	1	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0
AA011723	0	2	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012411	2	0	2	2	0	2	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0
AA019237	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA019238	0	0	0	2	2	2	0	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2
AA019239	0	2	0	0	0	-	0	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0
AA019235	0	2	0	0	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019240	1	0	-	0	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0
AA019241	2	0	2	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0
AA019242	0	0	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA019243	0	1	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA019251	0	0	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA019252	2	0	0	0	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012405	2	-	0	1	0	0	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0
AA012406	0	0	0	2	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA012409	0	0	2	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA012499	0	2	2	2	2	0	0	0	2	2	2	0	0	1	-	-	2	0	0	0	2	2	2	0	0	0	1	-	2	-	2
AA012422	1	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	2	0	0	0	2	2	0	0	0
AA012434	2	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012469	0	0	0	2	2	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0
AA012500	2	0	1	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	2	0	0	0	2	2	0	0	0	1
AA032799	2	0	0	2	2	2	1	2	0	0	2	0	0	0	2	0	2	2	2	1	1	0	0	0	2	2	0	0	0	1

0	Homozygous reference allele
1	Heterozygous
2	Homozygous alternate allele
-	Missing

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Dataframe
LOCUS METADATA
AlleleID, CloneID, AlleleSequence, SNP, SnpPosition, CallRate, OneRatioRef, OneRatioSnp, FreqHomRef, FreqHomSnp, FreqHets, PICRef, PICSnp, AvgPIC, AvgCountRef, AvgCountSnp, RepAvg

Dataframe
INDIVIDUAL METADATA
Latitude Longitude
Maturity Sex

INDIVIDUALS

	LOCI																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
AA010915	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0	1
UC_00126	2	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA032760	0	0	-	0	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA013214	0	2	0	0	0	2	2	0	0	0	1	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0
AA011723	0	2	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012411	2	0	2	2	0	2	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0
AA019237	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA019238	0	0	0	2	2	2	0	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2
AA019239	0	2	0	0	0	-	0	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0
AA019235	0	2	0	0	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019240	1	0	-	0	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019241	2	0	2	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019242	0	0	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2
AA019243	0	1	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA019251	0	0	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2
AA019252	2	0	0	0	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012405	2	-	0	1	0	0	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0
AA012406	0	0	0	2	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA012409	0	0	2	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA012499	0	2	2	2	2	0	0	0	2	2	2	0	0	1	-	-	2	0	0	0	2	2	2	0	0	0	1	-	2	-	2
AA012422	1	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0
AA012434	2	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012469	0	0	0	2	2	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0
AA012500	2	0	1	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0	1
AA032799	2	0	0	2	2	2	1	2	0	0	2	0	0	0	2	0	2	2	2	1	1	0	0	0	2	2	0	0	0	1

0	Homozygous reference allele
1	Heterozygous
2	Homozygous alternate allele
-	Missing

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

gl@other$loc.metrics

Dataframe
LOCUS METADATA
AlleleID, CloneID, AlleleSequence, SNP, SnpPosition, CallRate, OneRatioRef, OneRatioSnp, FreqHomRef, FreqHomSnp, FreqHets, PICRef, PICSnp, AvgPIC, AvgCountRef, AvgCountSnp, RepAvg

gl@other$ind.metrics

Dataframe
INDIVIDUAL METADATA
Latitude Longitude
Maturity Sex

INDIVIDUALS

	LOCI																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
AA010915	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0	1
UC_00126	2	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA032760	0	0	-	0	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA013214	0	2	0	0	0	2	2	0	0	0	1	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0
AA011723	0	2	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012411	2	0	2	2	0	2	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0
AA019237	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA019238	0	0	0	2	2	2	0	2	0	0	2	0	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2
AA019239	0	2	0	0	0	-	0	-	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0
AA019235	0	2	0	0	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019240	1	0	-	0	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019241	2	0	2	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0
AA019242	0	0	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2
AA019243	0	1	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0
AA019251	0	0	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2
AA019252	2	0	0	0	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012405	2	-	0	1	0	0	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0
AA012406	0	0	0	2	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA012409	0	0	2	0	2	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0	0	0	2
AA012499	0	2	2	2	2	0	0	0	2	2	2	0	0	0	1	-	-	2	0	0	0	2	2	0	0	0	1	-	2	-	2
AA012422	1	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0	0
AA012434	2	2	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0
AA012469	0	0	0	2	2	2	0	0	2	0	0	2	1	2	2	2	0	0	2	0	0	0	0	2	1	1	2	2	2	2	0
AA012500	2	0	1	2	1	2	2	2	0	0	2	0	0	0	2	1	1	2	2	2	2	0	0	0	2	2	0	0	0	1
AA032799	2	0	0	2	2	2	1	2	0	0	2	0	0	0	2	0	2	2	2	1	1	0	0	0	2	2	0	0	0	1

0	Homozygous reference allele
1	Heterozygous
2	Homozygous alternate allele
-	Missing

NA

Flags
History
Verbosity Setting
etc

dartR Object

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Follow along if you like. Maybe use your own genlight object?

Quick and Nasty

testset.gl

```
/// GENLIGHT OBJECT /////////

// 250 genotypes,  255 binary SNPs, size: 752 Kb
7868 (12.34 %) missing data

// Basic content
  @gen: list of 250 SNPbin
  @ploidy: ploidy of each individual  (range: 2-2)

// Optional content
  @ind.names:  250 individual labels
  @loc.names:  255 locus labels
  @loc.all:  255 alleles
  @position: integer storing positions of the SNPs
  @pop: population of each individual (group size range: 1-11)
  @other: a list containing: loc.metrics  latlong  ind.metrics  history
     loc.metrics.flags
```

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

{adegenet} Accessors

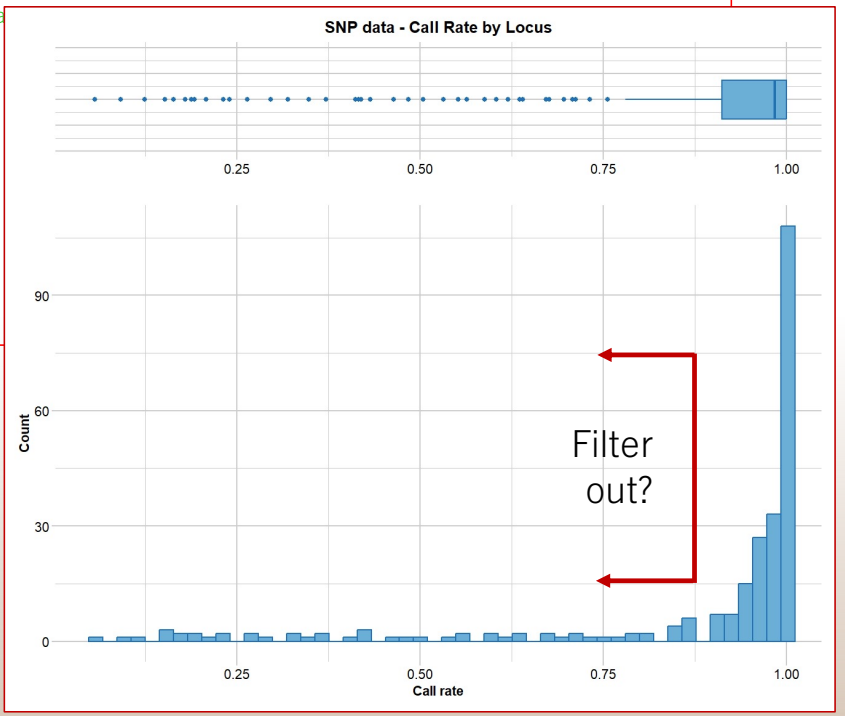
```
gl <- testset.gl
nInd(gl)
[1] 250
nLoc(gl)
[1] 250
nPop(gl)
[1] 30
popNames(gl) or indNames(gl) or locNames(gl)

table(pop(gl))
EmmacBrisWive  EmmacBurdMist  EmmacBurnBara  EmmacClarJack  EmmacClarYate  EmmacCoopAvin  EmmacCoopCully
           10             10             11              5              5             10             10
EmmacCoopEulb  EmmacFitzAllig  EmmacJohnWari  EmmacMaclGeor  EmmacMaryBoru  EmmacMaryPetr  EmmacMDBBowm
           10             10             10             11              6              4             10
EmmacMDBCond  EmmacMDBCudg  EmmacMDBForb  EmmacMDBGwyd  EmmacMDBMaci EmmacMDBMurrMung  EmmacMDBSanf
           10             10             11             9             10             10             10
EmmacNormJack  EmmacNormLeic  EmmacNormSalt  EmmacRichCasi     EmmacRoss  EmmacRussEube  EmmacTweeUki
           6              1              1             10             10             10             10
EmsubRopeMata  EmvicVictJasp
            5             5

as.matrix(gl)[1:7,1:5]
         100049687-12-C/T 100049698-16-G/A 100049728-23-A/G 100049805-56-T/A 100049816-51-A/G
AA010915                2               NA                0                0                0
UC_00126                2               NA                0                0                0
AA032760               NA               NA                0                0                0
AA013214                2               NA                0                0                0
AA011723                2               NA                0                0                0
AA012411                2               NA                0                0                0
AA019237                2               NA                0                0                0
```

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

gl.report functions

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
[Starting gl.report.callrate
  Processing genlight object with SNP da
  Reporting Call Rate by Locus
  No. of loci = 255
  No. of individuals = 250
    Minimum    : 0.056
    1st quartile : 0.912
    Median     : 0.984
    Mean       : 0.8765804
    3r quartile : 1
    Maximum    : 1
    Missing Rate Overall: 0.1234

Completed: gl.report.callrate
```



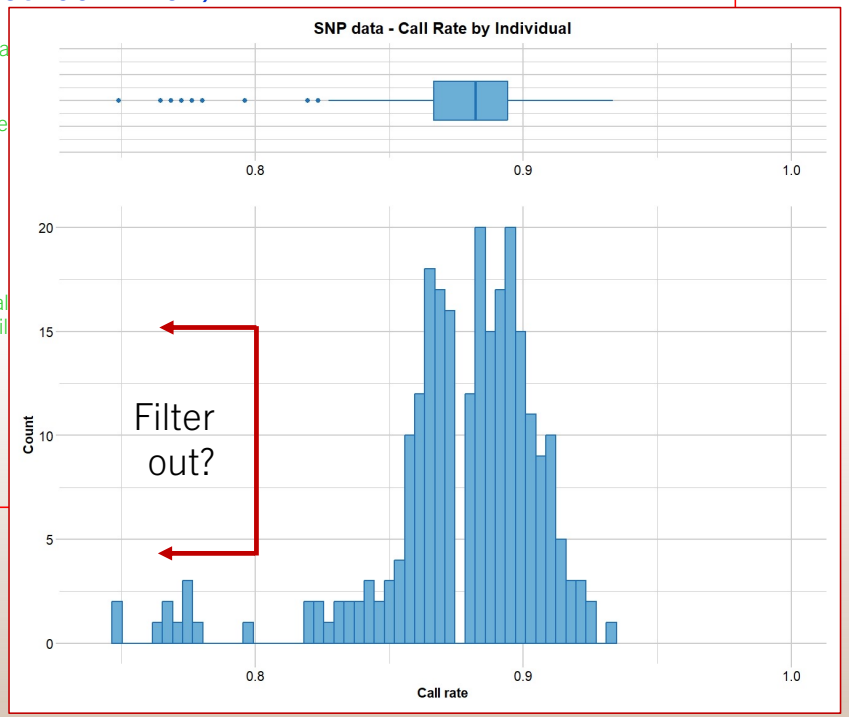
dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

gl.report functions

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
Starting gl.report.callrate
 Processing genlight object with SNP da
 Reporting Call Rate by Individual

Listing 30 populations and their average
 Monitor again after filtering
      Population CallRate  N
1   EmmacBrisWive   0.8839 10
2   EmmacBurdMist   0.8808 10
3   EmmacBurnBara   0.8859 11
     ..........
Listing 20 individuals with the lowest Cal
 Use this list to see which individuals wil
Individual  CallRate
1   AA063722 0.7490196
2   AA063726 0.7490196
3   AA063732 0.7647059
 ..........
Completed: gl.report.callrate
```

Who will be lost?



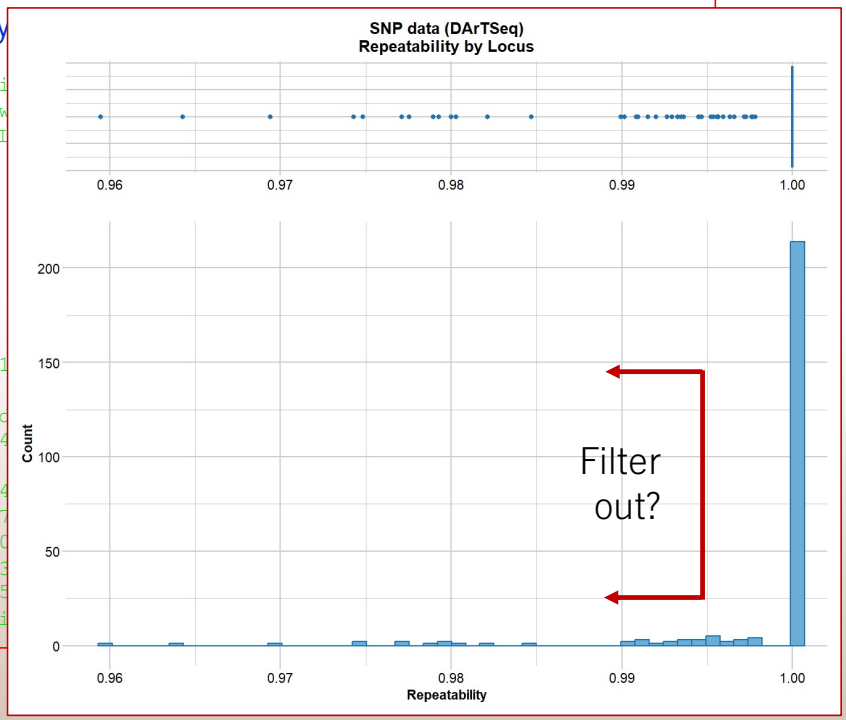
dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

gl.report functions

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility
```

```
gl.report.reproducibility(gl)
Starting gl.report.reproducibi
  Processing genlight object w
  Reporting Repeatability by I
  No. of loci = 255
  No. of individuals = 250
    Minimum      :  0.959459
    1st quartile :  1
    Median       :  1
    Mean         :  0.9981525
    3r quartile  :  1
    Maximum      :  1
    Missing Rate Overall:  0.1

   Quantile Threshold Retained
1      100%  1.000000      214
.....
17      20%  1.000000      214
18      15%  0.997674      217
19      10%  0.994536      230
20       5%  0.984694      243
21       0%  0.959459      255
Completed: gl.report.reproduci
```



dartR Fundamentals

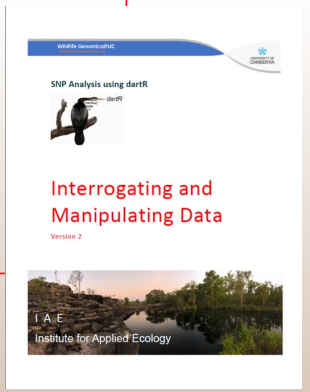
- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Try some of these in your own time

You get the gist . . . .

- gl.report.callrate()
- gl.report.reproducibility()
- gl.report.secondaries()
- gl.report.rdepth()
- gl.report.monomorphs()
- gl.report.overhang()
- gl.report.hamming()
- gl.report.overshoot()
- gl.report.locmetric()

. . . . . etc



Keep up . . .

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

dartR Fundamentals

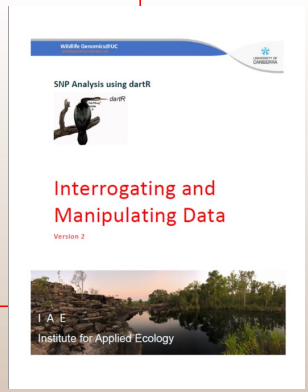
- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Try some of these in your own time

Subsetting your data

- gl.keep.ind()
- gl.drop.ind()
- gl.keep.loc()
- gl.drop.loc()
- gl.keep.pop()
- gl.drop.pop()
- gl.merge.pop()
- gl.subsample.ind()
- gl.subsample.loc()

…… etc



Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Recall …..

- gl.**report**.callrate()
- gl.**report**.reproducibility()
- gl.**report**.secondaries()
- gl.**report**.rdepth()
- gl.**report**.monomorphs()
- gl.**report**.overhang()
- gl.**report**.hamming()
- gl.**report**.overshoot()
- gl.**report**.locmetric()

…….. etc

Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Basic Filtering

- gl <- gl.**filter**.callrate()
- gl <- gl.**filter**.reproducibility()
- gl <- gl.**filter**.secondaries()
- gl <- gl.**filter**.rdepth()
- gl <- gl.**filter**.monomorphs()
- gl <- gl.**filter**.overhang()
- gl <- gl.**filter**.hamming()
- gl <- gl.**filter**.overshoot()
- gl <- gl.**report**.locmetric()

  …… etc

Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

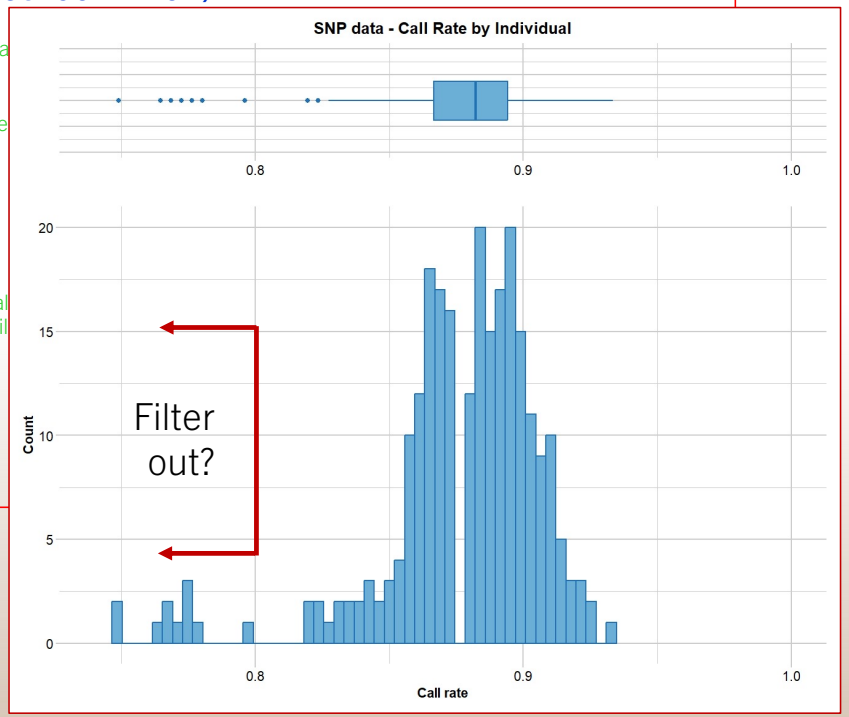
dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- **Basic Filtering**
  (refer Renee's sessions)
- Tutorials

Recall ….

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
Starting gl.report.callrate
 Processing genlight object with SNP da
 Reporting Call Rate by Individual

Listing 30 populations and their average
 Monitor again after filtering
      Population CallRate  N
1   EmmacBrisWive  0.8839 10
2   EmmacBurdMist  0.8808 10
3   EmmacBurnBara  0.8859 11
     ……..
Listing 20 individuals with the lowest Cal
 Use this list to see which individuals wil
Individual  CallRate
1   AA063722 0.7490196
2   AA063726 0.7490196
3   AA063732 0.7647059
 ……..
Completed: gl.report.callrate
```

Who will be lost?



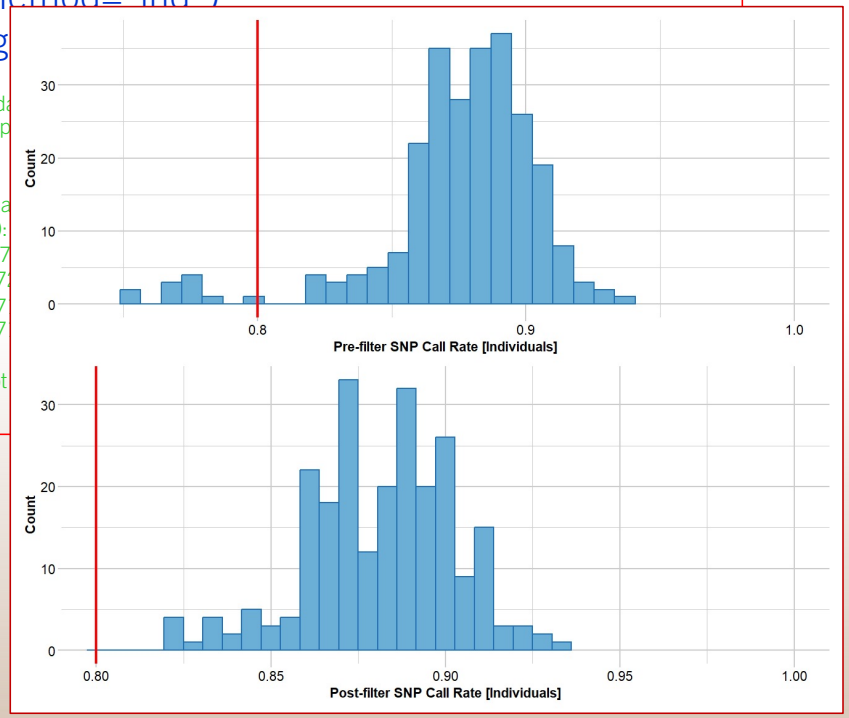
dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

gl.filter functions

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(g
Starting gl.filter.callrate
  Processing genlight object with SNP da
  Warning: Data may include monomorp
           calculations for filtering
  Recalculating Call Rate
  Removing individuals based on Call Ra
  Individuals deleted (CallRate <= 0.8 ):
AA032760[EmmacMDBMaci], AA0637
AA063722[EmmacCoopAvin], AA0637
AA063708[EmmacCoopAvin], AA0637
AA063714[EmmacCoopAvin], AA0637
  Note: Locus metrics not recalculated
  Note: Resultant monomorphic loci not
Completed: gl.filter.callrate
```

Who has been discarded?

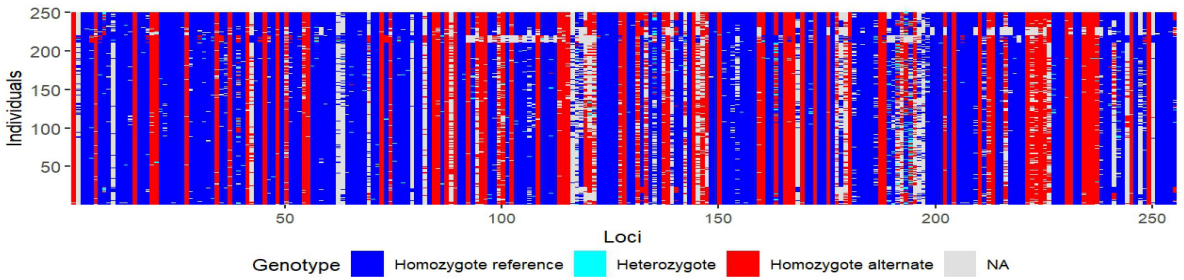


dartR Fundamentals

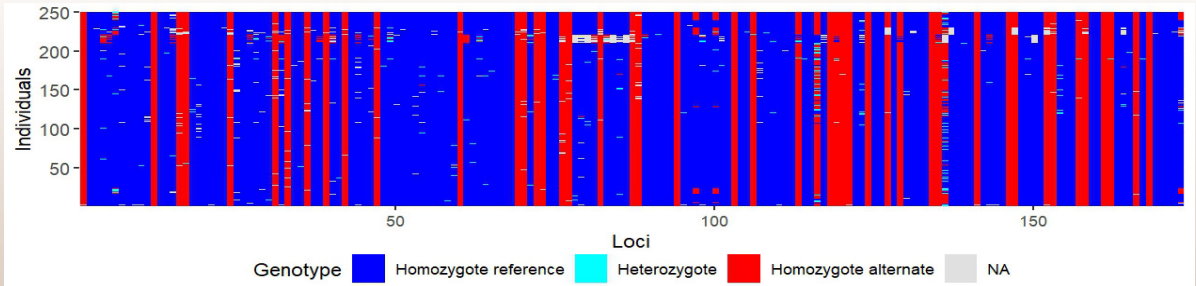
- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Quick graphical evaluation

```
gl <- testset.gl
gl.smearplot(gl)
```




```
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,method="ind",threshold=0.80,verbose=0)
gl.smearplot(gl)
```



dartR Fundamentals

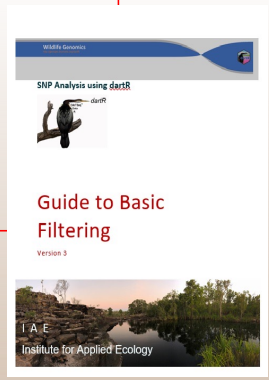
- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials

Try some of these in your own time

Basic Filtering

- gl <- gl.**filter**.callrate()
- gl <- gl.**filter**.reproducibility()
- gl <- gl.**filter**.secondaries()
- gl <- gl.**filter**.rdepth()
- gl <- gl.**filter**.monomorphs()
- gl <- gl.**filter**.overhang()
- gl <- gl.**filter**.hamming()
- gl <- gl.**filter**.overshoot()
- gl <- gl.**report**.locmetric()

  …… etc



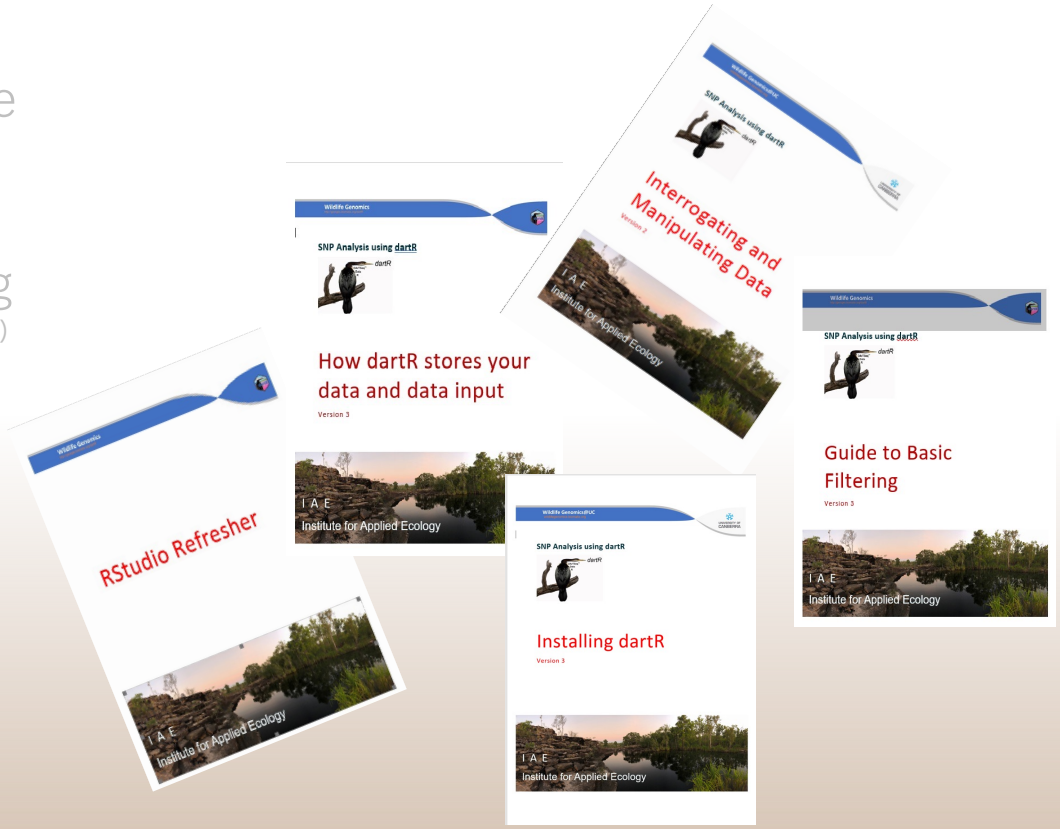
Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

dartR Fundamentals

- Data structure
- Interrogation
- Subsetting
- Basic Filtering
  (refer Renee's sessions)
- Tutorials
  - inline help
  - ?function
  - CRAN documentation

Tutorials



Keep up …

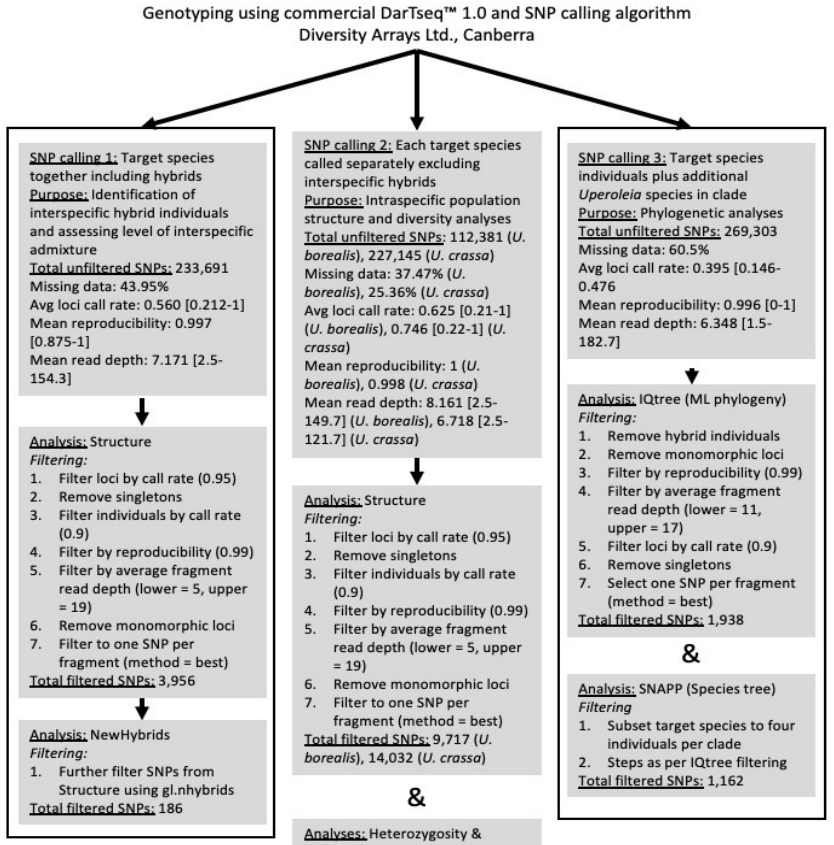
```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

Filtering strategies (or the Art of Filtering)

- One set of SNPs and one set of filters cannot answer all your questions about a single species
- It will give you an answer but there is a good chance it is wrong

This week I will cover:

1. What are the basic filters and considerations? (this session)
2. What are some key considerations for filtering for specific conservation questions? (session 2)
3. What do you need to think about when calling SNPs? (session 3)



Population genomics and sexual signals support reproductive character displacement in *Uperoleia* (Anura: Myobatrachidae) in a contact zone

Frederick R. Jaya, Jessie C. Tanner, Michael R. Whitehead, Paul Doughty, J. Scott Keogh, Craig C. Moritz, Renee A. Catullo

Filtering for call rate (method = “loc”)


Keep up …

What is it?

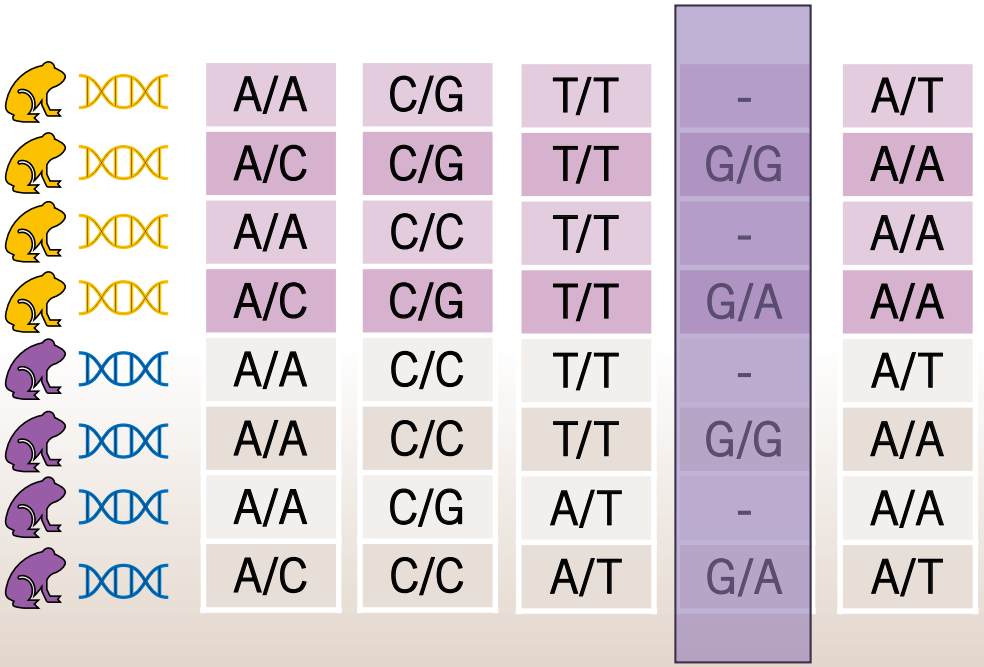
- Removing SNPs because they didn’t work across individuals to a set level

Why is it important?

- Missing loci can add “noise” and computation time to an analysis

Key Considerations:

- Trade-off between strength of filter and number of loci
- Call rate filters matter for different metrics
  - Strong filters disproportionately remove heterozygous sites
  - Many population structure methods work better without high levels of missing data



A/A	C/G	T/T	-	A/T
A/C	C/G	T/T	G/G	A/A
A/A	C/C	T/T	-	A/A
A/C	C/G	T/T	G/A	A/A
A/A	C/C	T/T	-	A/T
A/A	C/C	T/T	G/G	A/A
A/A	C/G	A/T	-	A/A
A/C	C/C	A/T	G/A	A/T

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

Filtering for call rate (method = “ind”)

Keep up …

testset.gl

```
gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]
```

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

```
gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method=“ind”, threshold=0.80)
```

```
gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method=“ind”, threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

What is it?

- Removes *individuals* that did not sequence to the specified level of completeness

Why is it important?

- Deletes the key thing you need to do a study
- Removes individuals that may be misleading as they are sequencing outliers

Key Considerations:

- The filtering step that costs you the very most amount of money
- Generally not a first step – recommend filtering lightly here in the first go and seeing if other filters improve the individuals completeness

A/A	C/G	T/T	G/A	A/T
A/C	C/G	T/T	G/G	A/A
A/A	C/C	T/T	G/A	A/A
A/C	C/G	T/T	G/A	A/A
-	-	T/T	G/G	-
A/A	C/C	T/T	G/G	A/A
A/A	C/G	A/T	G/G	A/A
A/C	C/C	A/T	G/A	A/T

Filtering for reproducibility (gl.filter.reproducibility)

What is it?

- Diversity Arrays duplicates individuals during library prep, to assess whether the same answer is found for every locus
- A control

Why is it important?

- Provides confidence in your base calls which is pretty fundamental

Key Considerations:

- If reproducibility is low, you need your SNPs re-assessed
- If doing sequencing through AGRF or other facilities, make sure to include duplicated individuals (this is an additional cost but do it anyway)

Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

Filtering for read depth (gl.filter.rdepth)

What is it?

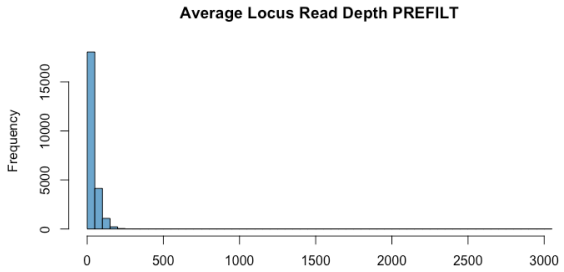
- The mean number of sequencing reads for a particular locus, across all individuals

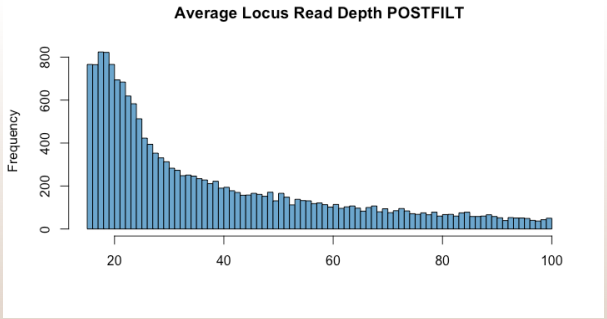
Why is it important?

- Number of reads in a stack tells you how confident you can be in your base calls
- A low read depth means that your heterozygous sites have pretty low coverage

Key Considerations:

- If your read depth is not high enough, there are questions you can't confidently answer (e.g., heterozygosity)
- Very high read depth suggests paralogs (genes with multiple copies) being assembled in to one fragment






Keep up …

testset.gl

```
gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]
```

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

```
gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)
```

```
gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

Filtering out minor alleles (gl.filter.maf)


Keep up …

testset.gl

```
gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]
```

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

```
gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)
```

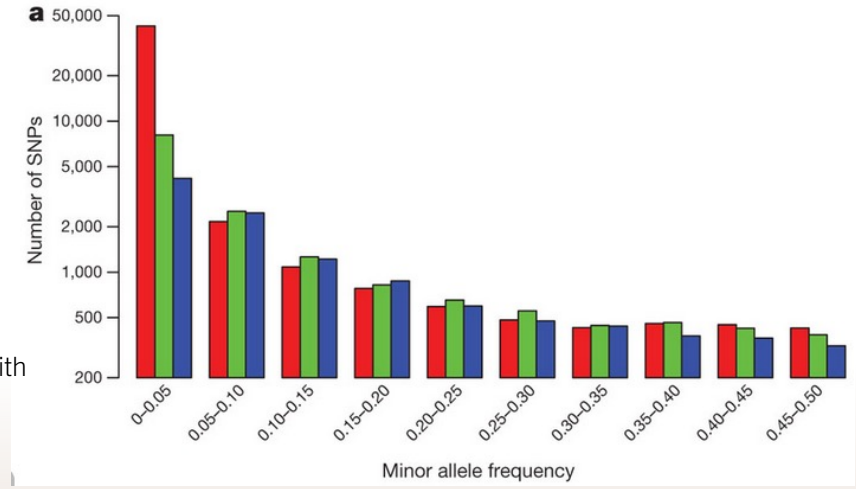
```
gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

What is it?

- Minor allele frequency is the frequency of the second most common allele in a given population
- Removing SNPs based on their relative proportion
- In the literature, often 0.05 BUT…

Key Considerations:

- 0.05 can be high if you have lot of individuals
  - If you have 300 diploid individuals, then you delete SNPs with less than 30 copies of the minor allele
- Or singleton if you have a small number of individuals
- Should depend on your question
  - Rare alleles can be very important to certain questions
  - Most SNPs are rare alleles, as are most heterozygous sites
  - Rare alleles are important in expansion processes
  - NOT important in structure or phylogenetic analyses



https://www.nature.com/articles/nature11174

Filtering secondaries (gl.filter.secondaries)

What is it?

- When there are two SNPs on a single fragment, choosing to keep only one.

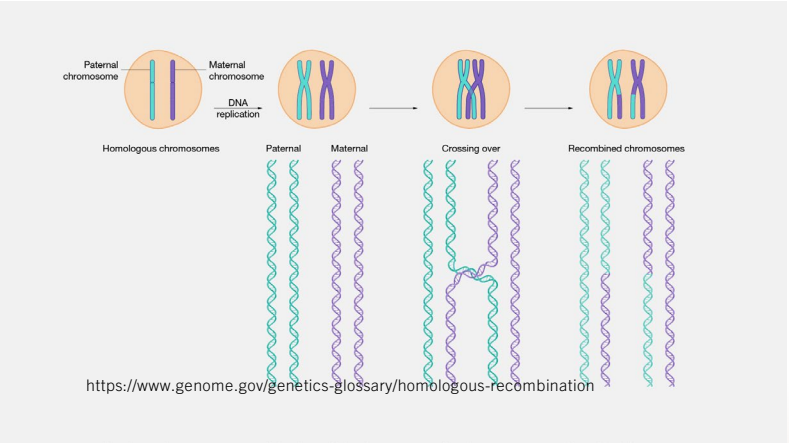
Why is it important?

- Loci that are very close together in the genome are not independently inherited as they are too close together for recombination to split.

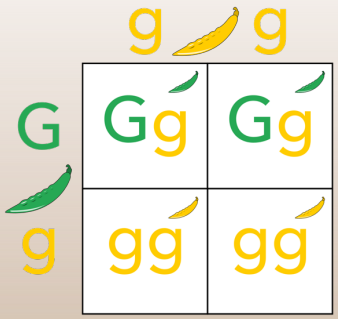
Key Considerations:

- Can mislead structure and phylogenetic analyses
- Can improve representations of heterozygosity

Recombination



Genetic diversity



Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

What's the right filtering order?

- There isn't
- Be iterative – test different options
- Might do the same filter twice (call rate)

As a starting point I would get rid of loci I don't believe in and then individuals that didn't work properly at all:

- gl.filter.callrate(gl, method = "loc", threshold = 0.7)

  Get rid of really poorly sequenced loci

  But don't cut hard
- gl.filter.callrate(gl, method = "ind", threshold = 0.25)

  Very low filter – this is only to get rid of your really bad individuals
- gl.filter.monomorphs(gl)

  Always run this after removing individuals – removes loci that are no longer variable
- gl.filter.reproducibility(gl)

  Get rid of unreliable loci
- gl.filter.rdepth(gl, lower = X, higher = X)

  Get rid of low and super high read depth loci

Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

What's the right filtering order?

- There isn't
- Be iterative – test different options
- Might do the same filter twice (call rate)

Then I would filter more strongly as appropriate for my question. For a population structure analysis I would,

- gl.filter.callrate(gl, method = "loc", threshold = 0.95)

  Structure dislikes missing data

- gl.filter.maf(gl, threshold = 1/(2*nInd(gl))

  I usually set up the threshold so it is just removing singletons to improve computation time

- gl.filter.secondaries(gl)

  Always do this as the last loci filter so that you've cut for quality before you cut because there are two SNPs

- gl.filter.callrate(gl, method = "ind", threshold = .9)

  Filter on individuals. You can usually be a bit flexible at this point.

- gl.filter.monomorphs(gl)

  Always run this after removing individuals


Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

What is right for one situation is probably wrong for a different situation

- Be flexible
- Know what each filter is doing to your data
- Think carefully about whether the filter is appropriate to the test you want to run
- Analyse your data many different ways
- Don't over-interpret your PCoA

Keep up …

```
testset.gl

gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]

gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)

gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)

gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```

Conclusion
Where have we come?

Discussion

Keep up …

testset.gl

```
gl <- testset.gl
nInd(gl)
nLoc(gl)
nPop(gl)
popNames(gl)
indNames(gl)
locNames(gl)
table(pop(gl))
as.matrix(gl)[1:7,1:5]
```

```
gl <- testset.gl
gl.set.verbosity(3)
gl.report.callrate(gl)
gl.report.callrate(gl,method="ind")
gl.report.reproducibility(gl)
```

```
gl.report.callrate(gl,method="ind")
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80)
```

```
gl <- testset.gl
gl.smearplot(gl)
gl <- gl.filter.callrate(gl,verbose=0)
gl <- gl.filter.callrate(gl,
method="ind", threshold=0.80,
verbose=0)
gl.smearplot(gl)
```