Genetic Diversity

Helene Wagner, University of Toronto

Applied Goals:

- Basic check of genetic data (HWE, LD, null alleles)
- Quantify genetic diversity

Technical challenges:

Video 1:
1. Basic checks of pop gen data
2. Review of hypothesis testing

Video 2:
3. Data manipulation in R
4. Rarefaction
5. R comes with no warranty!

Related:
Genetic differentiation (Week 4)
Genetic distance (Week 5)

Source: amphibianrescue.org/category/why-frogs-matter

Basic Checks of Genetic Data

Are markers polymorphic?

Higher variability = more information

	# Alleles	He	
Locus A	1	0	drop!
Locus B	2	0.3	
Locus C	12	0.8	

He = Probability that 2 sampled alleles are different

Hardy-Weinberg equilibrium?

HWE = randomly mating population

P-values	Locus B	Locus C	Locus E
Pop 1	0.81	0.52	0.04
Pop 2	0.01	0.04	0.02
Pop 3	0.19	0.8	0.03

Consistent pattern across locus or population?

Presence of null alleles?

Null alleles = biased allele frequencies

Proportions	Estimate	Lower	Upper
Locus B	0.11	0.00	0.21
Locus C	0.07	0.00	0.13
Locus D	0.21	0.08	0.35

Drop (or redesign) loci with null alleles

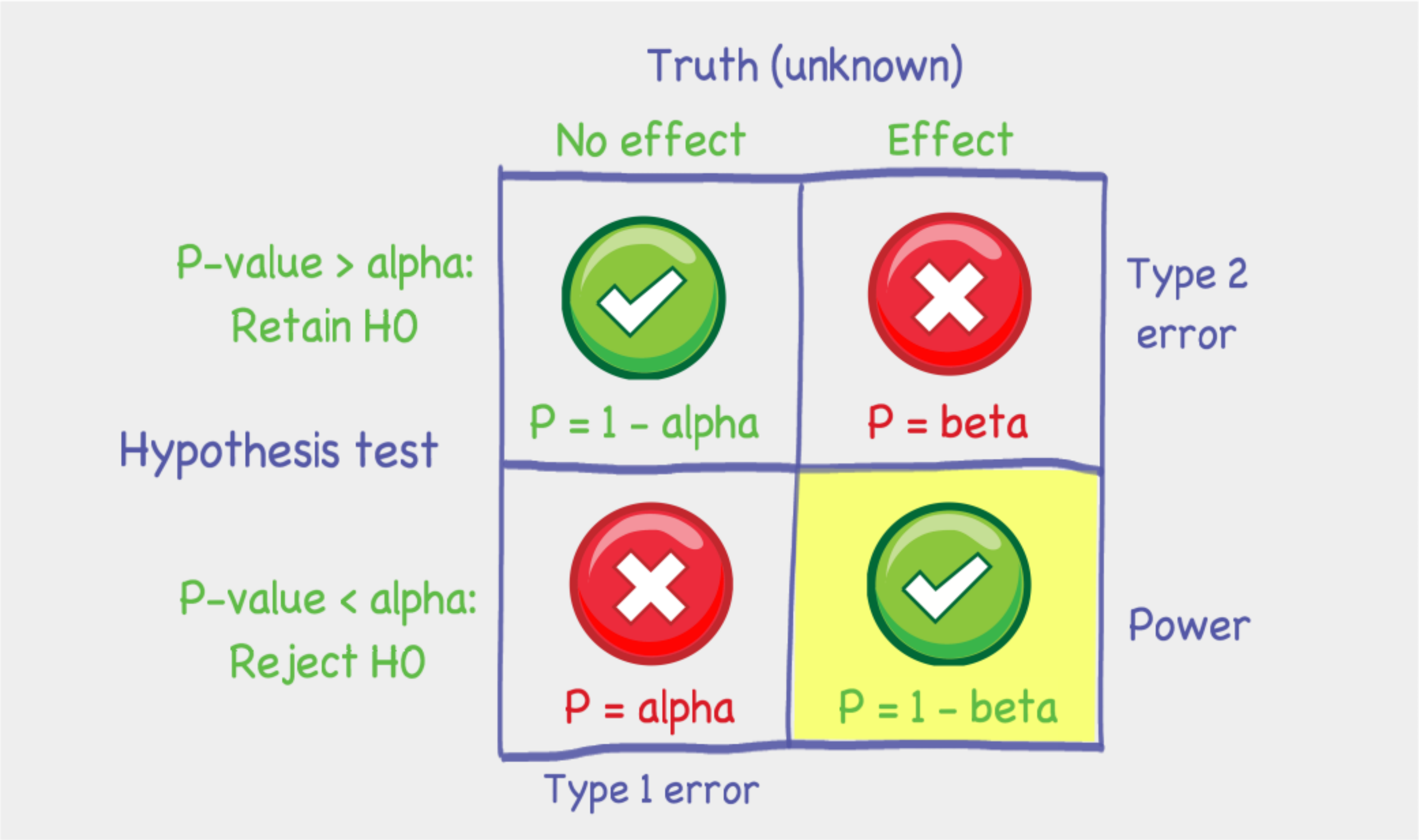
Linkage disequilibrium?

LD = non-independent markers

P-values	Locus B	Locus C	Locus F
Locus C	0.04		
Locus F	0.65	0.31	
Locus G	0.49	0.17	0.13

If two loci are linked across multiple pops, drop one

Statistical Power



Truth (unknown)

Hypothesis test

	No effect	Effect	
P-value > alpha: Retain H0	✓ P = 1 - alpha	✗ P = beta	Type 2 error
P-value < alpha: Reject H0	✗ P = alpha	✓ P = 1 - beta	Power
	Type 1 error		

Hypothesis Testing

Parametric Tests

Hypothesis pair:	HA: Translate biological hypothesis H0: Nothing going on
Test statistic:	Calculated from sample, e.g.: t-statistic, chi-squared, F, z-score
Distribution (H0):	Theoretical distribution (degrees of freedom?)
Conditions:	Theoretical distribution applicable

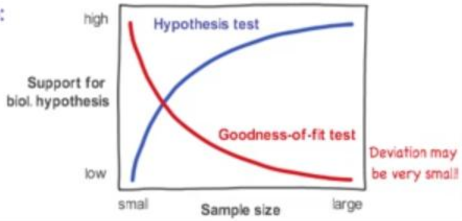
Permutation Tests

Hypothesis pair:	HA: Translate biological hypothesis H0: Nothing going on
Test statistic:	User defined, calculated from sample
Distribution (H0):	Calculated from permuted data: e.g. 499 permutations + obs = 500
Conditions:	Permutation represents H0

Goodness of Fit Tests

Hypothesis pair:	HA: Data don't fit expectation H0: Biological hyp. = no deviation!

Problem:



Deviation may be very small!

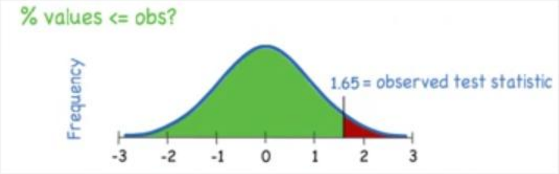
P-value < alpha?

P-value = Percentile of observed test statistic:

pt(obs, df)

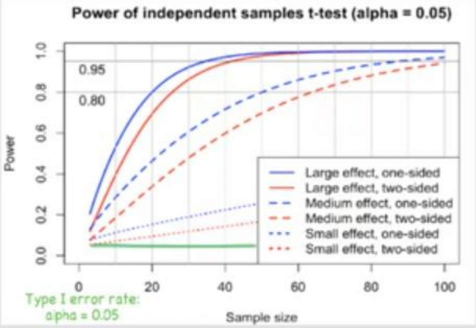
pt(obs, df, lower.tail=FALSE)

% values <= obs?

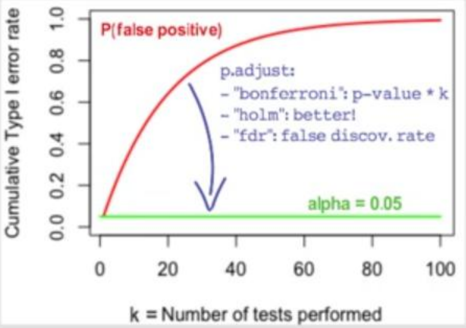


- alternative = "greater": upper tail
- alternative = "two.sided": two-sided
- alternative = "less": lower tail

Statistical Power

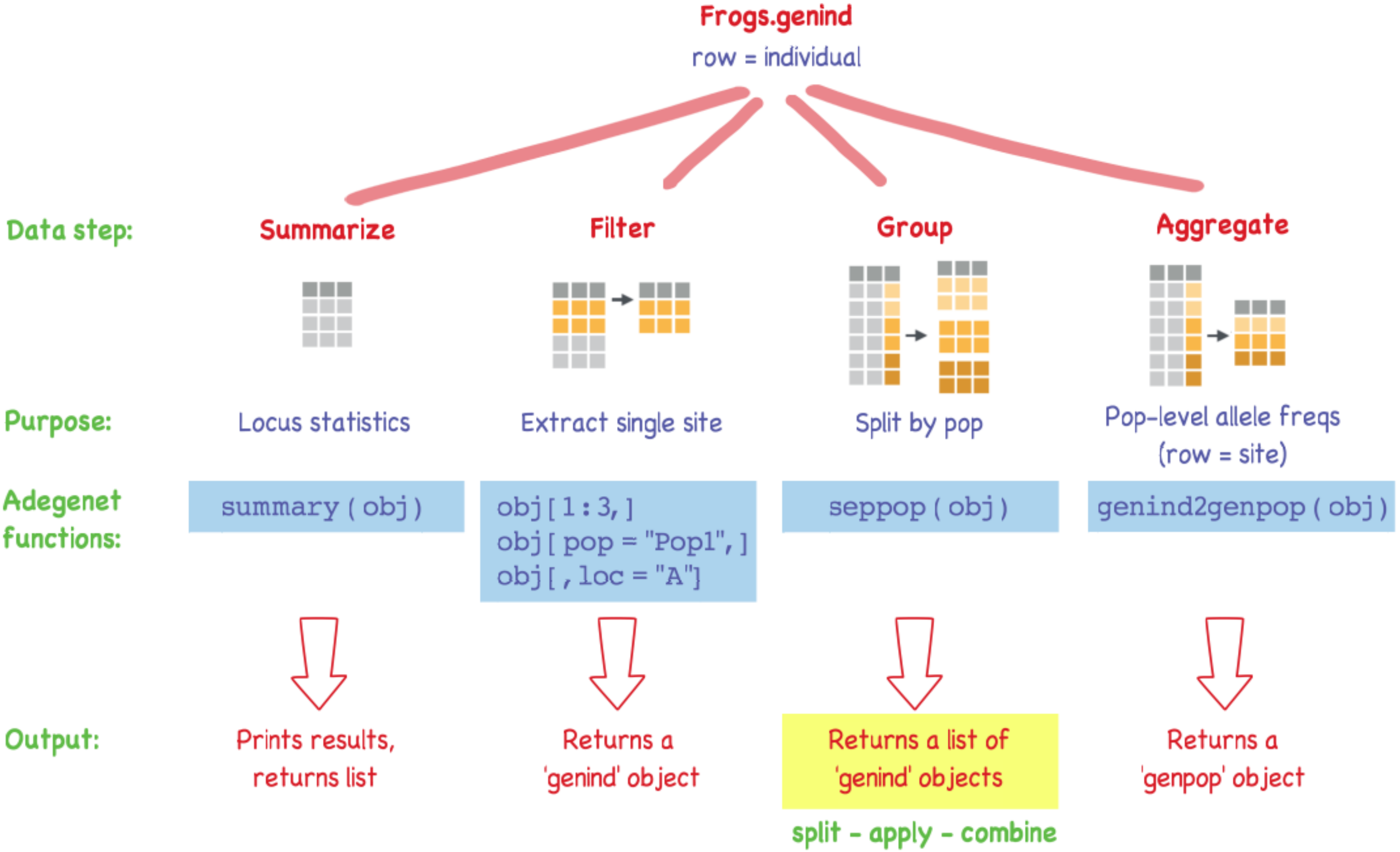


Accounting for Multiple Tests



p.adjust:
- "bonferroni": p-value * k
- "holm": better!
- "fdr": false discov. rate

Aggregating Genetic Data



Frogs.genind

row = individual

Data step:	Summarize	Filter	Group	Aggregate
Purpose:	Locus statistics	Extract single site	Split by pop	Pop-level allele freqs (row = site)

Adegenet functions:	`summary(obj)`	`obj[1:3,]`
		`obj[pop = "Pop1",]`
		`obj[,loc = "A"]`
			`seppop(obj)`	`genind2genpop(obj)`

Output:	Prints results, returns list	Returns a 'genind' object	Returns a list of 'genind' objects	Returns a 'genpop' object
			split - apply - combine	

Your New Best Friend: 'lapply'

Simple form:

```
lapply ( my.list, my.function )
```

```
lapply ( my.list, nrow )
```

General form:

```
lapply ( my.list, function ( ls ) my.function ( ls ) )
```

"Take the list 'my.list' and apply the function 'my.function' to each list element 'ls'."

```
lapply ( my.list, function(x) nrow(x) )
```

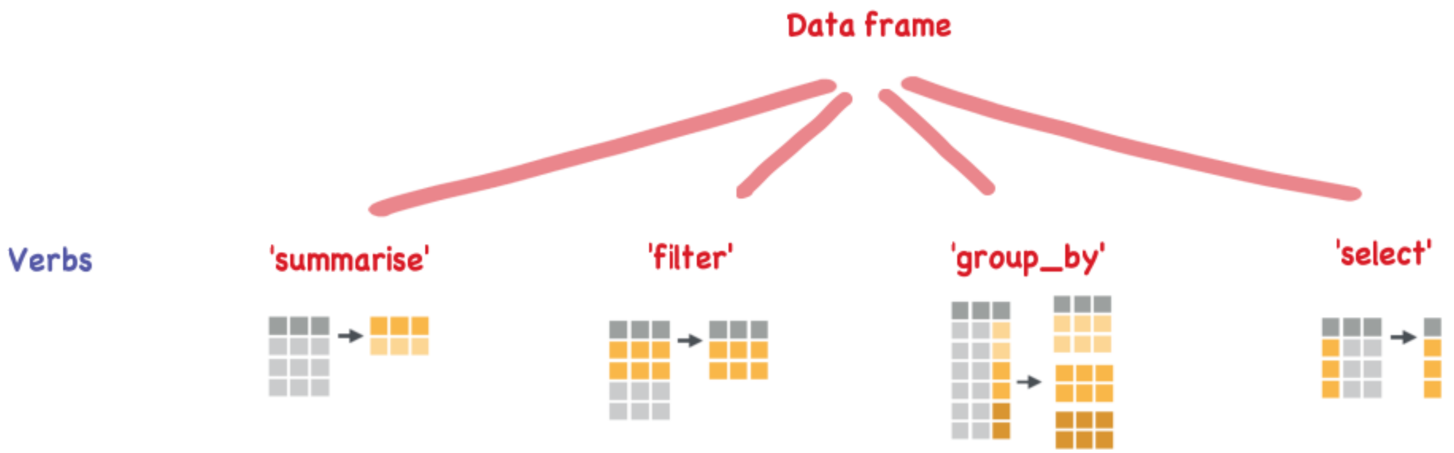
Related:

	takes	returns
'lapply'	List	List
'sapply'	List	Vector or matrix
'mapply'	2 (or more) lists	List (default)
'apply'	Matrix (or array)	Vector (matrix, array)

Example:

```
propTyped( Frogs.genind, by = "loc" )

tmp <- seppop ( Frogs.genind )

lapply ( tmp, function(x)  propTyped(x, by = "loc"))

sapply ( tmp, function(x)  propTyped(x, by = "loc"))

sapply( tmp, propTyped( by = "loc"))
```

R Grammar: Data Manipulation with 'dplyr'



Data frame

Verbs

'summarise'

'filter'

'group_by'

'select'

Examples

```
summarise_all( df, funs( n( ) ))
summarise_all( df, funs( n = n( ), valid = sum ( ! is.na( . ) )) )
filter( df, pop == "Egg")
group_by( df, pop )
select( df, A : H)
```

Logical Operators

<	less than	is.na( )	missing
>	greater than	!is.na( )	not missing
<=	less or equal	&	and
>=	greater or equal	|	or
==	equal	isTrue()	is 'TRUE'
!=	not equal	%in%	is element of

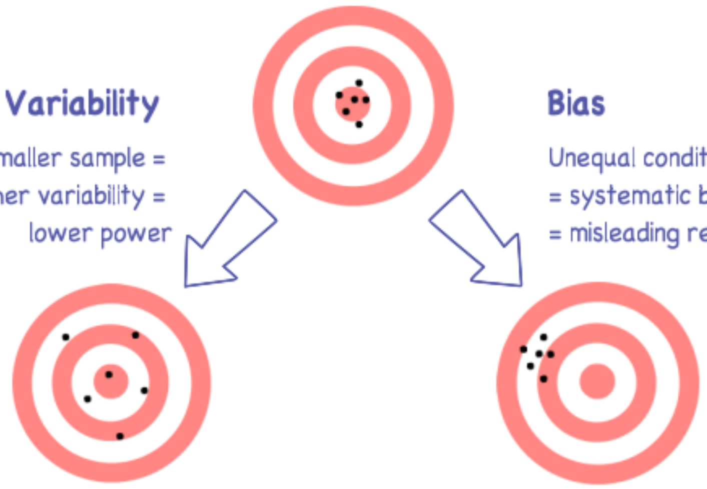
Combine with Pipes: %>% means 'then do'

```
df %>% group_by( pop ) %>% select( A : H ) %>% summarise_all( funs( mean( !is.na( . ) )))
```

Proportion of non-missing values by population and locus:
"Take 'df', then do: group by 'pop', then do: select columns 'A' - 'H', then do: summarize by calculating the proportion of missing values."

Unequal Sample Size?

Bias and variability



Variability

Smaller sample =
higher variability =
lower power

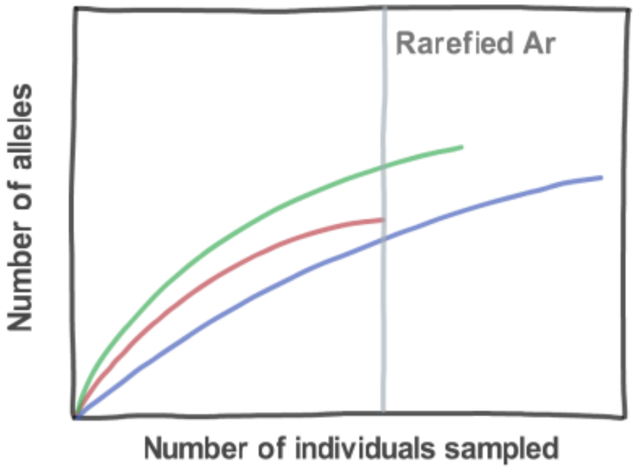
Bias

Unequal conditions
= systematic bias
= misleading results

Allele frequencies:
larger sample =
better estimate

Allelic richness:
larger sample
= more alleles

Rarefaction



In R?

```
PopGenReport :: allel.rich ( genind.obj )
```

R Comes With No Warranty!

Stata
Minitab
R
SPSS
IBM
sas

Different implementations = different results?

```
hierfstat :: fstat ( Frogs.genind )
```

Fst = 0.2004

```
hierfstat :: basic.stats ( Frogs.genind )
```

Fst = 0.1742

AMOVA: 'ade4' != 'pegas' != 'vegan'

R
Base
R
Contributed

What can you do?

1. Read the help file
2. Check user forums
3. Inspect source code

Where to find the source code?

Try this first:

```
fstat
```

Takes S3 objects:

```
methods( mean )
```

```
mean.default
```

Takes S4 objects:

```
showMethods( "seppop" )
```

```
getMethod( "seppop", "ge
```