



Lecture 5:

Inference of pairwise relatedness and Pedigree reconstruction

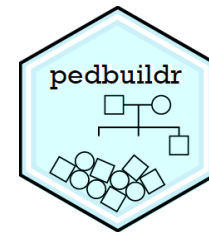
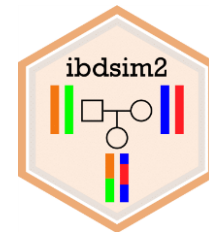
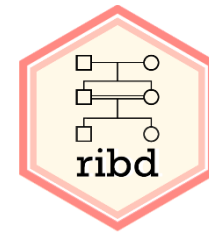
Pedigree analysis in R

ISFG Summer School - Virtual Edition 2021

Magnus Dehli Vigeland

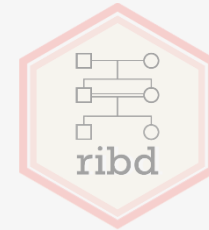
Plan

- Part 1: Measures of relatedness
- Part 2: Realised relatedness
- Part 3: Inference of pairwise relatedness
- Part 4: Pedigree reconstruction

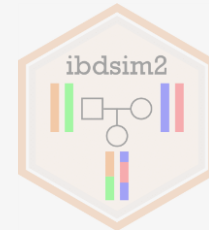


Plan: Today

- Part 1: Measures of relatedness



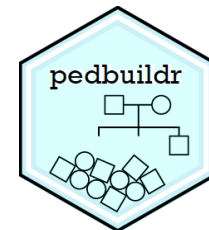
- Part 2: Realised relatedness



- Part 3: Inference of pairwise relatedness

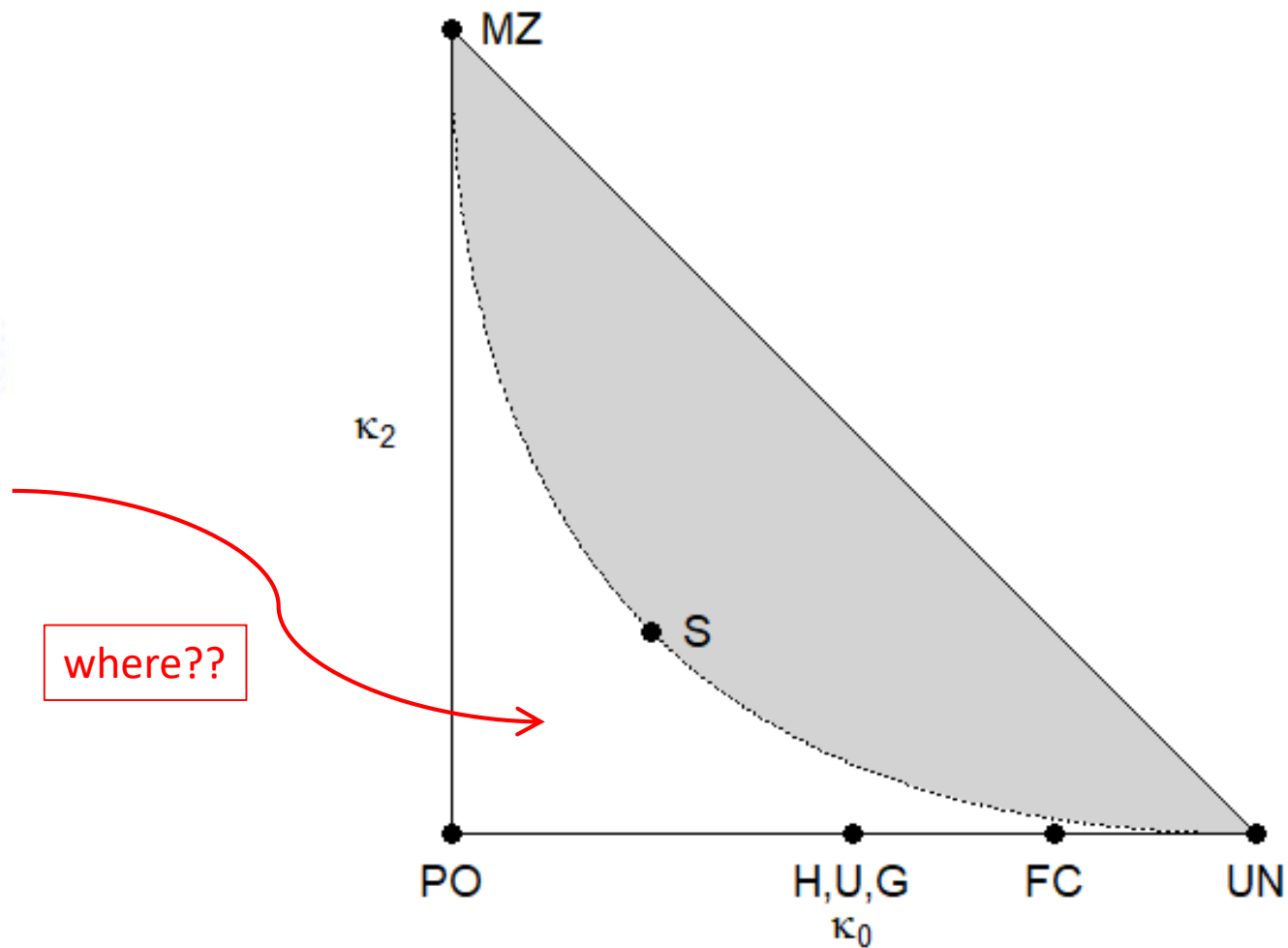


- Part 4: Pedigree reconstruction





Part 3: Inference of pairwise relatedness

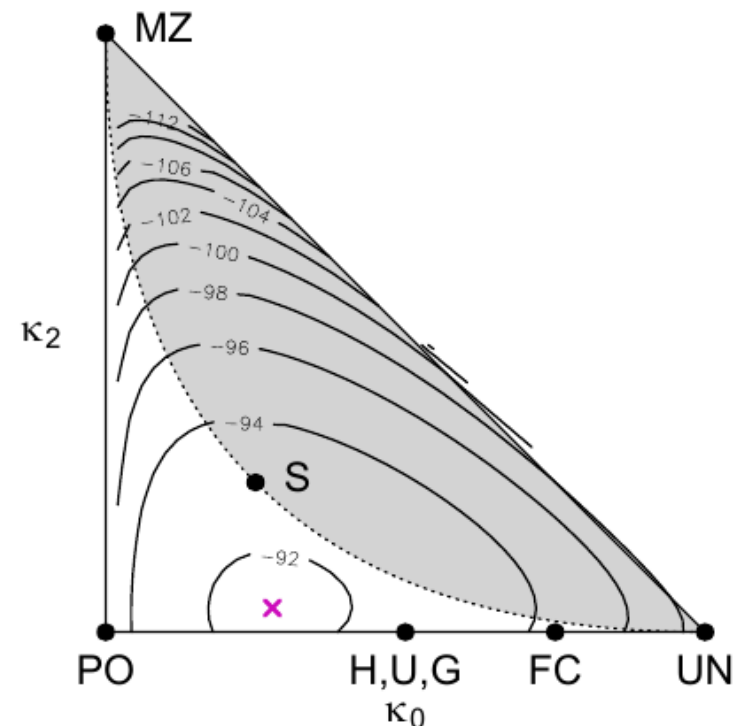


Maximum likelihood estimation of $\kappa = (\kappa_0, \kappa_1, \kappa_2)$

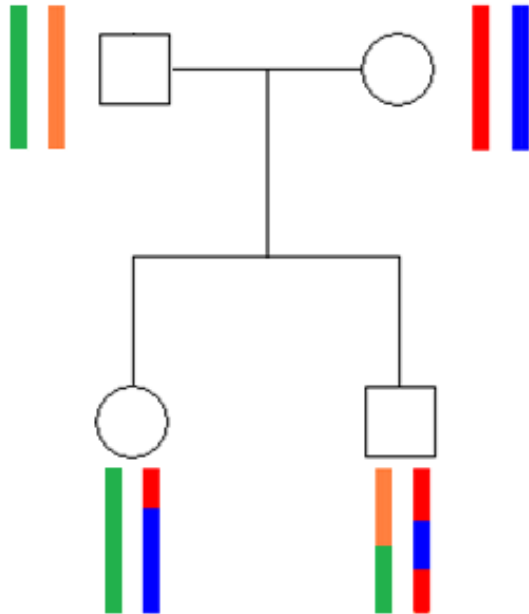
- Thompson (1975)
 - Given: marker genotypes for two individuals
 - The likelihood function

$$L(\kappa) = P(\text{genotypes} \mid \kappa)$$

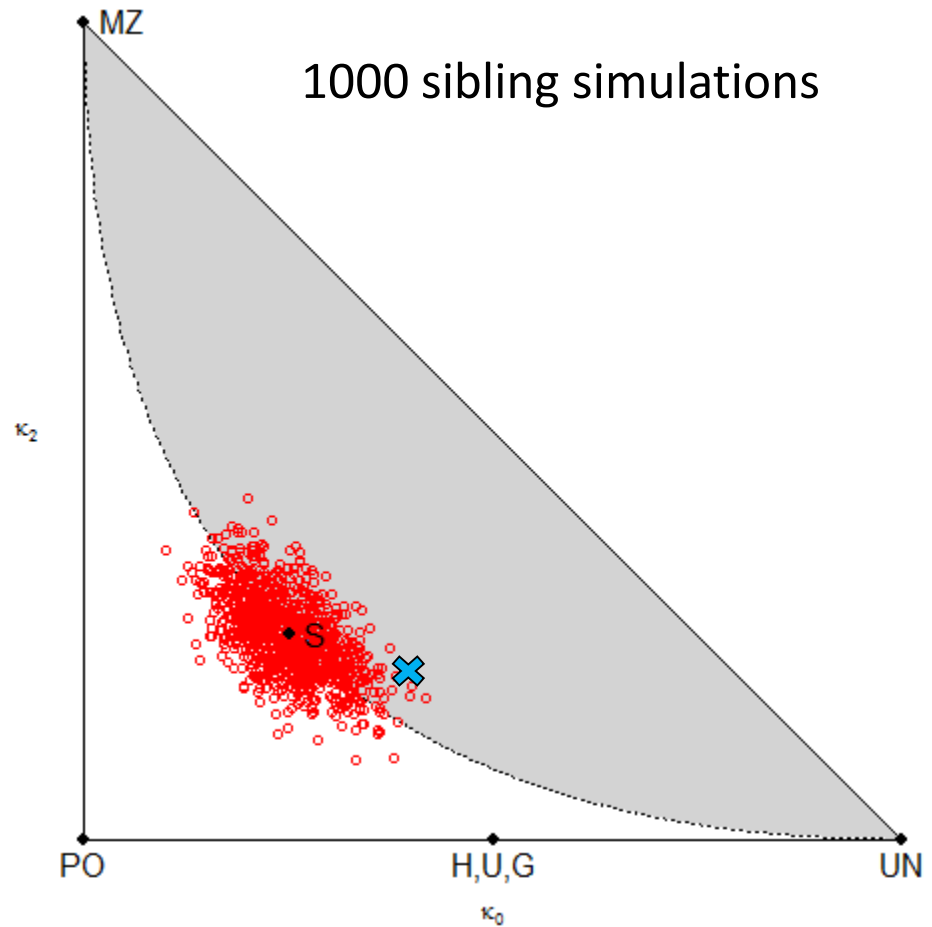
- Find the point k which maximizes L !
 - Called the maximum likelihood estimate (MLE)
- Assumptions:
 - known allele freqs
 - HWE
 - no inbreeding



What are we estimating?



Answer: The *realised* coefficients!



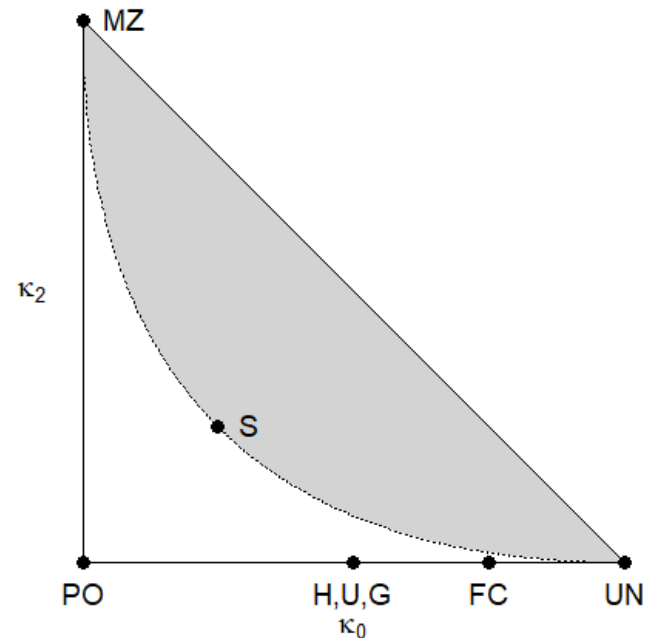
Implementations

- R

- pedsuite (forrel)
- SNPrelate, GWASTools (optimized for association studies)
- CrypticIBDcheck (as above, slow with many markers)
- +++

- Other

- PLINK
- KING
- Beagle
- +++



Pairwise inference with forrel



- Key functions

```
> ibdEstimate()           # estimate kappa
> showInTriangle()        # visualize!
> ibdBootstrap()          # bootstrap confidence
> checkPairwise()         # detect pedigree errors
```

- Simulation

```
> markerSim()             # iid markers
> profileSim()            # complete profiles
```

(Both of these support conditioning on known genotypes)

Pairwise inference with forrel: Example



Simulate 100 SNPs for a pair of siblings

```
> library(forrel)

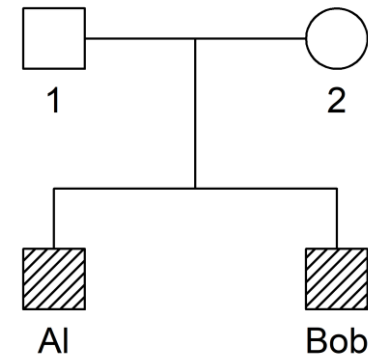
> ids = c("Al", "Bob")
> x = nuclearPed(children = ids)

> x = markerSim(x, N = 100, ids = ids,
               alleles = 1:2, seed = 1234)

> x
  id fid mid sex <1> <2> <3> <4> <5>
1   *   *   1  -/- -/- -/- -/- -/-
2   *   *   2  -/- -/- -/- -/- -/-
Abe  1   2   1  1/1 1/2 1/1 1/2 2/2
Bob  1   2   1  1/1 1/2 1/1 1/2 2/2
```

Only 5 (out of 100) markers are shown.

```
> dat = list(subset(x, "Al"),
             subset(x, "Bob"))
```

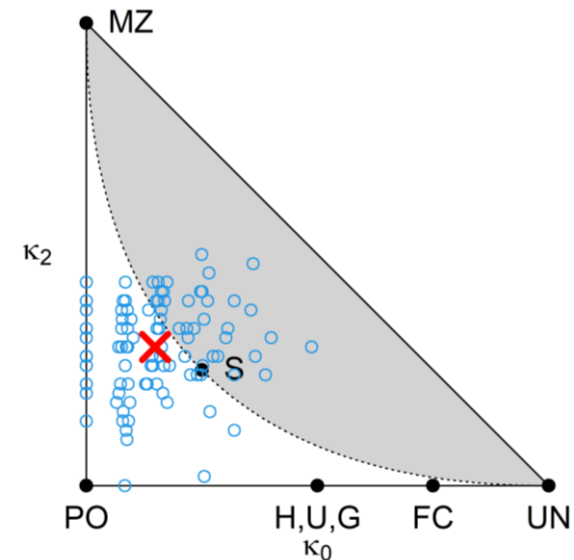
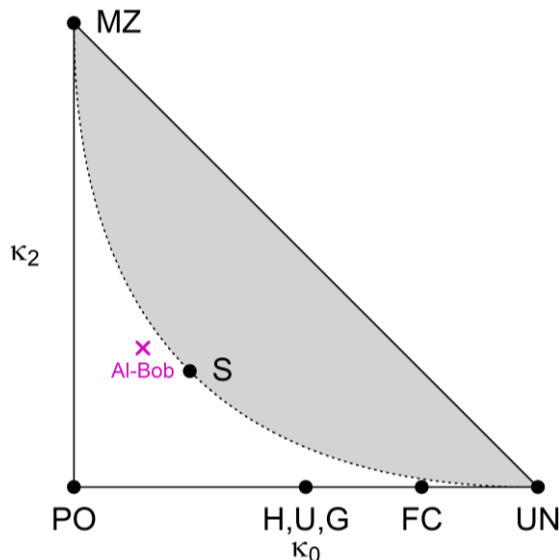
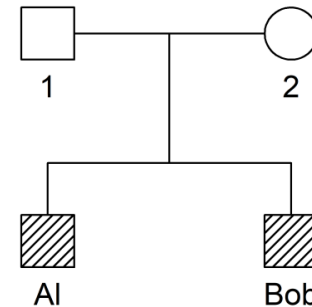


Pairwise inference with forrel: Example



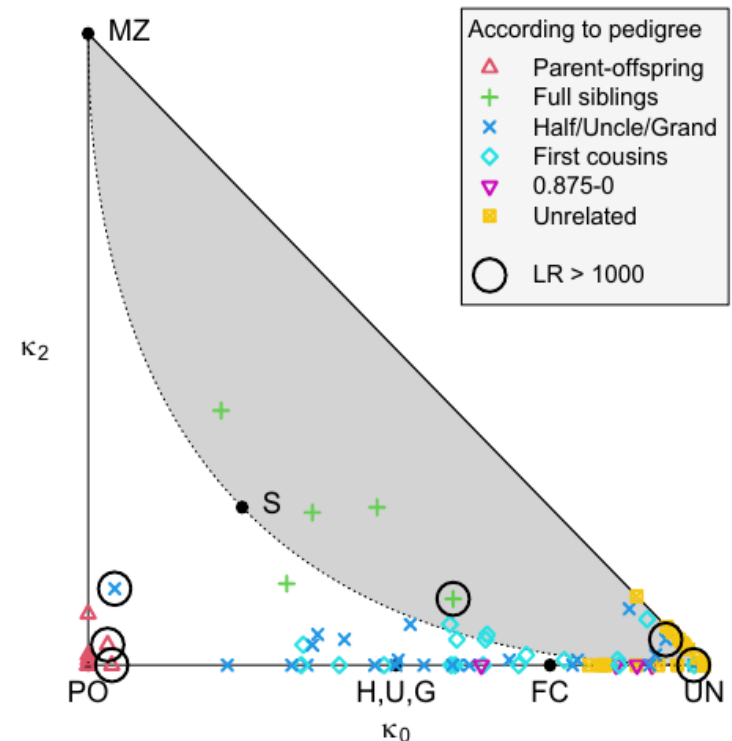
Estimate kappa from the data

```
> k = ibdEstimate(dat)
> k
  id1 id2   N    k0    k1    k2
1  Al Bob 100 0.1486 0.55139 0.30002
> showInTriangle(k, labels = T)
> bs = ibdBootstrap(dat, ids, N = 100,
  param = "kappa")
```



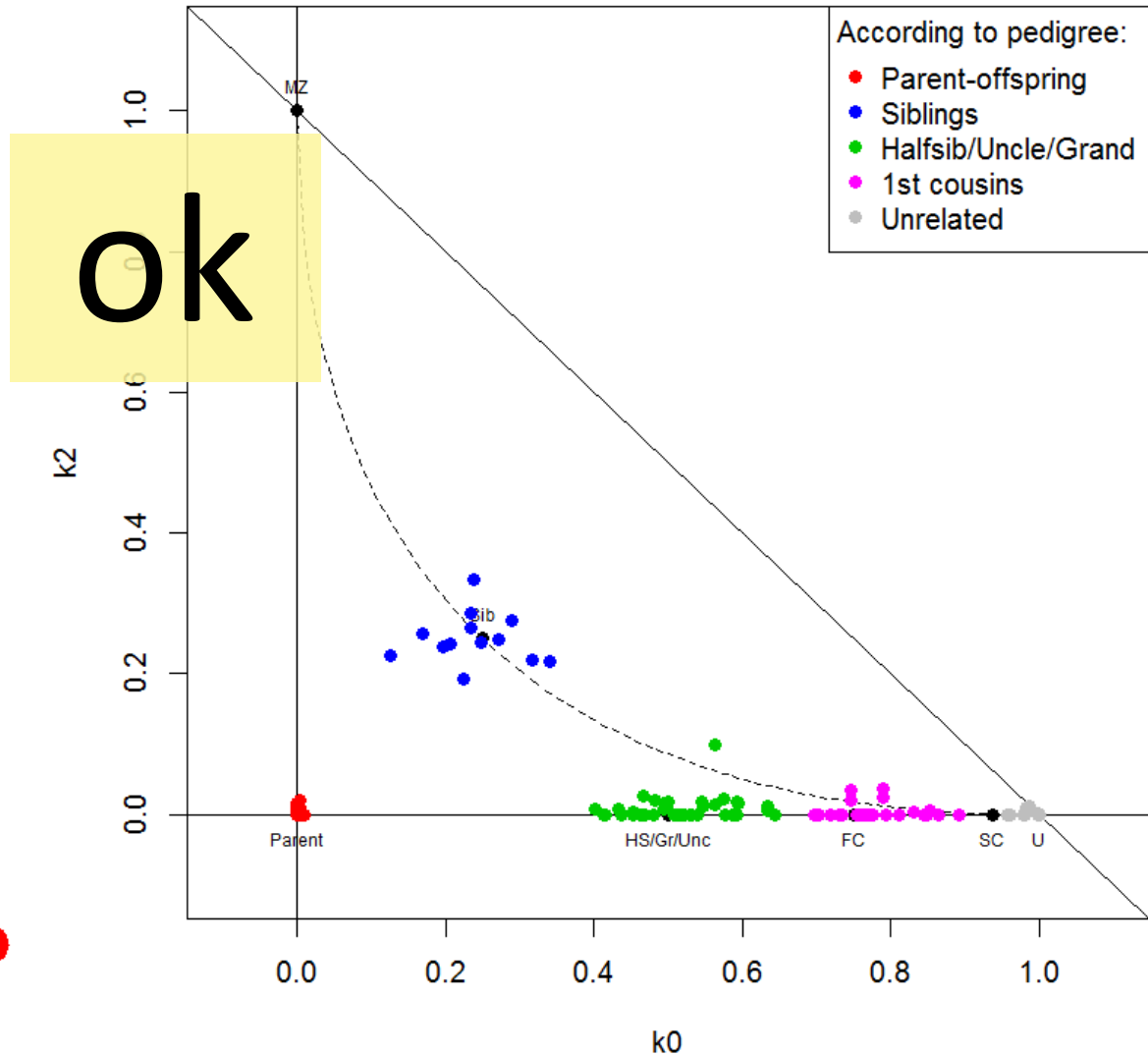
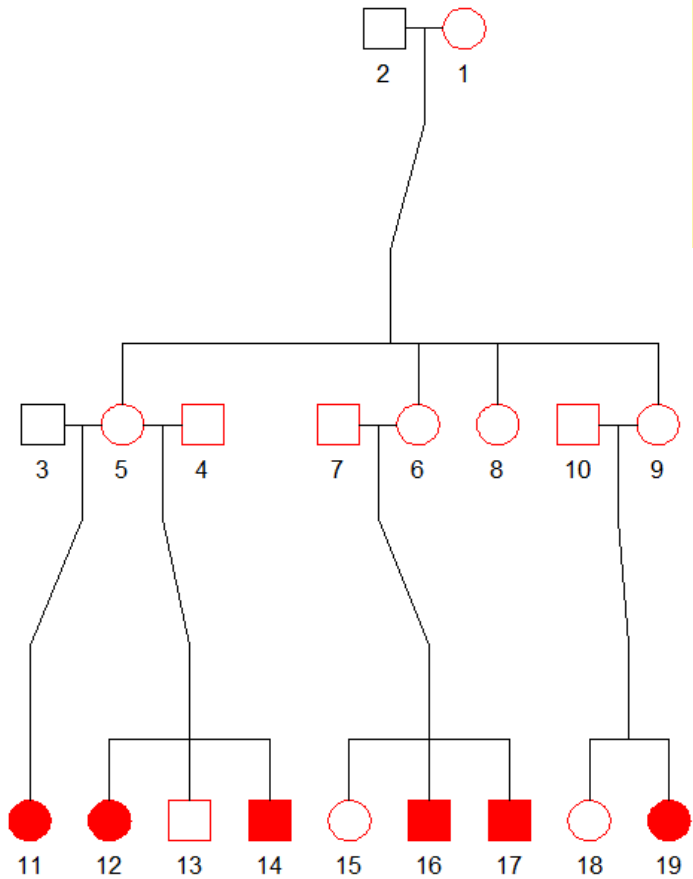
Application: Detecting pedigree errors

- Suppose \mathbf{x} is a pedigree object with attached markers
- The function `checkPairwise(\mathbf{x})` computes:
 - pedigree-based kappa for all pairs: `kappaIBD(\mathbf{x})`
 - marker-based kappa estimates for all pairs: `ibdEstimate(\mathbf{x})`
 - LR comparing the two
 - Color-coded plot according to relationship claimed by pedigree



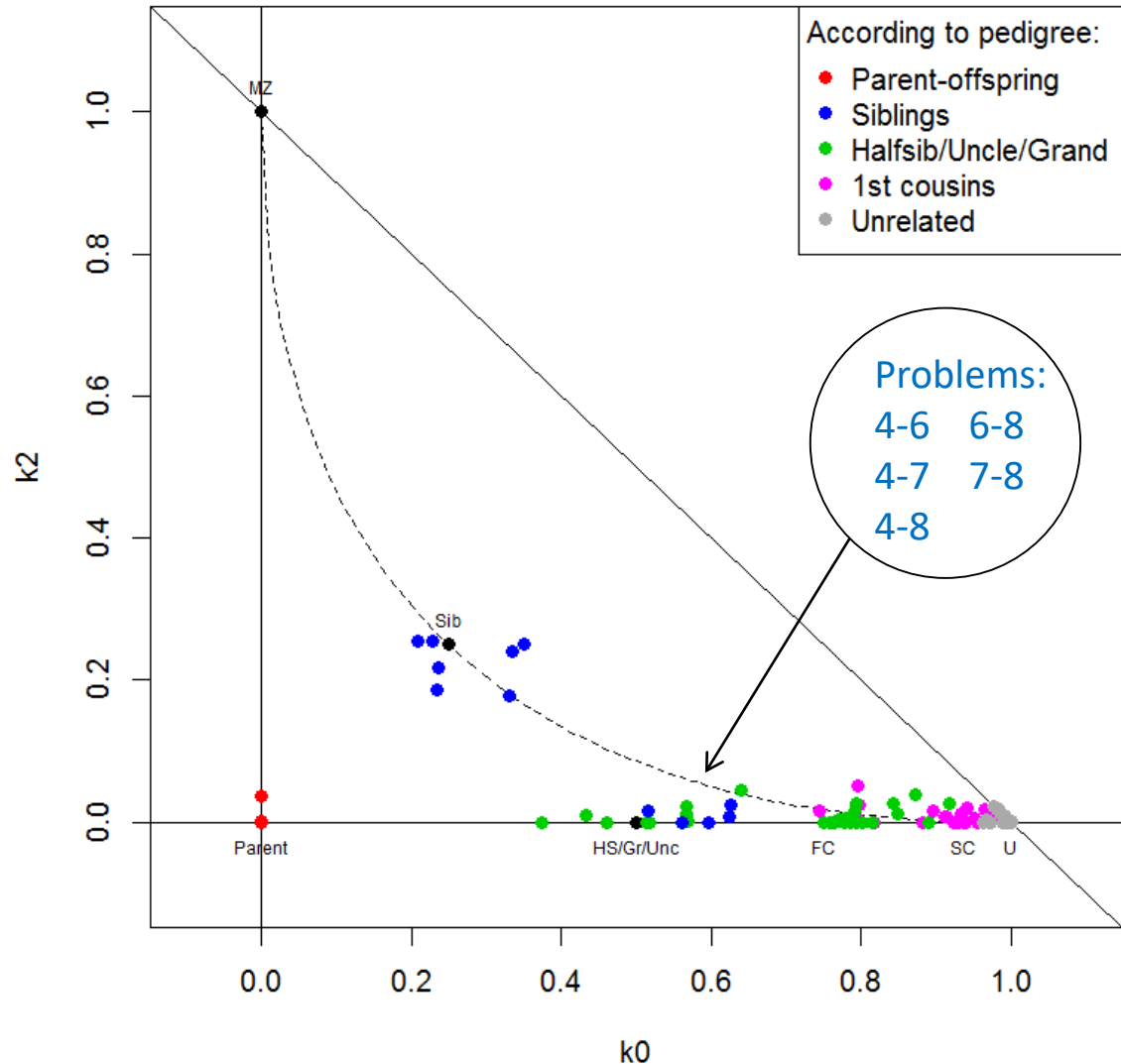
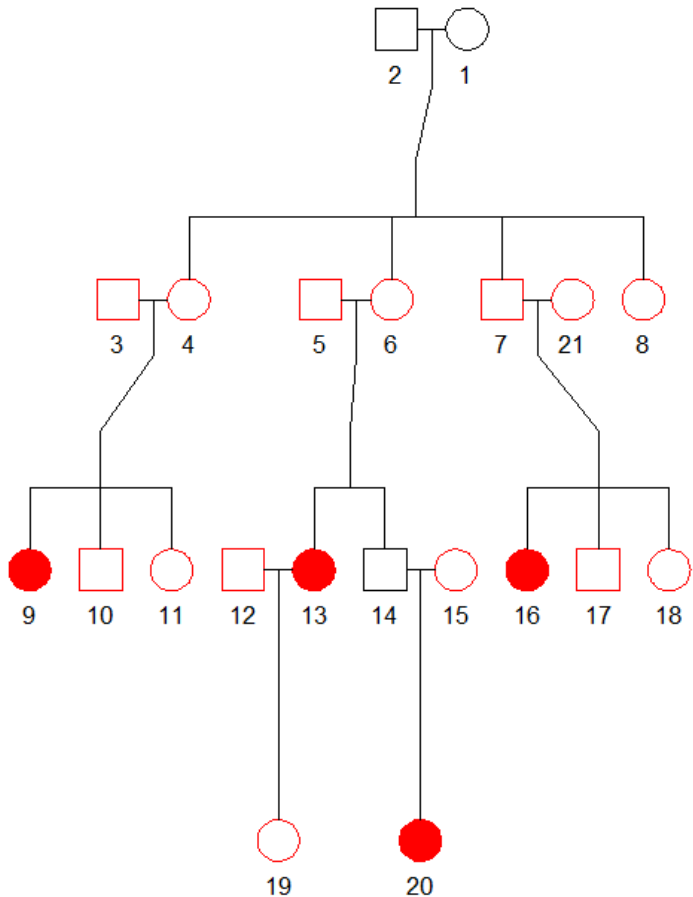
checkPedigree: Example 1

Family 22



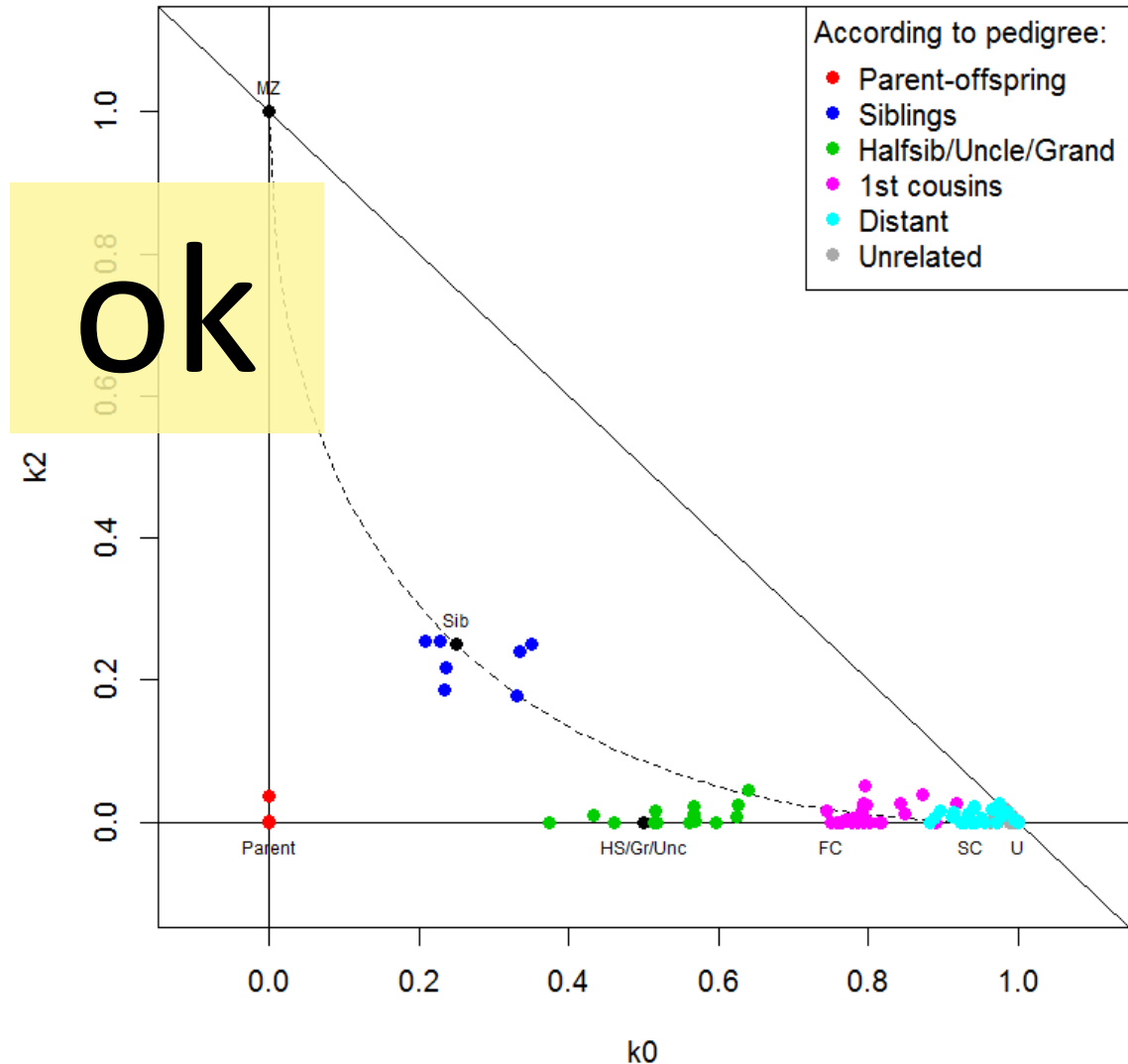
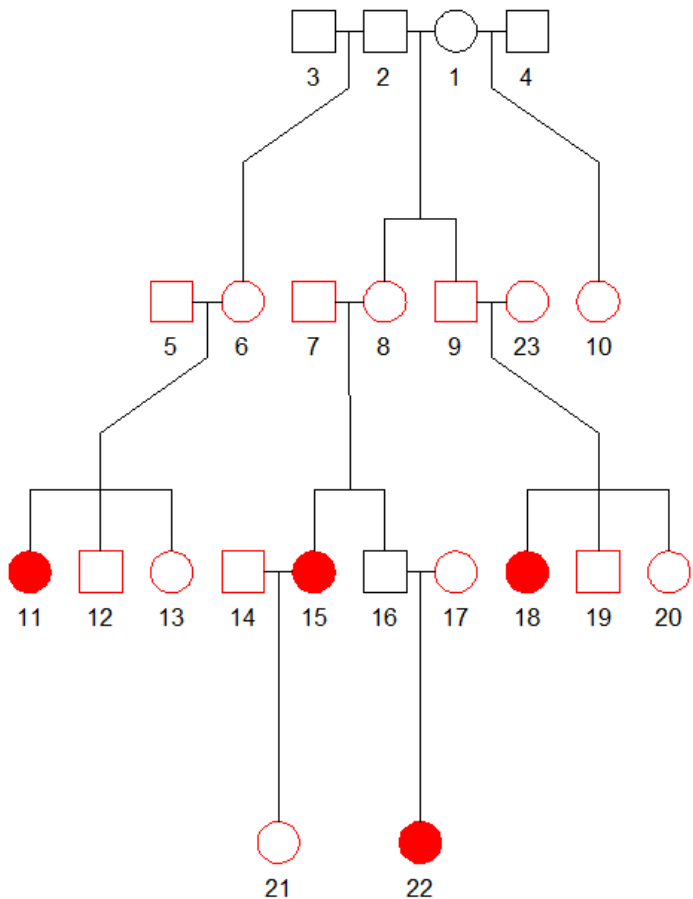
checkPedigree: Example 2

Family 16



checkPedigree: Example 2 - corrected

Family 16



Relatedness inference vs. allele frequencies

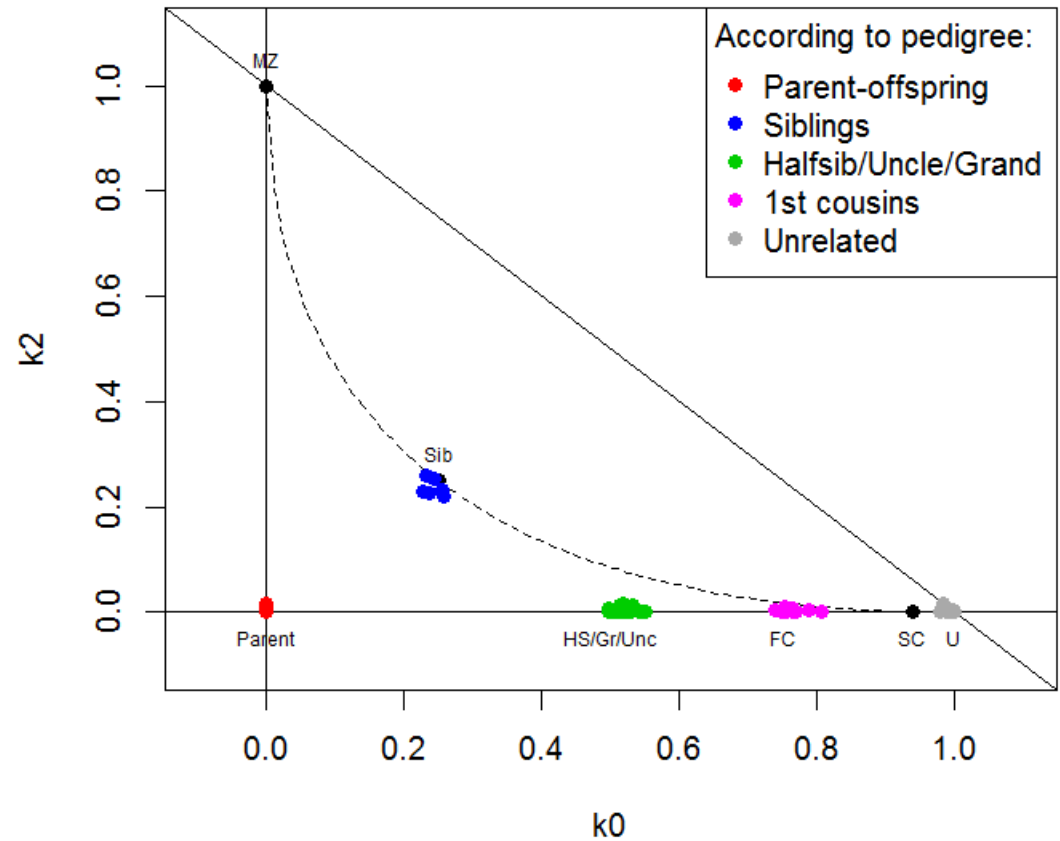
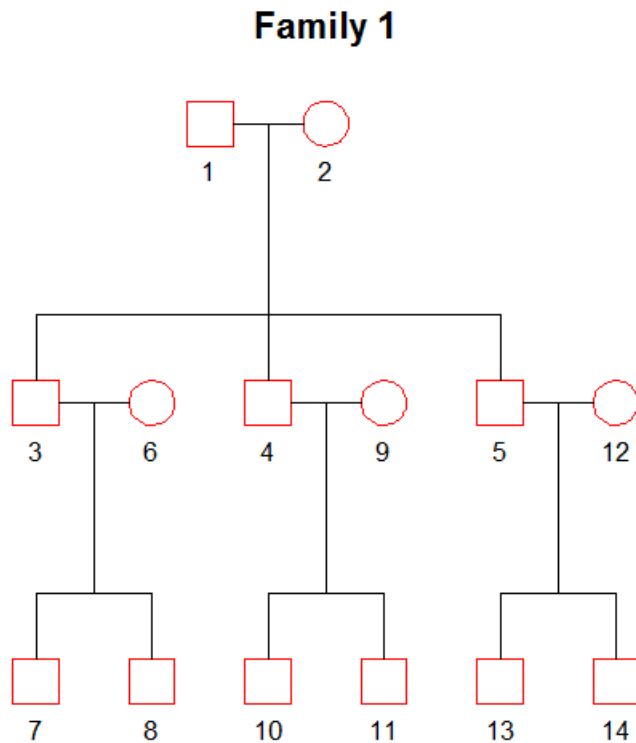
- A little simulation experiment!

Simulation example

SNPs: 10 000

True frequency distr: Unif(0,1)

Frequencies used: Correct

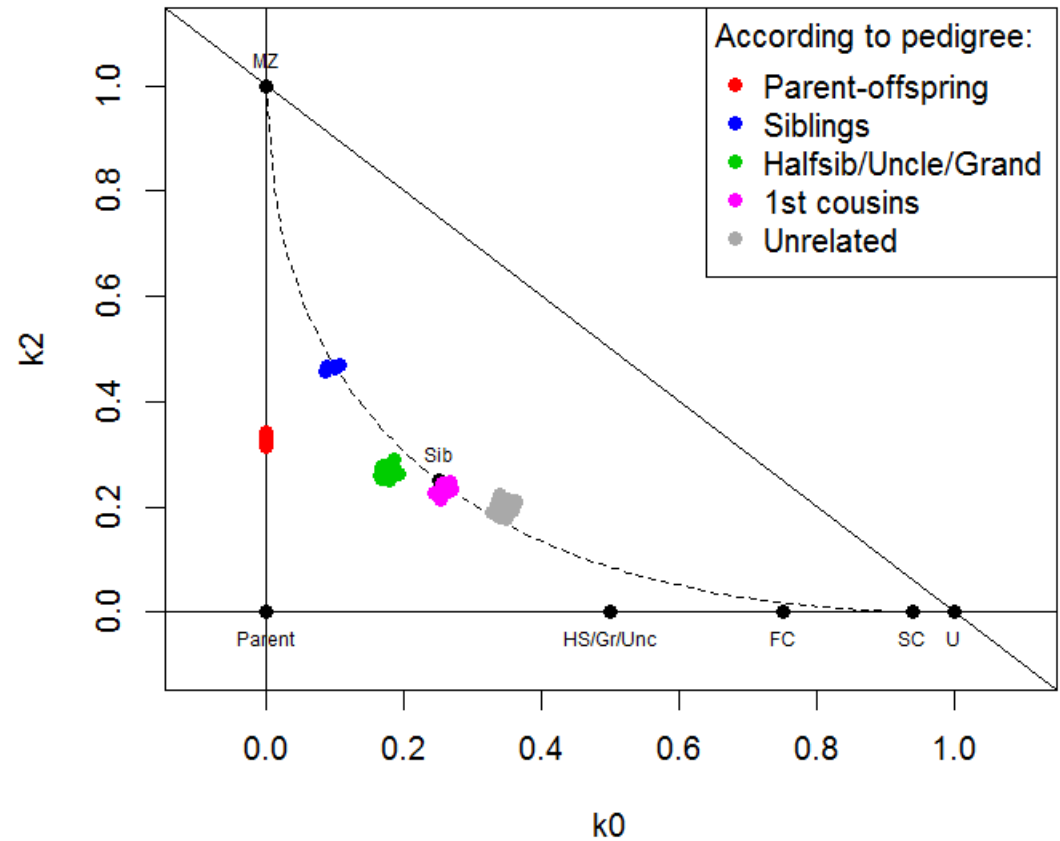
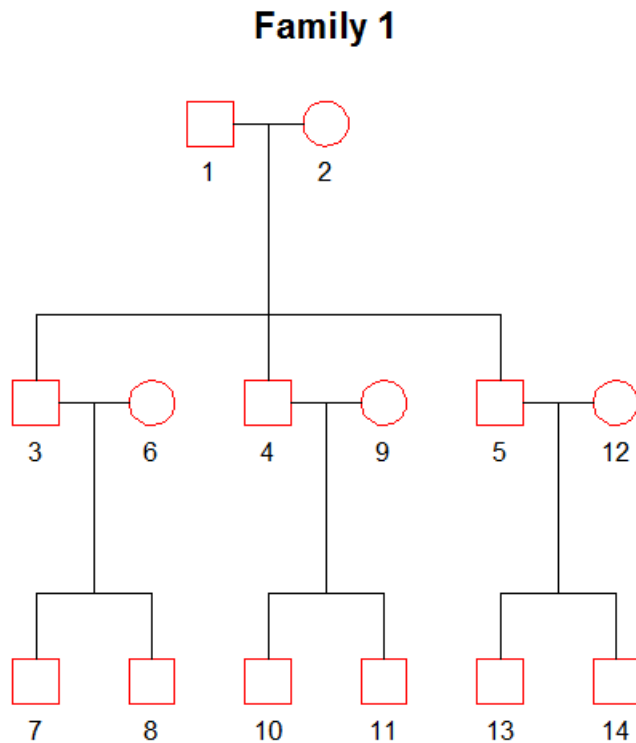


Simulation example

SNPs: 10 000

True frequency distr: Unif(0,1)

Frequencies used: All = 0.5

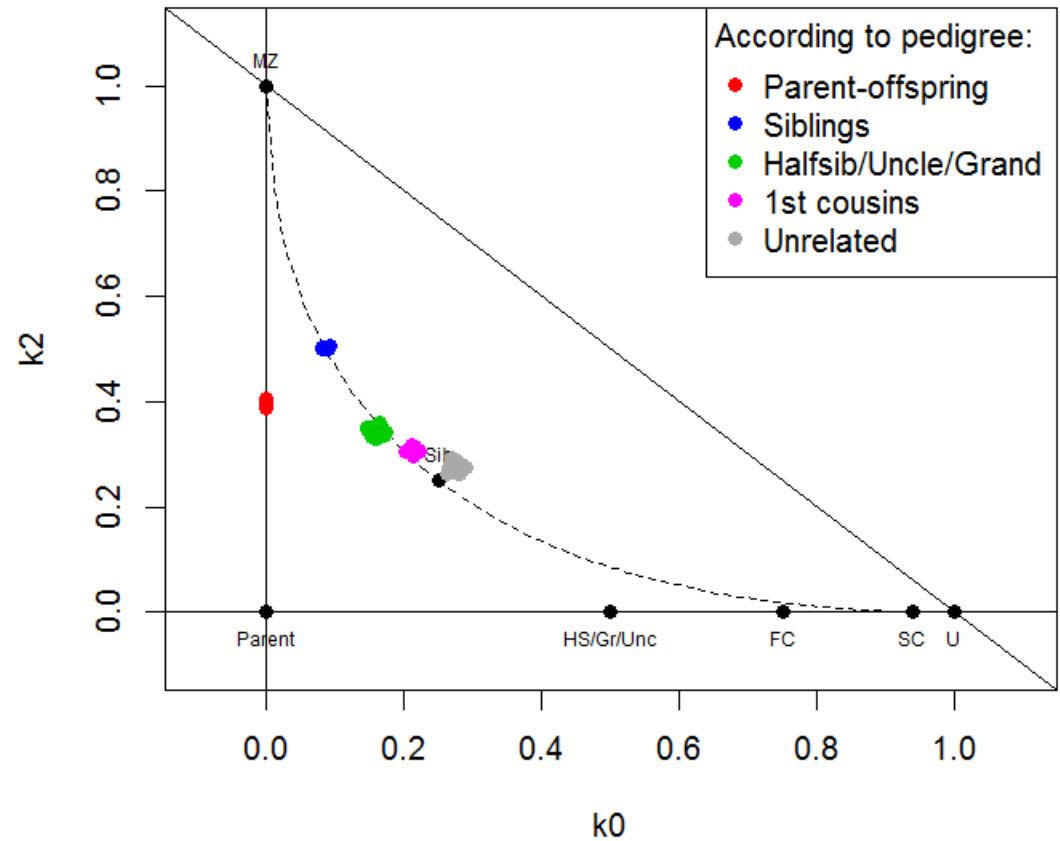
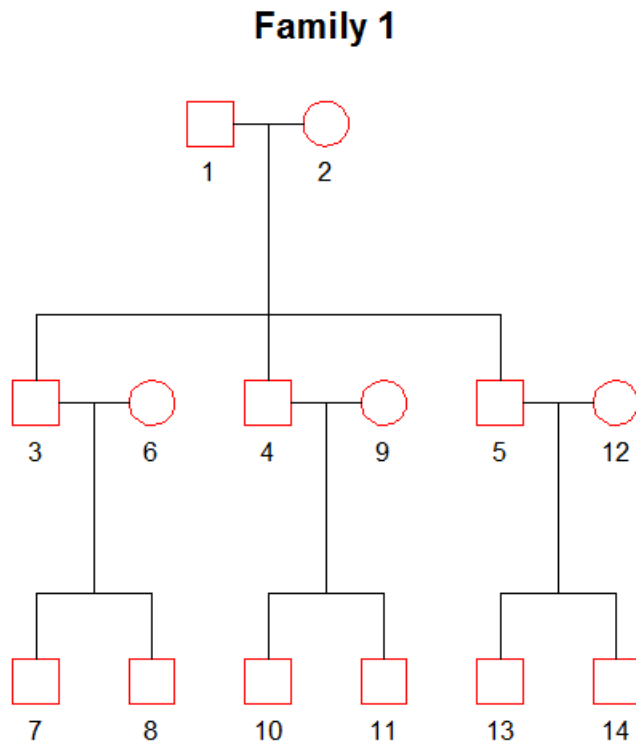


Simulation example

SNPs: 10 000

True frequency distr: Unif(0,1)

Frequencies used: Unif(0,1)

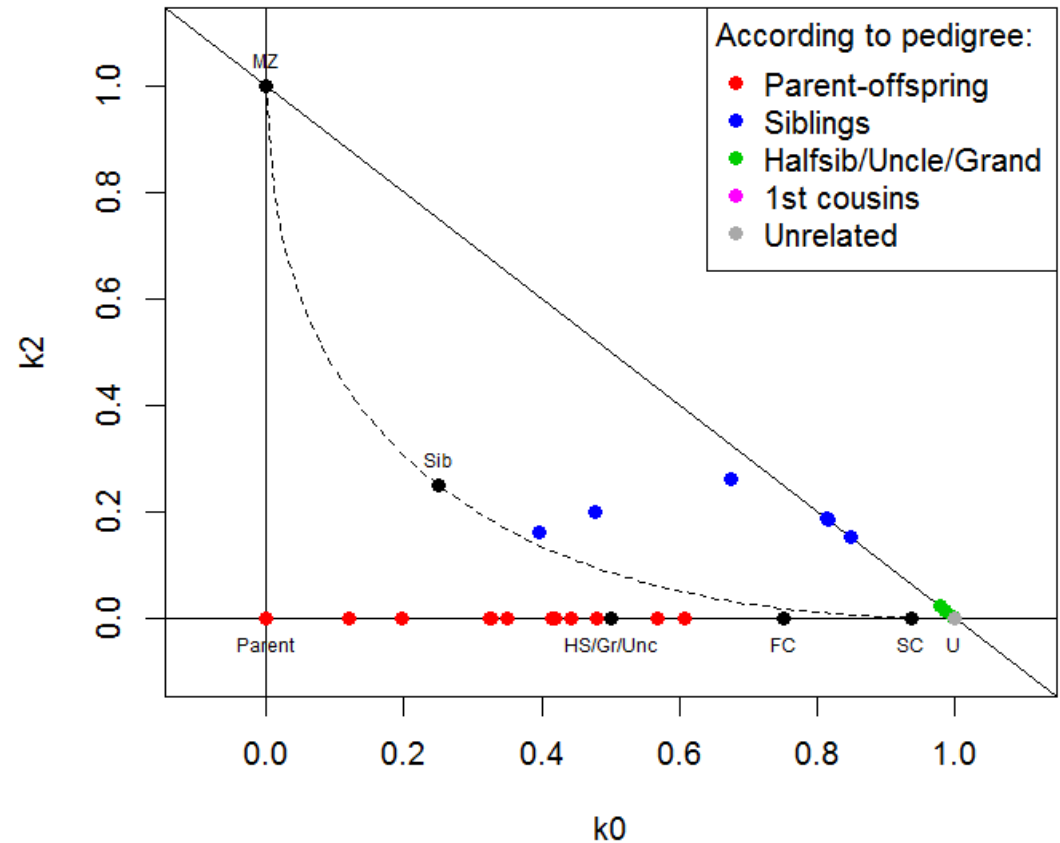
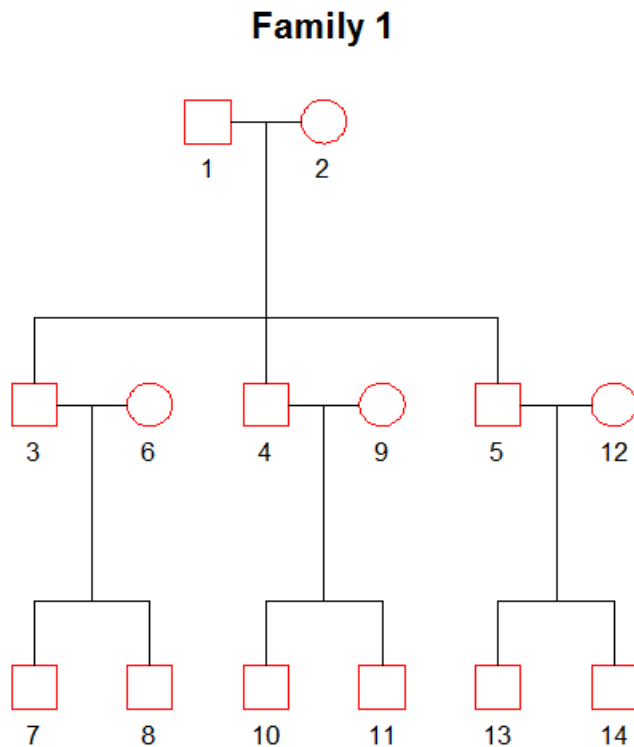


Simulation example

SNPs: 10 000

True frequency distr: Unif(0,1)

Frequencies used: Family estimate

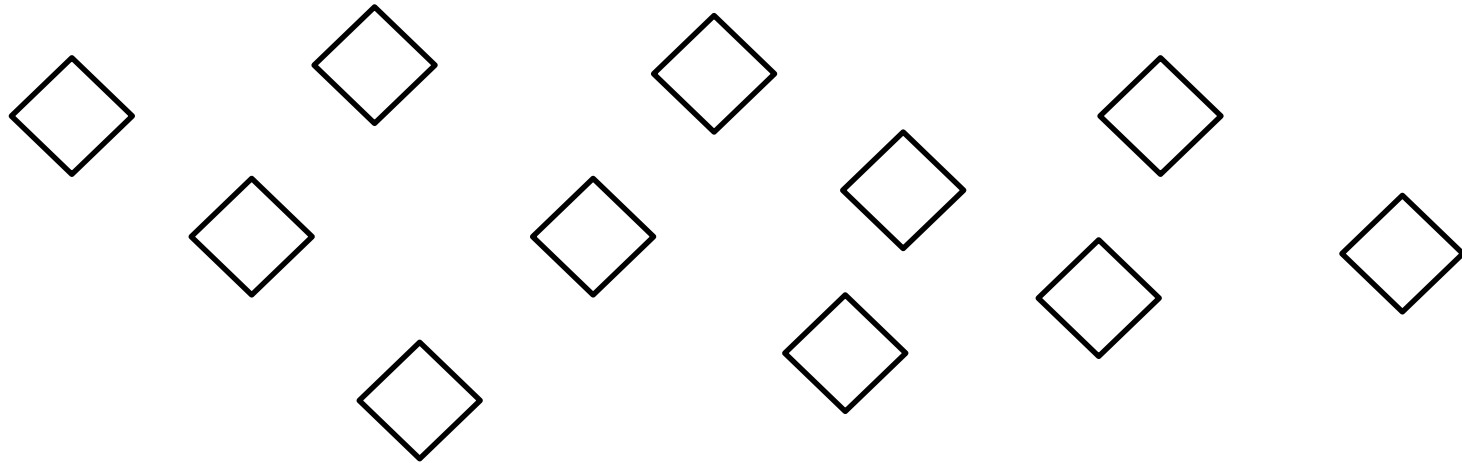


- Conclusion: The pairwise inference is quite sensitive to wrong allele frequencies



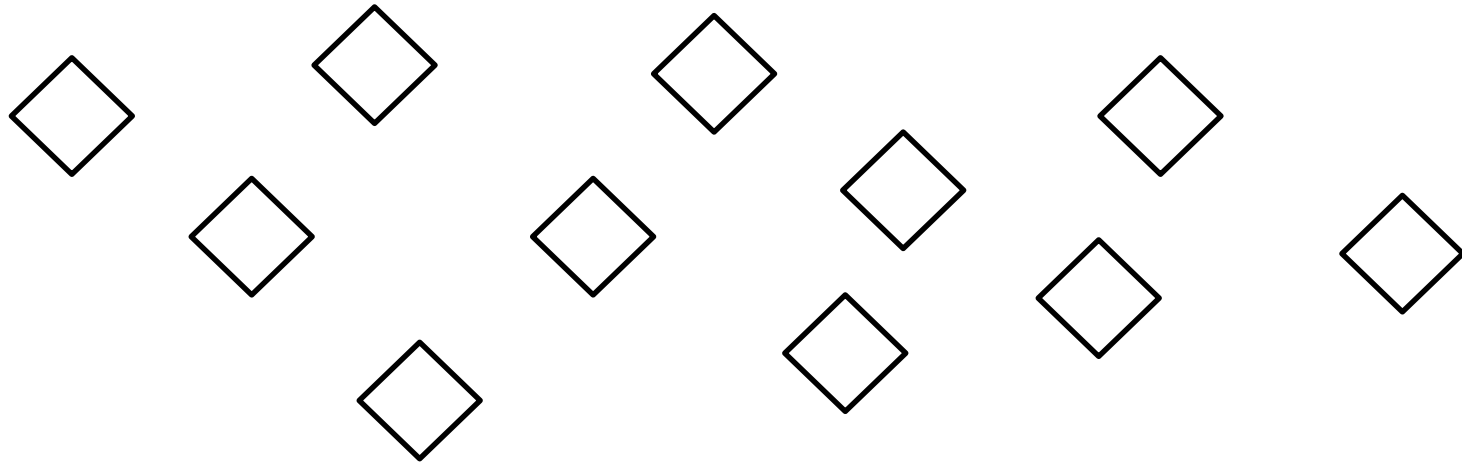
Part 4: Pedigree reconstruction

Pedigree reconstruction



Goal: Reconstruct the complete pedigree from genotype data

Pedigree reconstruction



Naive approach

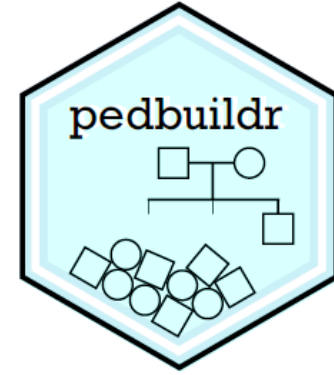
Step 1: Genders

Step 2: Estimate **pairwise** relationships

- Connect parent-child
- Exploit siblings

Step 3: **Solve the puzzle!**

Alternative method: R/pedbuildr



Idea:

- Generate a list of "all possible" pedigrees connecting the individuals
- Compute the likelihood of each pedigree
- Sort and output the best pedigrees

Key functions:

```
> buildPeds()      # generate pedigrees

> reconstruct()    # main function!

> plot()           # plot top hits
```

pedbuildr: Example

Same dataset as before:

Simulate 100 SNPs for a pair of siblings

```
> library(forrel)

> ids = c("Al", "Bob")
> x = nuclearPed(children = ids)

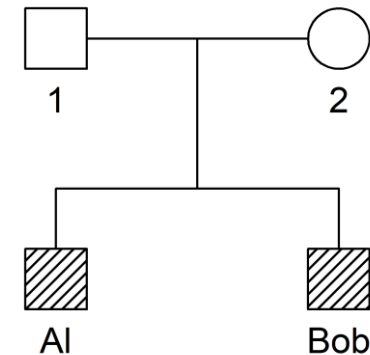
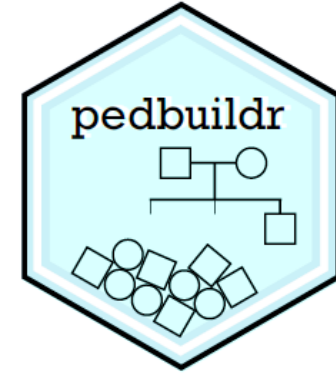
> x = markerSim(x, N = 100, ids = ids,
               alleles = 1:2, seed = 1234)

> x
```

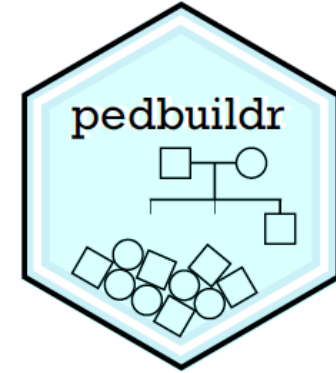
id	fid	mid	sex	<1>	<2>	<3>	<4>	<5>
1	*	*	1	-/-	-/-	-/-	-/-	-/-
2	*	*	2	-/-	-/-	-/-	-/-	-/-
Abe	1	2	1	1/1	1/2	1/1	1/2	2/2
Bob	1	2	1	1/1	1/2	1/1	1/2	2/2

Only 5 (out of 100) markers are shown.

```
> dat = list(subset(x, "Al"),
              subset(x, "Bob"))
```



pedbuildr: Example



Reconstruct the most likely

```
> library(pedbuildr)
```

```
> r = reconstruct(x)
```

Pedigree parameters:

ID labels: Al, Bob

Sex: 1, 1

Extra: parents

Age info: -

Known PO: -

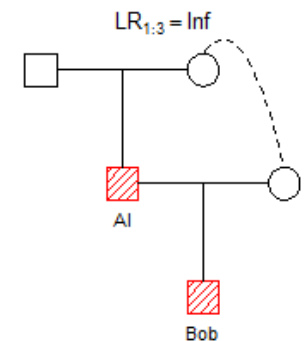
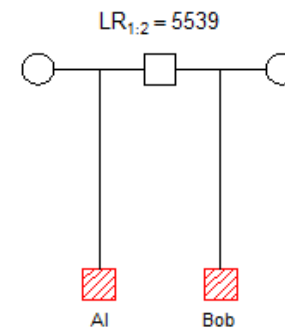
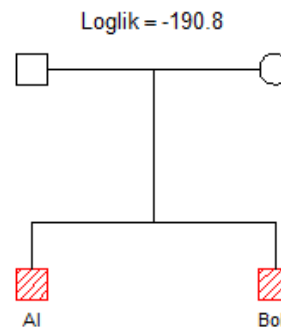
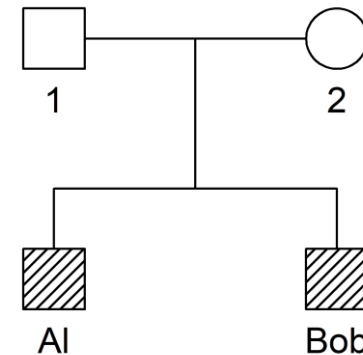
...

Building pedigree list:

...

Computing the likelihood of 6 pedigrees.

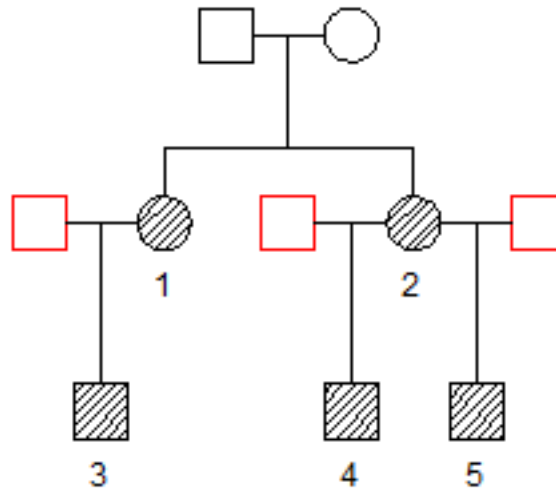
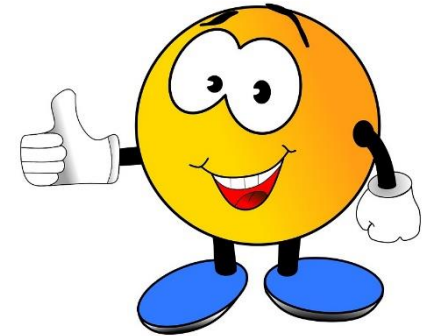
```
> plot(r, top = 3)
```



Optional parameters for restricting the search space

- **extra**: The number of extra individuals allowed to connect the original individuals
- **age**: A numerical age vector, or a character vector describing age inequalities
- **inferPO**: If TRUE, an initial stage of pairwise IBD estimation is done
- **knownPO**: Known parent–offspring pairs
- **notPO**: Pairs known not to be parent–offspring
- **allKnown**: If TRUE, then knownPO is the complete list of parent–offspring pairs
- **noChildren**: Individuals known to have no children
- **linearInb**: Set to FALSE to disallow inbreeding between linear descendants.
- **connected**: If TRUE (default), only connected pedigrees are considered.

Your turn: Exercises!



Q: Do any of the children have the same father?