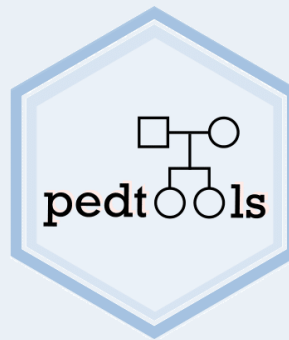
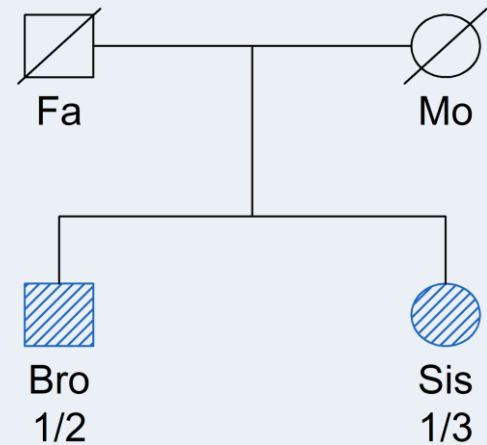




+



=

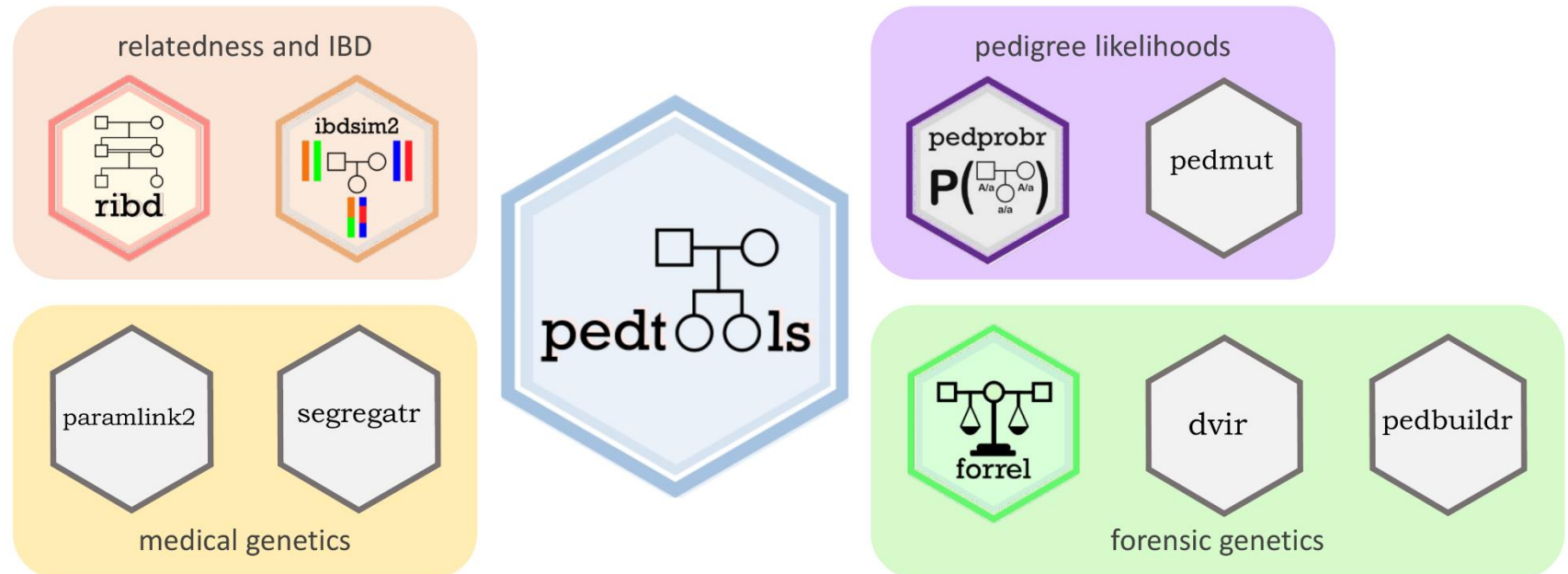


Workshop 1: Pedigrees and relatedness in R

Zakopane - 23.05.2022

Magnus Dehli Vigeland

The **ped suite**: A collection of packages for pedigree analysis in R



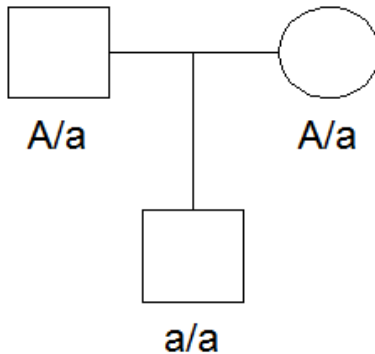
Home page:

<https://magnusdv.github.io/pedsuite>

Source code available on GitHub:

<https://github.com/magnusdv>

pedtools: Tools for working with pedigrees in R



What it contains

pedigrees

markers

What it does

- Create
- Manipulate
- Investigate
- Plot

kinship2

How do I load my
pedigree into R?



Either:

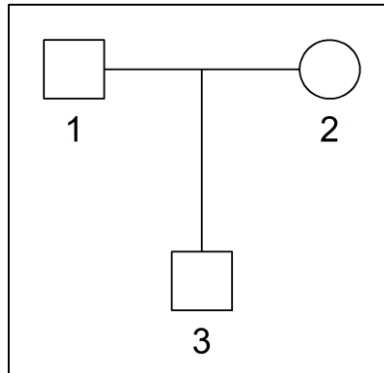
- ✓ Build with **pedtools**
- ✓ Read from file:
 - `readPed()`
 - `readFam()`



Building pedigrees

```
> library(pedtools)

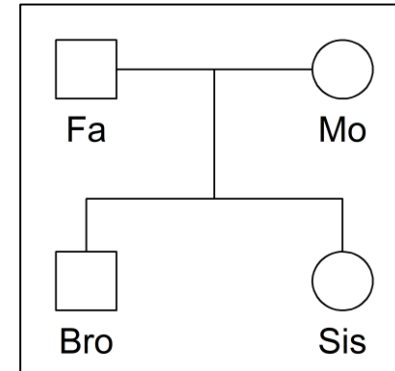
> x = nuclearPed()
> plot(x)
```



Names and sex:

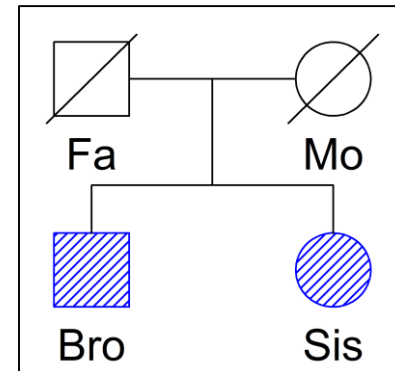
```
> y = nuclearPed(father = "Fa",
                 mother = "Mo",
                 child = c("Bro", "Sis"),
                 sex = 1:2)

> plot(y)
```

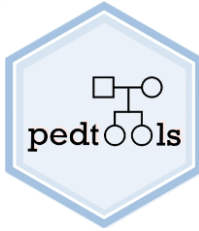


Many ways to tweak the plot!

```
> plot(y,
       deceased = c("Fa", "Mo"),
       hatched = c("Bro", "Sis"),
       col = list(blue = c("Bro", "Sis")),
       cex = 1.5)
```



Some useful functions



Create: basic

- singleton
- nuclearPed
- linearPed
- halfSibPed
- cousinPed
- halfCousinPed

Create: complex

- ancestralPed
- doubleCousins
- quadHalfFirstCousins
- fullSibMating
- randomPed

Manipulate

- addSon
- addDaughter
- addParents
- addChildren
- swapSex
- relabel
- removeIndividuals
- branch
- subset
- mergePed
- breakLoops

Member subsets

- founders
- nonfounders
- leaves
- males
- females
- typedMembers
- untypedMembers

Relatives

- father
- mother
- children
- siblings
- grandparents
- spouses
- ancestors
- descendants
- unrelated

Another example

```
> x = cousinPed(2)
> plot(x)
```

Change gender:

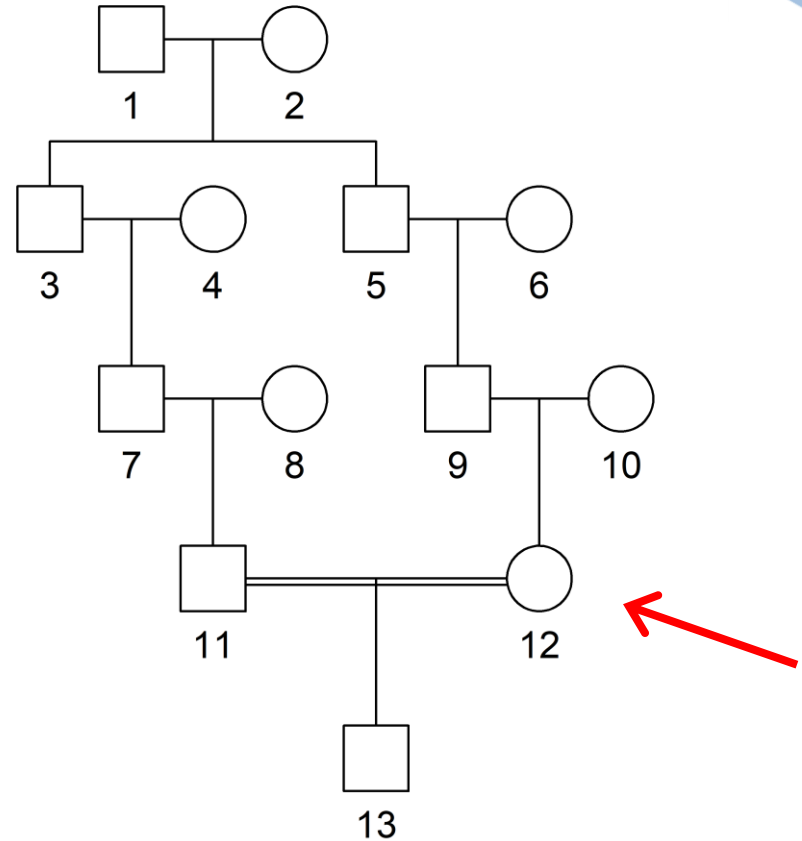
```
> x = swapSex(x, 12)
> plot(x)
```

Add inbred child

```
> x = addChildren(x, nch = 1,
  father = 11, mother = 12)
> plot(x)
```

Remember

- Store the result after each change!
- It is OK to use the same name (if you don't need the previous object)

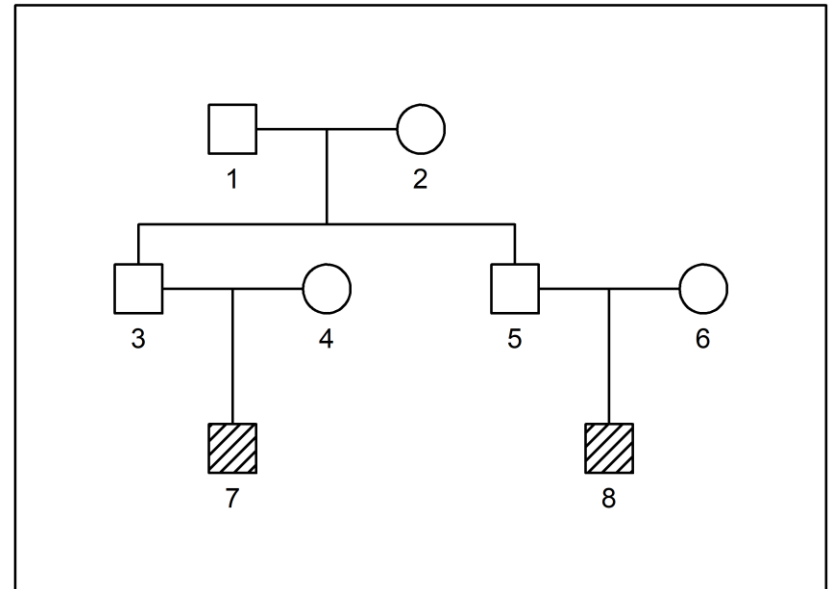
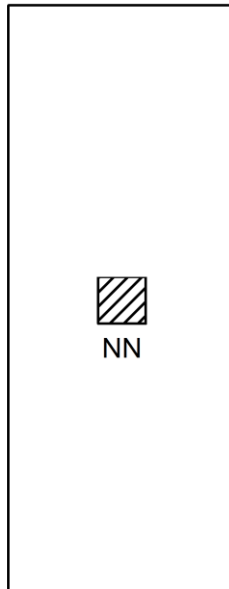
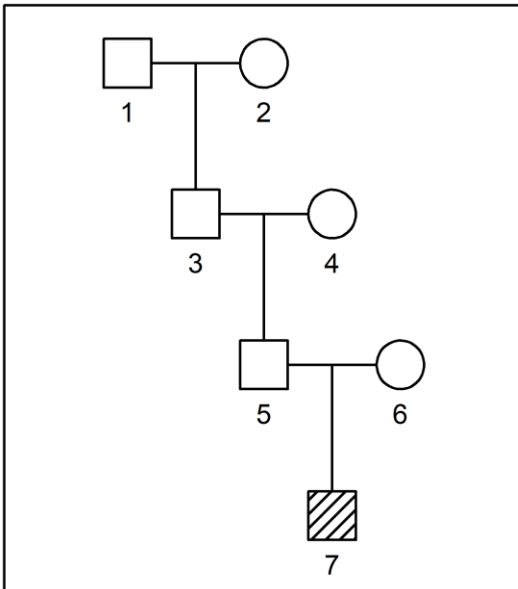


Shortcut command for this pedigree

```
> x = cousinPed(2, child = TRUE)
```

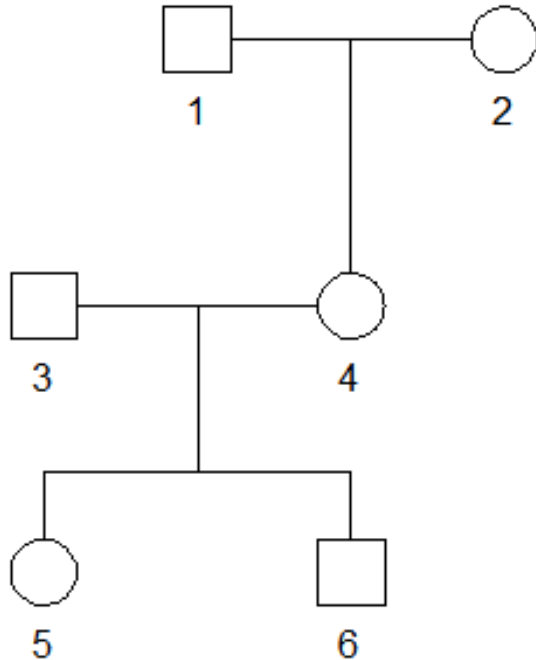
List of pedigrees

```
> peds = list(linearPed(3),  
              singleton("NN"),  
              cousinPed(1))  
  
> plotPedList(peds,  
              widths = c(2, 1, 3),  
              hatched = leaves)
```



Alternative pedigree creation: **ped file**

A text file describing a pedigree structure.



In R:

```
> x = readPed("example.ped")
> plot(x)
```

famid	id	fid	mid	sex
1	1	0	0	1
1	2	0	0	2
1	3	0	0	1
1	4	1	2	2
1	5	3	4	2
1	6	3	4	1

Contents of *example.ped*

Columns

famid = family ID (optional)

id = individual ID

fid = ID of father

mid = ID of mother

sex = 1 (male), 2 (female) or 0 (unknown)

Oh my! Do I have
to write these ped-
files by hand?



No, that is tedious
and error-prone!


Free online tool:
QuickPed¹




¹<https://magnusdv.shinyapps.io/quickped>

Quick demo of **QuickPed**

[*https://magnusdv.shinyapps.io/quickped*](https://magnusdv.shinyapps.io/quickped)



Ok, I've got my
pedigree in R now.
But what about
marker data?

- 
- ✓ Create with **pedtools**
 - ✓ Load from
 - ped file
 - fam file

Adding marker data

```
> x = nuclearPed(2)
> x = addMarker(x, geno = c(NA, NA, "a/b", "a/a"))
> plot(x, marker = 1)
```

Print to screen:

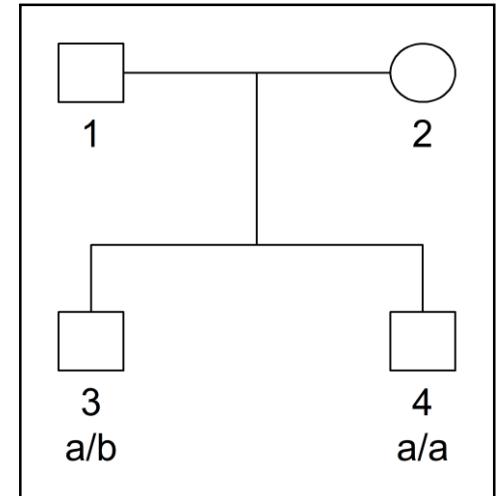
```
> x
  id fid mid sex <1>
  1   *   *   1 -/-
  2   *   *   2 -/-
  3   1   2   1 a/b
  4   1   2   1 a/a
```

Change allele frequencies:

```
> x = setAfreq(x, marker = 1,
               c(a = 0.3, b = 0.7))
```

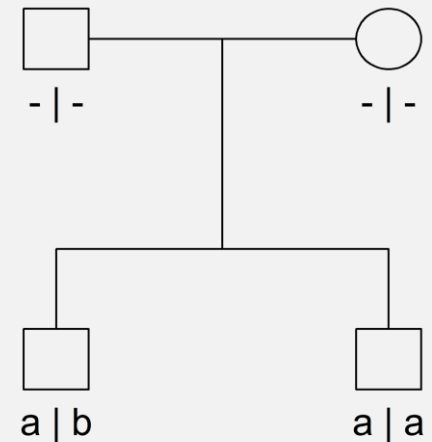
Inspect the result

```
> afreq(x, marker = 1)
  a    b
0.3 0.7
```



Many plot options:

```
> plot(x,
      marker = 1,
      labs = NULL,
      sep = " | ",
      showEmpty = T)
```



Functions for manipulating marker data

Get attributes

- alleles
- afreq
- mutmod
- genotype
- chrom
- posMb
- name

Set attributes

- setAfreq
- setMutationModel
- setGenotype
- setChrom
- setPosition
- setMarkername

Attach/remove

- addMarker
- setMarkers
- selectMarkers
- removeMarkers
- transferMarkers

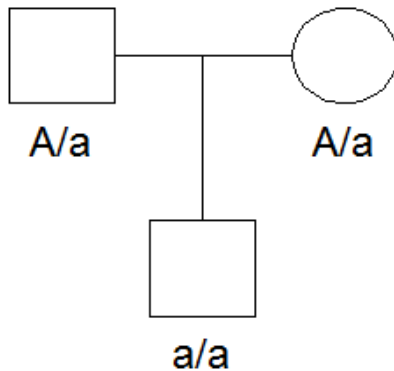
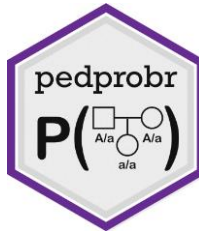
Marker map

- getMap
- setMap

Frequency database

- getFreqDatabase
- setFreqDatabase
- readFreqDatabase
- writeFreqDatase

pedprobr: Pedigree probabilities in R



What it does

Compute the probability

$$\mathcal{P}(\text{genotypes} \mid \text{pedigree}; \text{params})$$

Features

- arbitrary inbreeding
- autosomal & X-linked
- linked markers
- mutation models
- Elston-Stewart algorithm

A simple likelihood

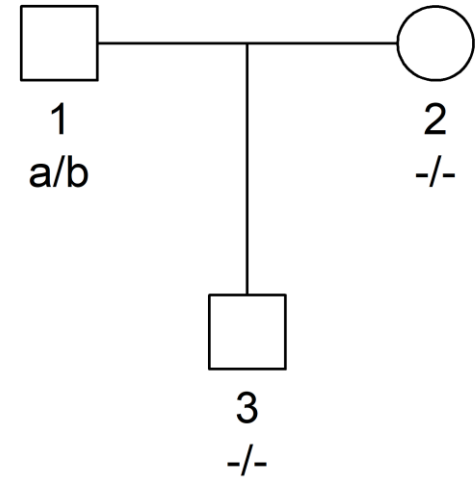
```
> library(pedprobr) # or library(pedsuite)
```

Create a pedigree with a SNP marker

```
> x = nuclearPed(1)
> x = addMarker(x, "1" = "a/b")
> plot(x, marker = 1)
```

Compute the pedigree likelihood

```
> likelihood(x, marker = 1)
[1] 0.5
```

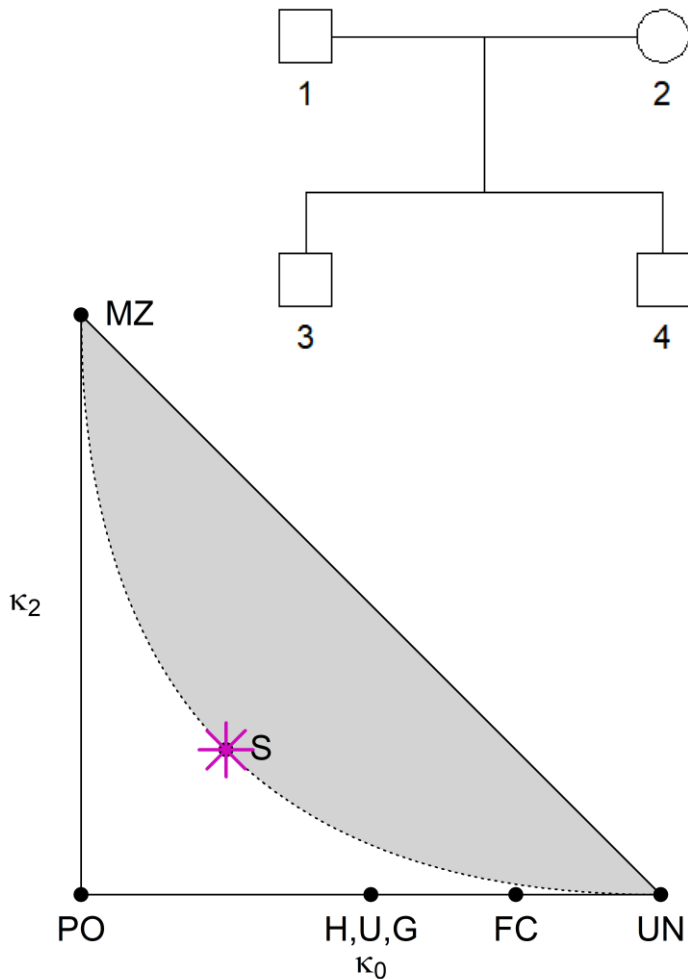
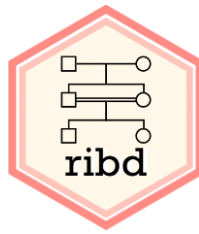


Manual calculation

- $P(a) = P(b) = 0.5$ (by default)
- HWE implies:

$$P(a/b) = 2pq = 2 \cdot 0.5 \cdot 0.5 = 0.5$$

ribd: Pedigree-based relatedness coefficients



```
> library(ribd)
> x = nuclearPed(2)

> kinship(x, ids = 3:4)
[1] 0.25

> kappaIBD(x)
id1 id2 kappa0 kappa1 kappa2
1   2   1.00    0.0    0.00
1   3   0.00    1.0    0.00
1   4   0.00    1.0    0.00
2   3   0.00    1.0    0.00
2   4   0.00    1.0    0.00
3   4   0.25    0.5    0.25

> k = kappaIBD(x, ids = 3:4)
> showInTriangle(k, pch = 8, cex = 2)
```

Now: Exercises!

