

ISFG Summer School 2025 - Workshops 10.1 and 10.2

# **Kinship Statistics and Pedigree Analysis**

**Teachers**

Magnus Dehli Vigeland

Thore Egeland

## Home page:

[https://magnusdv.github.io/pedsuite/articles/web\\_only/course-isfg2025.html](https://magnusdv.github.io/pedsuite/articles/web_only/course-isfg2025.html)

# Schedule

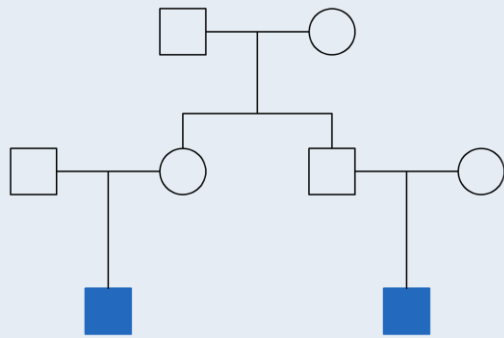
The course runs from 08 to 12 (CEST) each day, with a 15 minute break in the middle. The following schedule is tentative:

## Sept 4 (Thu). Pedigree analysis: Basic

- 08:00–09:00 [Pedigrees and measures of relatedness](#) (MDV)
- 09:00–10:00 [Exercises I](#)
- 10:00–10:15 *Break*
- 10:15–11:00 [Kinship testing](#) (TE)
- 11:00–11:45 [Exercises II](#)
- 11:45–12:00 Summary and discussion

## Sept 5 (Fri). Pedigree analysis: Advanced

- 08:00–09:00 **Relatedness inference and pedigree reconstruction** (MDV)
- 09:00–10:00 Exercises III
- 10:00–10:15 *Break*
- 10:15–11:00 **Disaster victim identification** (TE)
- 11:00–11:45 Exercises IV
- 11:45–12:00 Summary and discussion



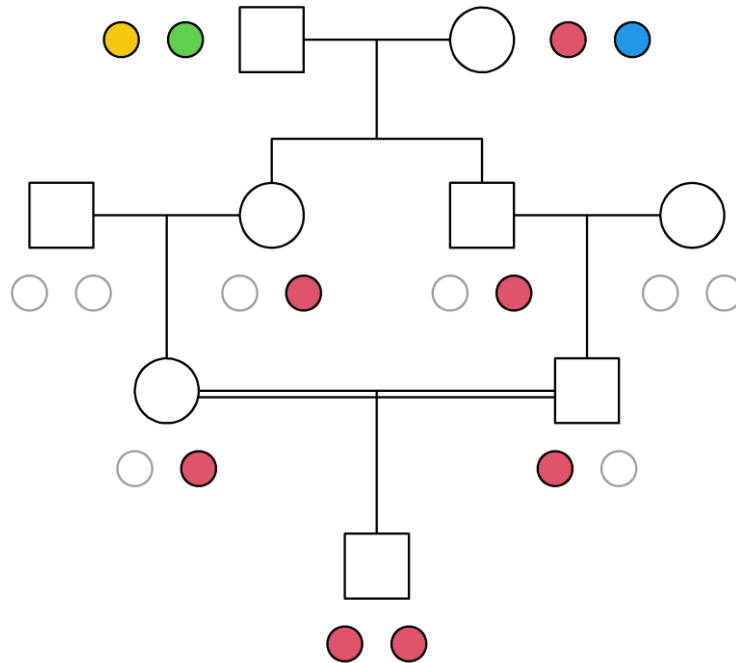
# Lecture 3: Relatedness inference and pedigree reconstruction

ISFG Summer School 2025 - Workshop 10.2

**Kinship Statistics and Pedigree Analysis: Advanced**

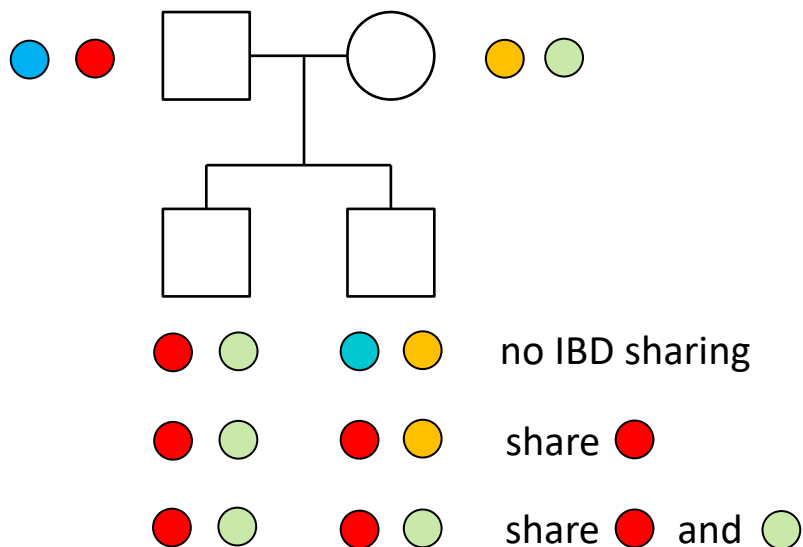
Magnus Dehli Vigeland

# Recap: Identity by descent (IBD)



# Recap: IBD coefficients

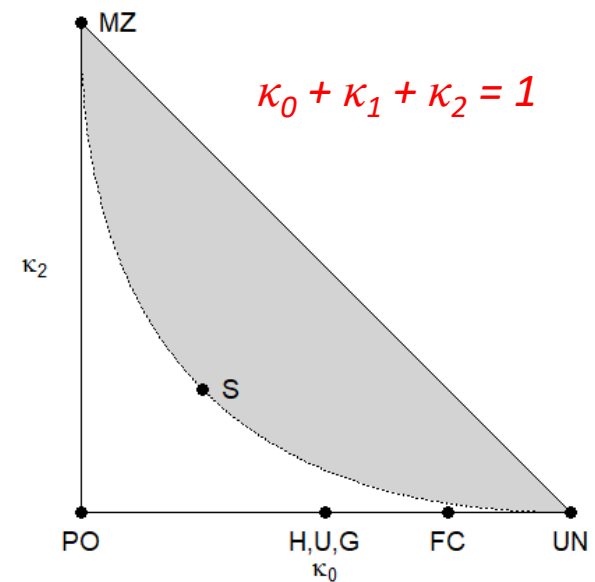
- How many alleles are IBD in each locus?



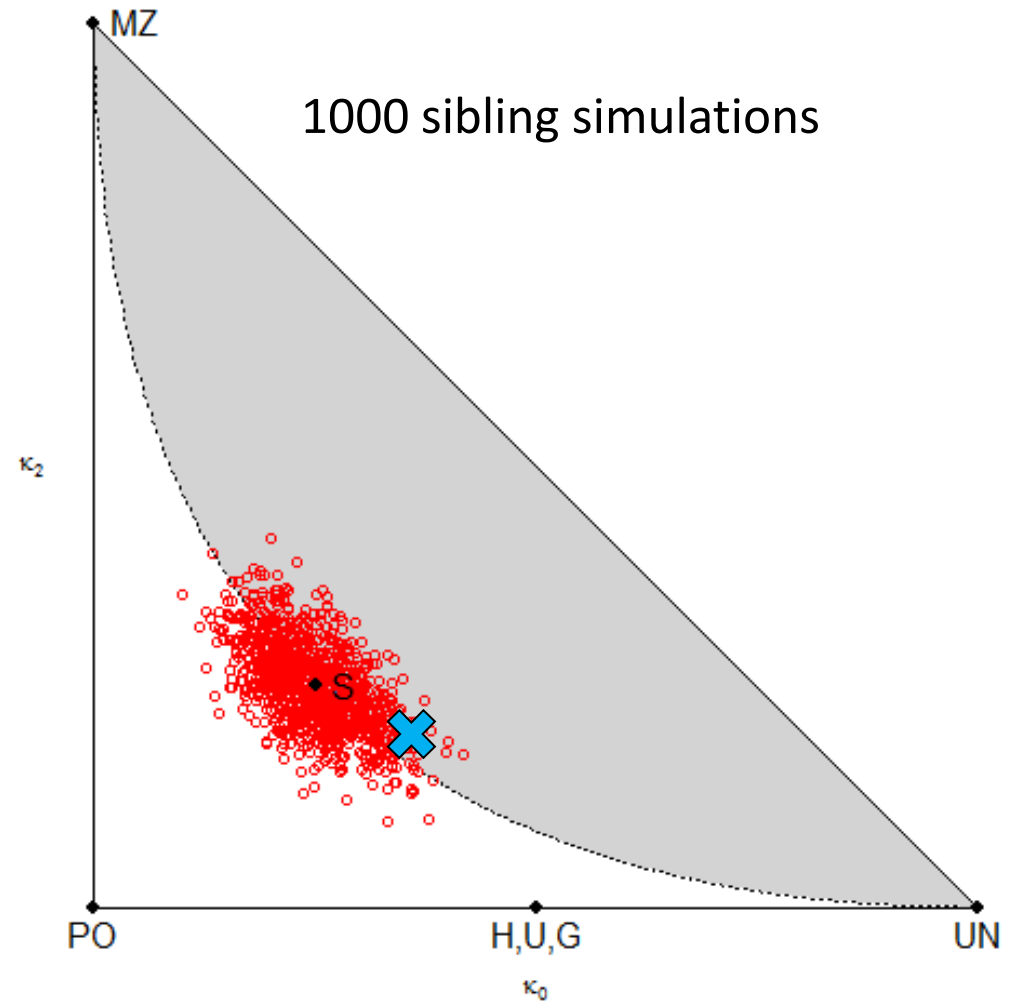
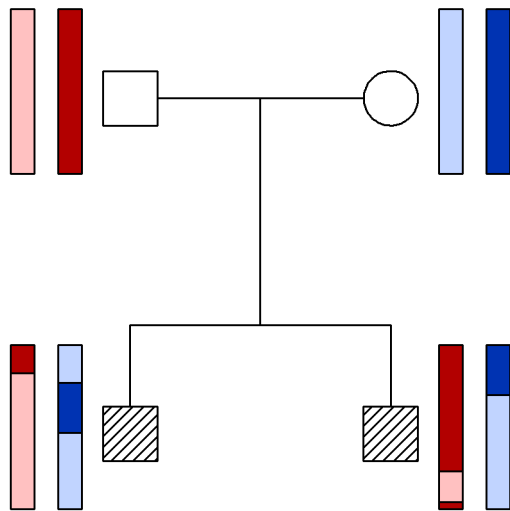
## Definition

- $\kappa_0 = P(0 \text{ alleles IBD})$
- $\kappa_1 = P(1 \text{ alleles IBD})$
- $\kappa_2 = P(2 \text{ alleles IBD})$

(at random autosomal locus)



# Recap: Realised relatedness





## Part I: Inference of *pairwise* relatedness

# Pairwise inference: Main approaches

## A. Based on IBD coefficients

- Typically with STR markers
- Maximum-likelihood estim.
- Assumes independence
  
- Complexity: **Easy**
- Accuracy: **Poor** (except PO/MZ)
- Scope: **Close relationships**

OLD SCHOOL

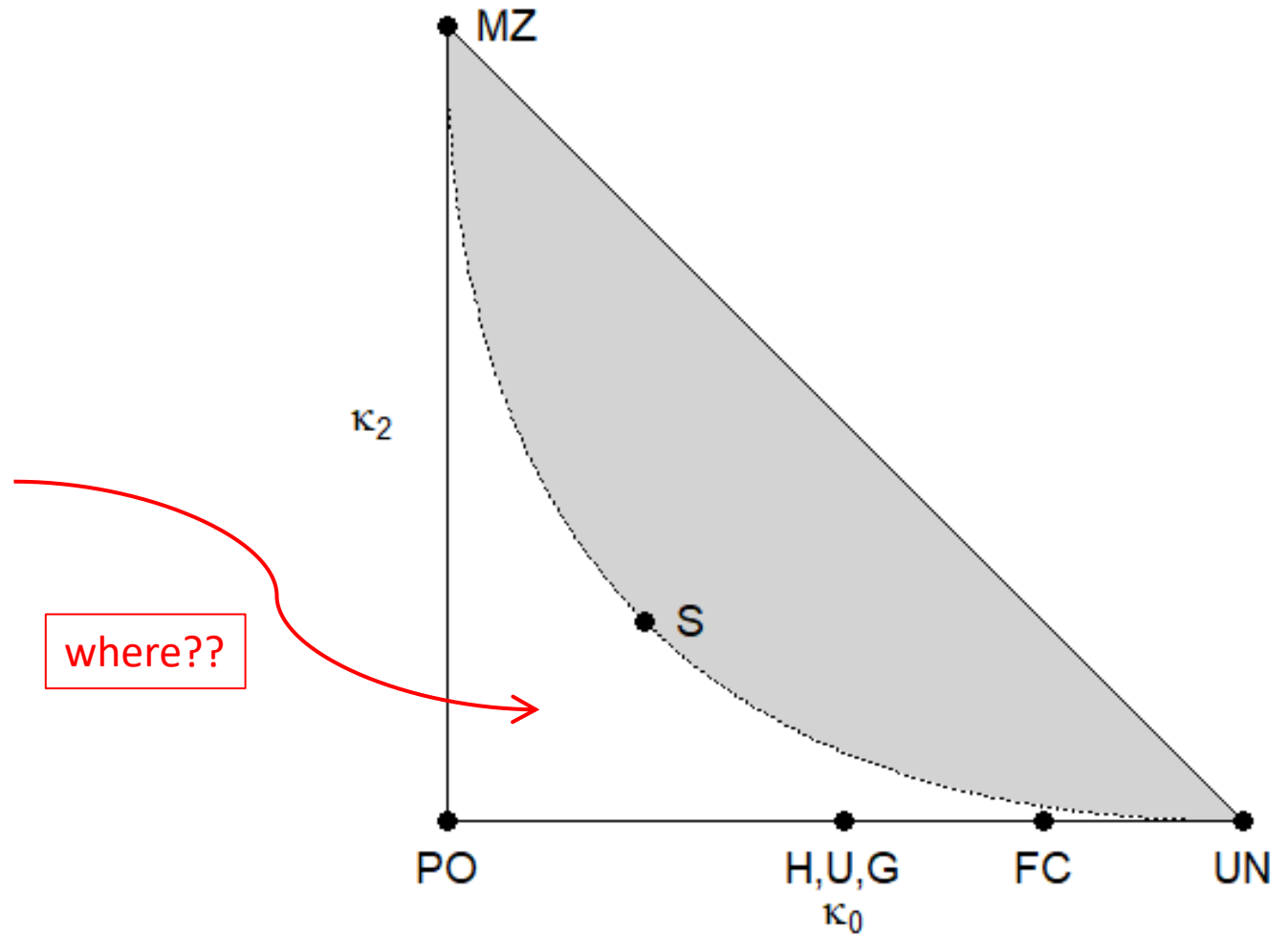
## B. Based on IBD segments

- Requires lots of SNPs
- Two steps:
  - 1) SNPs → IBD segments
  - 2) IBD segments → relatedness  
(Often different software)
  
- Complexity: **Medium/high**
- Accuracy: **Better**
- Scope: **Close + distant**

MODERN



# Approach A

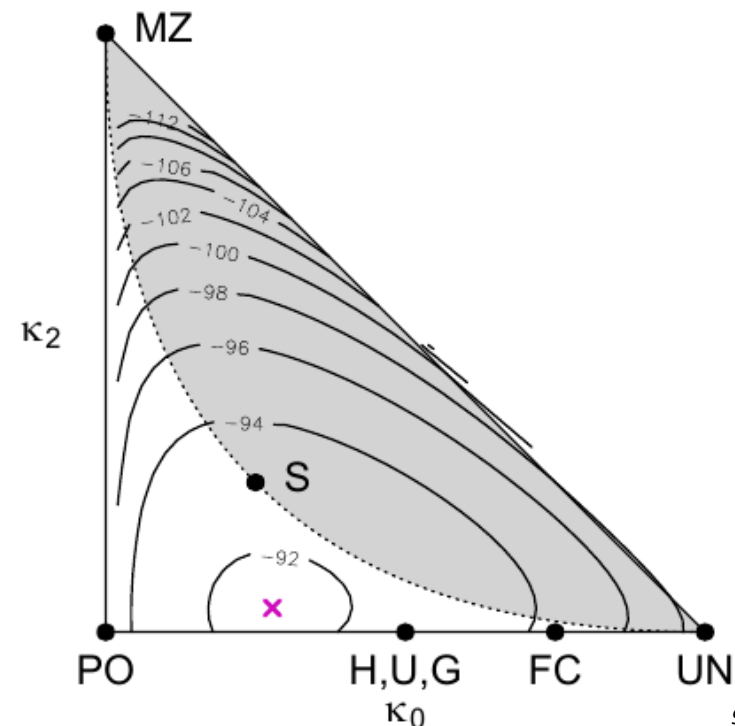


# Maximum likelihood estimation of $\kappa = (\kappa_0, \kappa_1, \kappa_2)$

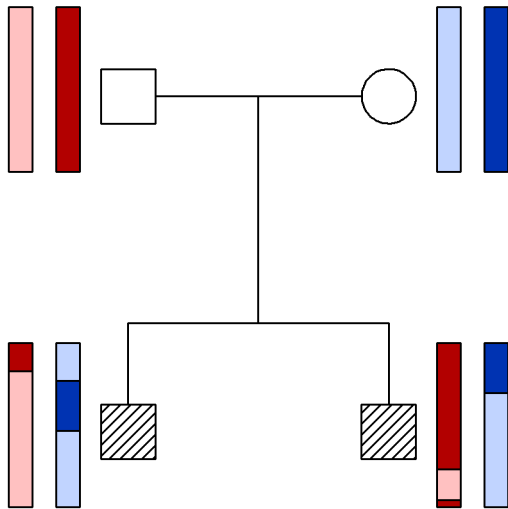
- Thompson (1975)
  - Given: marker genotypes for two individuals
  - The likelihood function

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

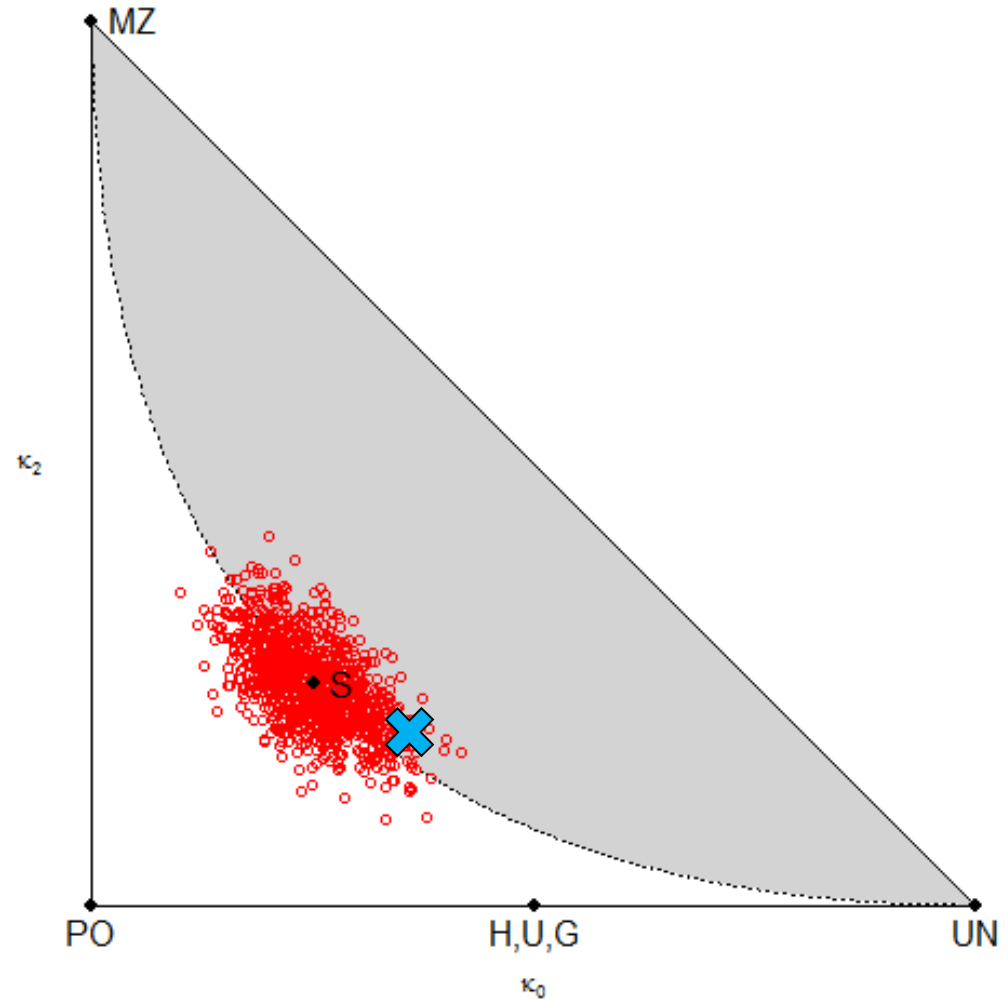
- Find the point  $k$  which maximizes  $L$ !
  - Called the maximum likelihood estimate (MLE)
- Assumptions:
  - known allele freqs
  - HWE
  - no inbreeding



# What are we estimating?



Answer: The *realised* coefficients!



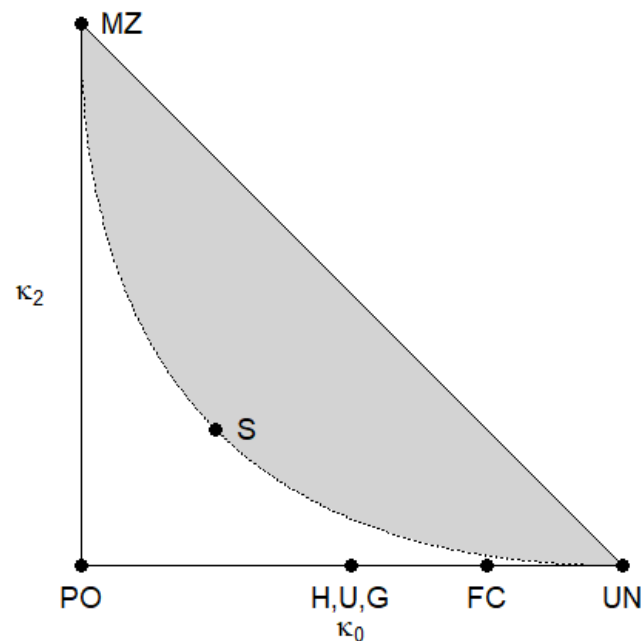
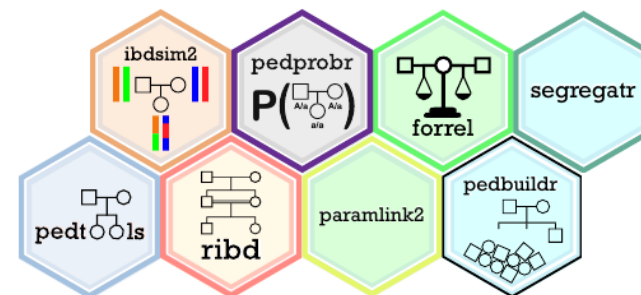
# Implementations

- R

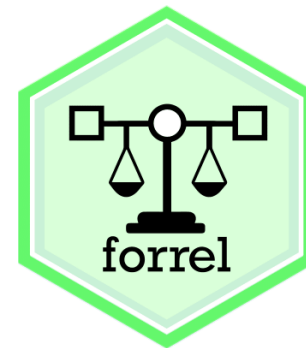
- pedsuite (forrel)
- SNPrelate, GWASTools (optimized for association studies)
- CrypticIBDcheck (as above, slow with many markers)
- +++

- Other

- PLINK
- KING
- Beagle
- +++



# Pairwise inference in R



- 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 in R: Example



## Simulate 100 SNPs for a pair of siblings

```
> library(pedsuite) # includes forrel

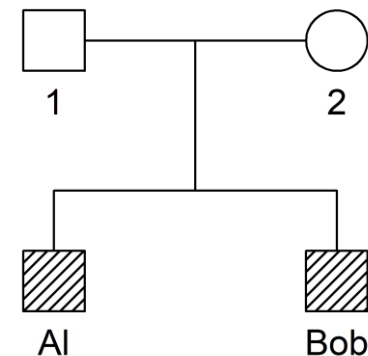
> ids = c("Al", "Bob")
> x = nuclearPed(children = ids)

> 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 -/- -/- -/- -/- -/-
  Al 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"))
```

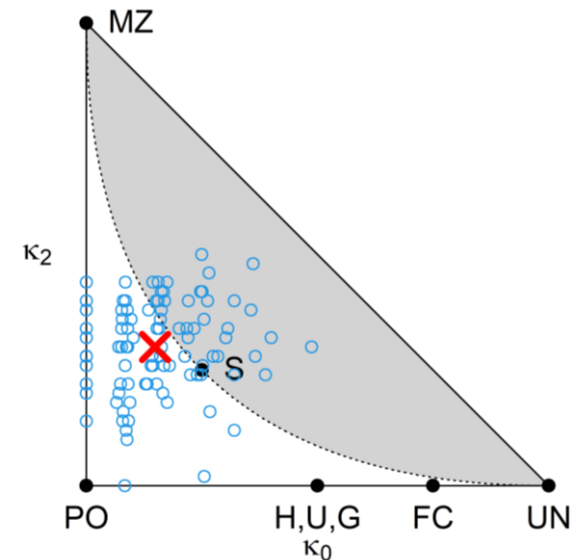
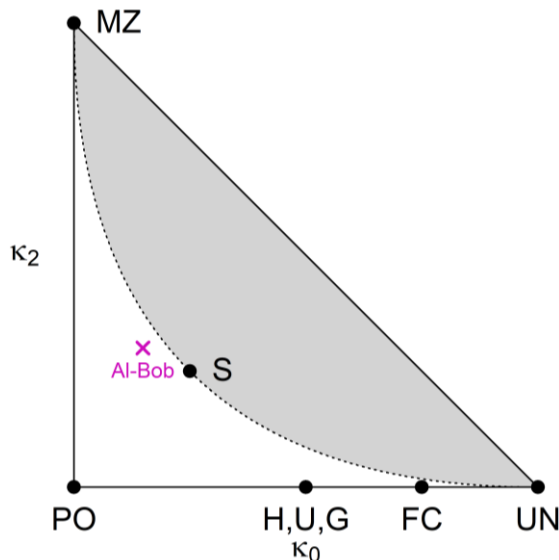
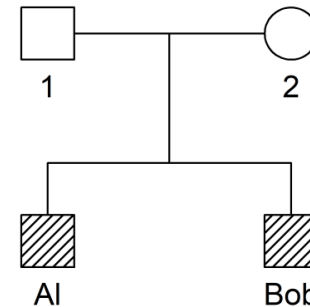


# Pairwise inference in R: Example



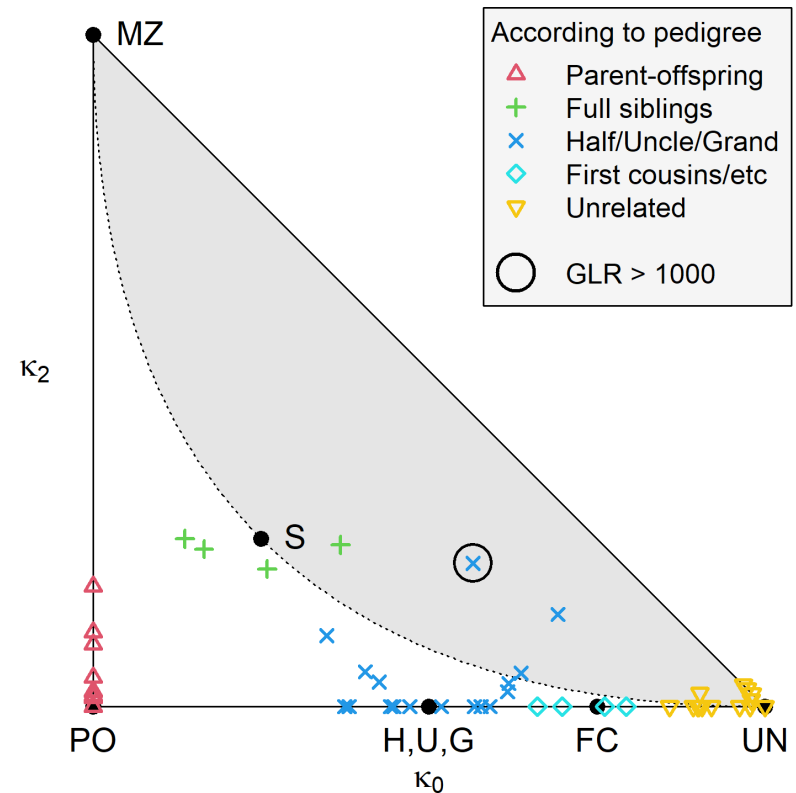
## Estimate kappa from the data

```
> 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 marker data
- 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 (actually: generalised LR\*)
  - Color-coded plot according to relationship claimed by pedigree

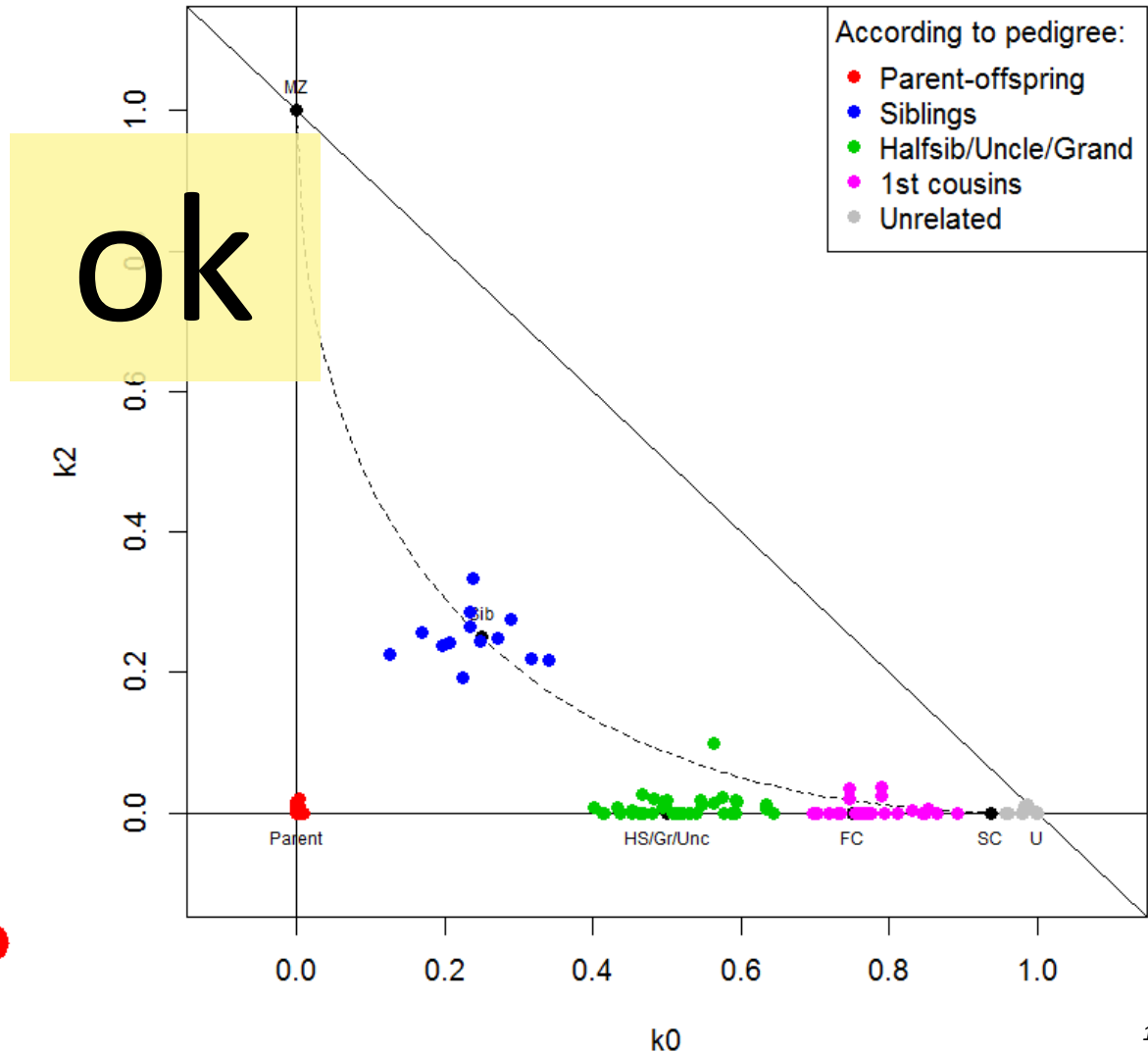
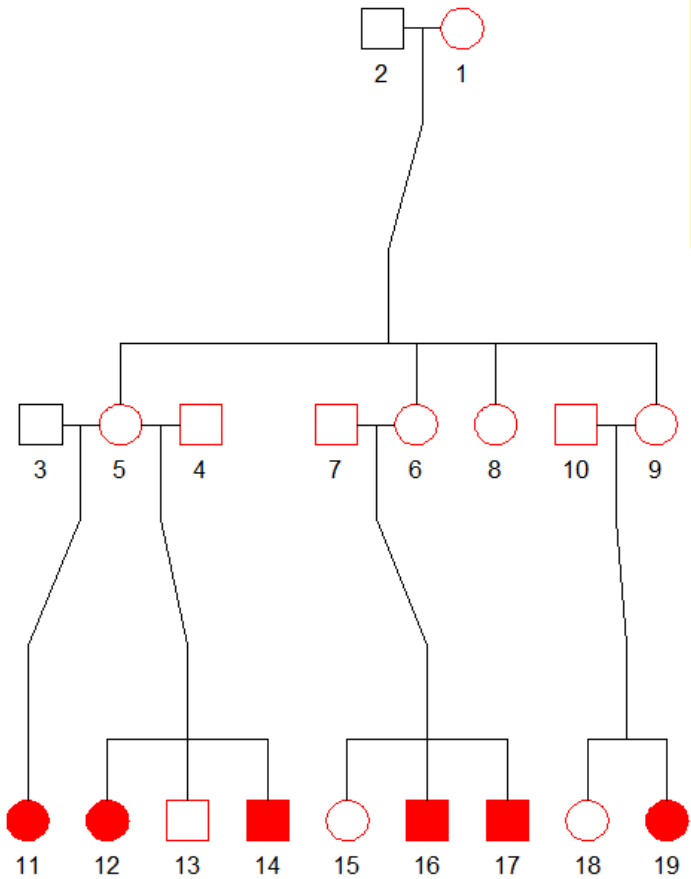


\*Egeland & Vigeland (FSI:Genetics, 2025): Kinship cases with partially specified hypotheses.



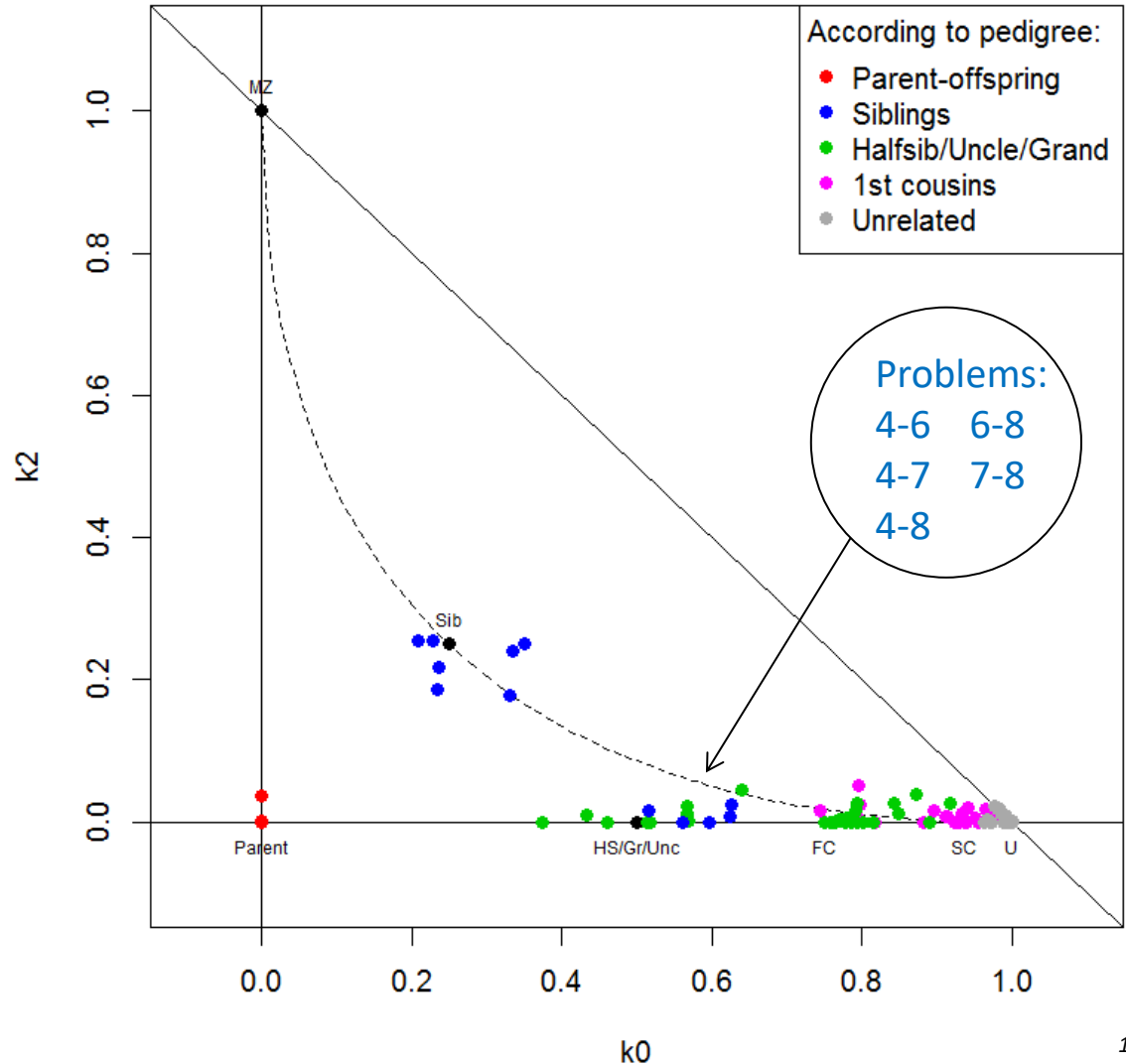
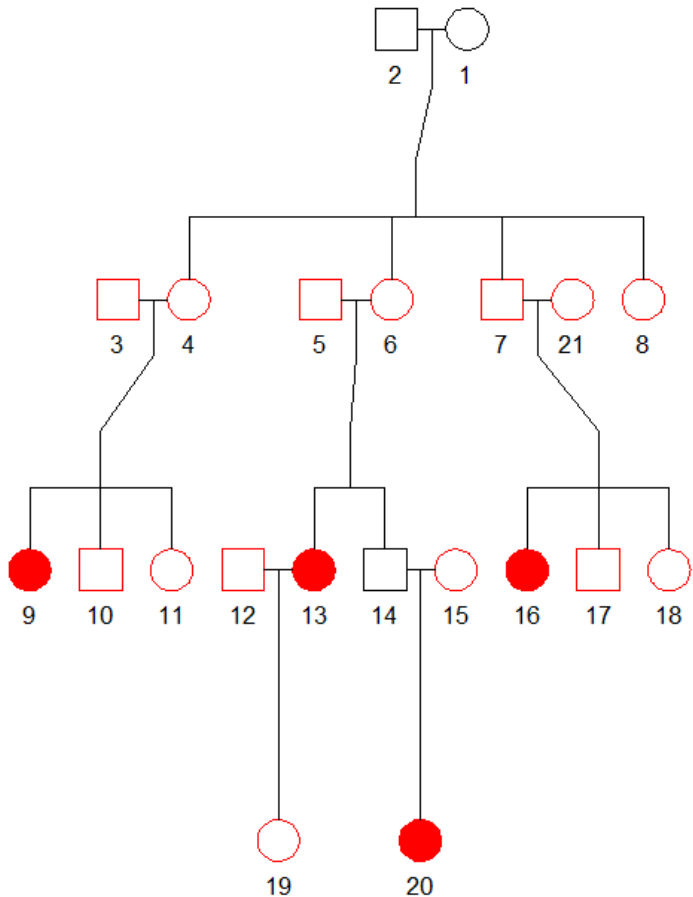
# checkPairwise(): Example 1

Family 22



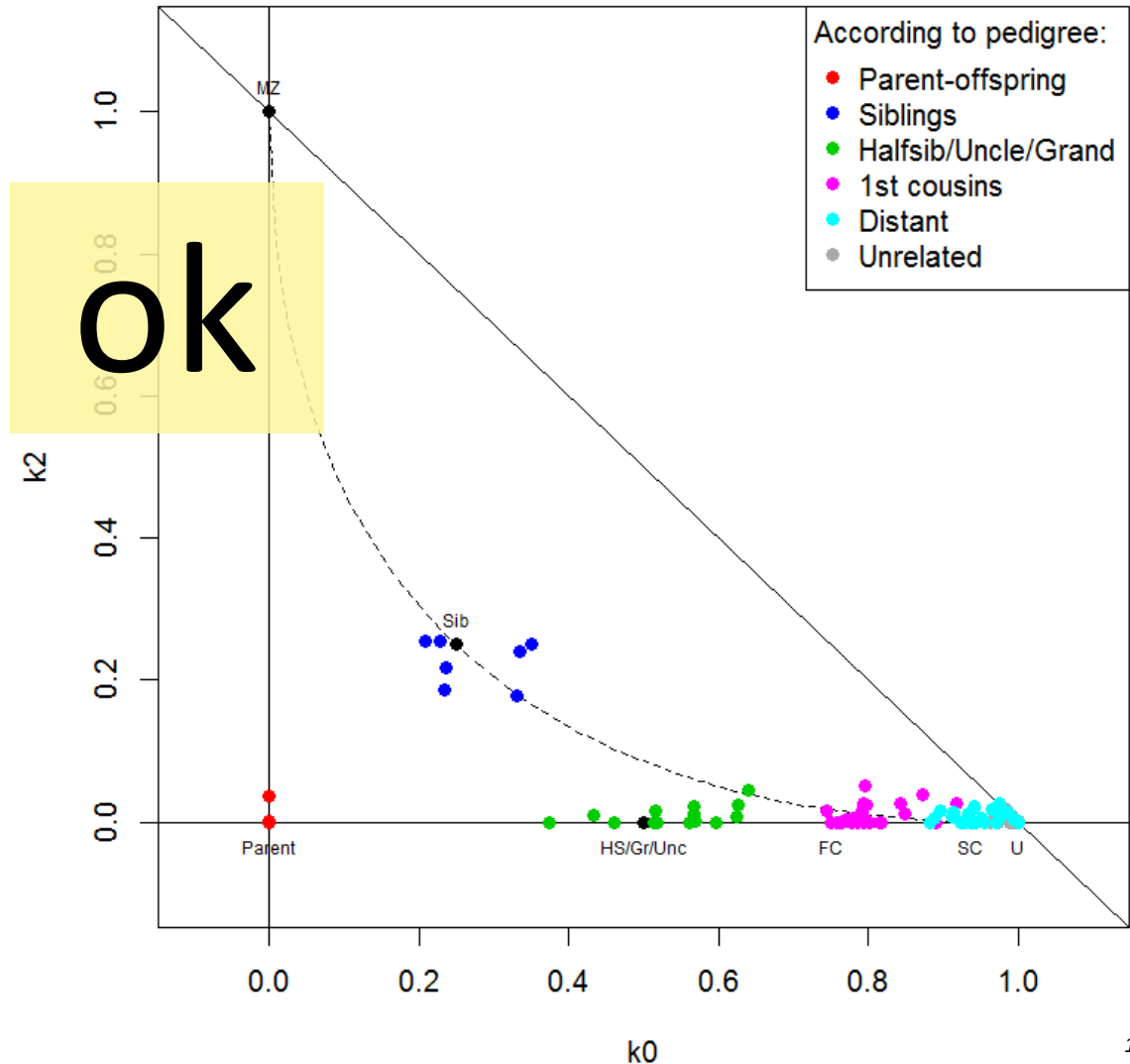
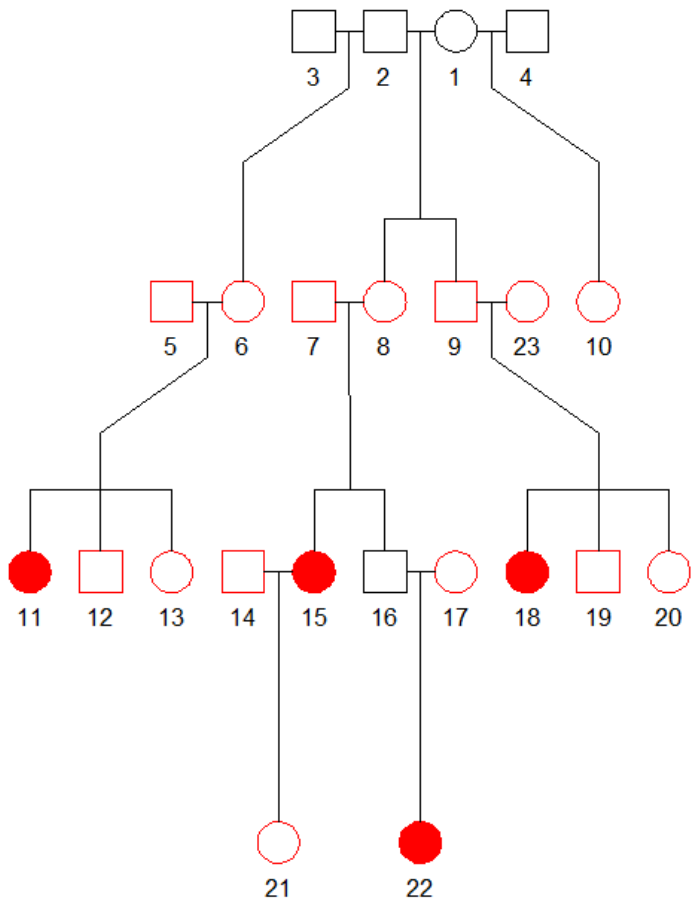
# checkPairwise(): Example 2

Family 16



# checkPairwise(): Example 2 - corrected

Family 16



# Pairwise inference: Main approaches

## A. Based on coefficients

- Typically with STR markers
- Maximum-likelihood estim.
- Assumes independence
- Complexity: **Easy**
- Accuracy: **Poor** (except PO/MZ)
- Scope: **Close relationships**

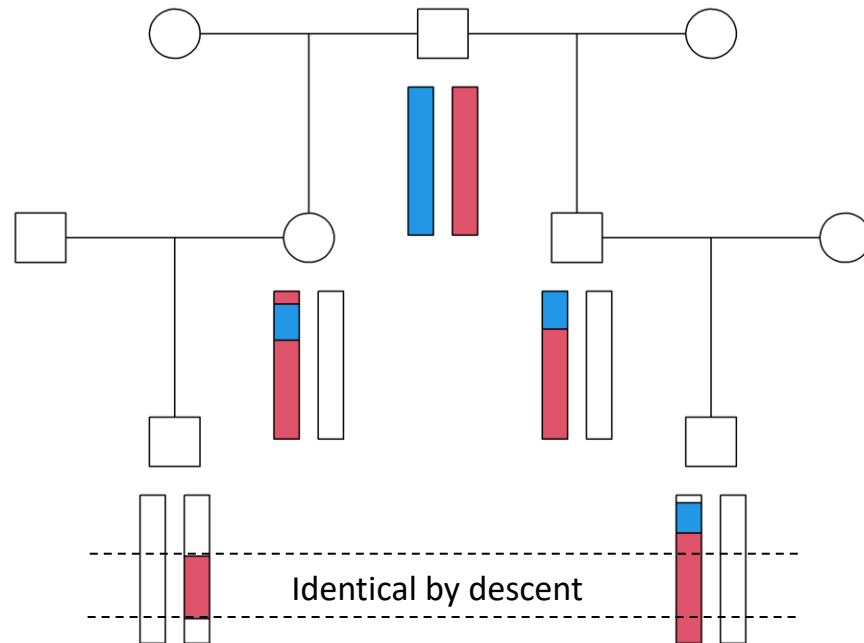
OLD SCHOOL

## B. Based on IBD segments

- Requires lots of SNPs
- Two steps:
  - 1) SNPs → IBD segments
  - 2) IBD segments → relatedness  
(Often different software)
- Complexity: **Medium/high**
- Accuracy: **Better**
- Scope: **Close + distant**

MODERN

# Remember: IBD comes in segments



# Software for IBD detection

- More than 30 published programs!
- But beware: Some of them ...
  - require > N markers (N = huge number)
  - require > M samples
  - require phased data
  - require massive installation+setup
  - don't work
  - don't exist anymore

Most unsuitable for forensic applications



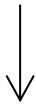
1	Software	Reference	Comments
2	PLINK	<a href="#">Purcell et al., 2007</a>	Probabilistic,
3	BEAGLE HMM	<a href="#">Browning and Browning, 2007</a>	Probabilistic,
4	GERMLINE	<a href="#">Gusev et al., 2009</a>	Not probabili
5	Relate	<a href="#">Albrechtsen et al., 2009</a>	Probabilistic,
6	KING	<a href="#">Manichaikul et al., 2010</a>	
7	BEAGLE IBD	<a href="#">Browning and Browning, 2010</a>	Probabilistic,
8	fastIBD	<a href="#">Browning and Browning, 2011</a>	
9	DASH	<a href="#">Gusev et al., 2011</a>	
10	MCMC_IBDfinder	<a href="#">Moltke et al., 2011</a>	
11	IBDL	<a href="#">Han and Abney, 2011</a>	Probabilistic,
12	ERSA	<a href="#">Huff et al., 2011</a>	
13	IBD_Haplo	<a href="#">Brown et al., 2012</a>	Probabilistic,
14	Refined IBD	<a href="#">Browning and Browning, 2013a</a>	
15	IBDseq	<a href="#">Browning and Browning, 2013b</a>	
16	IBD-Groupon	<a href="#">He, 2013</a>	
17	HapFABIA	<a href="#">Hochreiter, 2013</a>	
18	PREST-plus	<a href="#">Sun and Dimitromanolakis, 2014</a>	Likelihood
19	Parente2	<a href="#">Rodriguez et al., 2015</a>	
20	FISHR	<a href="#">Bjelland et al., 2017</a>	
21	hmmIBD	<a href="#">Schaffner et al., 2018</a>	Hidden Marko
22	DRUID	<a href="#">Ramstetter et al., 2018</a>	
23	TRUFFLE	<a href="#">Dimitromanolakis et al., 2019</a>	
24	RaPID	<a href="#">Naseri et al., 2019</a>	PBWT on pha
25	TRIBES	<a href="#">Twine et al., 2019</a>	
26	FastSMC	<a href="#">Nait Saada et al., 2020</a>	Hash/Extend
27	IBIS	<a href="#">Seidman et al., 2020</a>	Sliding Windo
28	Hap-IBD	<a href="#">Zhou et al., 2020</a>	Error adjuste
29	iLash	<a href="#">Shemirani et al., 2021</a>	Locality sensi
30	Phaseibd/TPBWT	<a href="#">Freyman et al., 2021</a>	TPWBT
31	ancIBD	<a href="#">Ringbauer et al., 2024</a>	
32	COANCESTRY	<a href="#">Wang, 2011</a>	
33	NGSremix		
34	NgsRelate		
35	IBDMAP	<a href="#">Bercovici et al., 2010</a>	HMM

# ibdfindr:

An R package for detecting IBD segments from SNP data

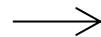
Input data

CHR	MARKER	CM	A1	FREQ1	CW063	CP620
1	rs944..	0.00	G	0.61	AG	AG
1	rs464..	0.33	C	0.64	AC	AC
1	rs1091..	1.24	T	0.63	CC	TT
1	rs669..	2.00	C	0.57	CC	CC
1	rs376..	4.81	T	0.59	GG	GT
1	rs736..	5.86	C	0.67	CT	CT
1	rs667..	6.77	G	0.56	AA	GG

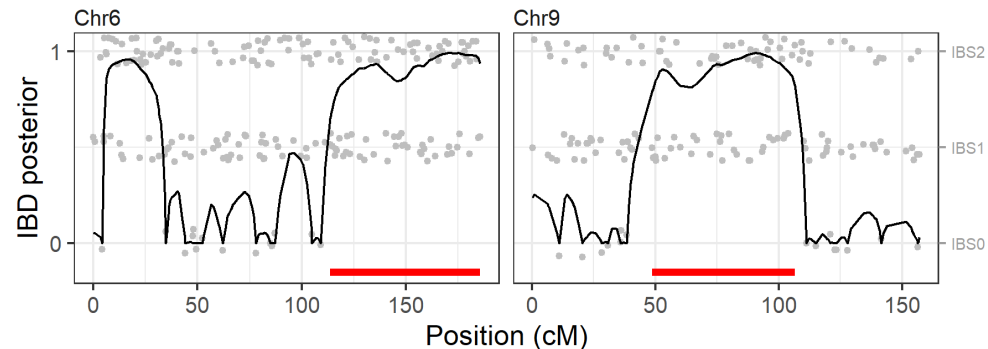


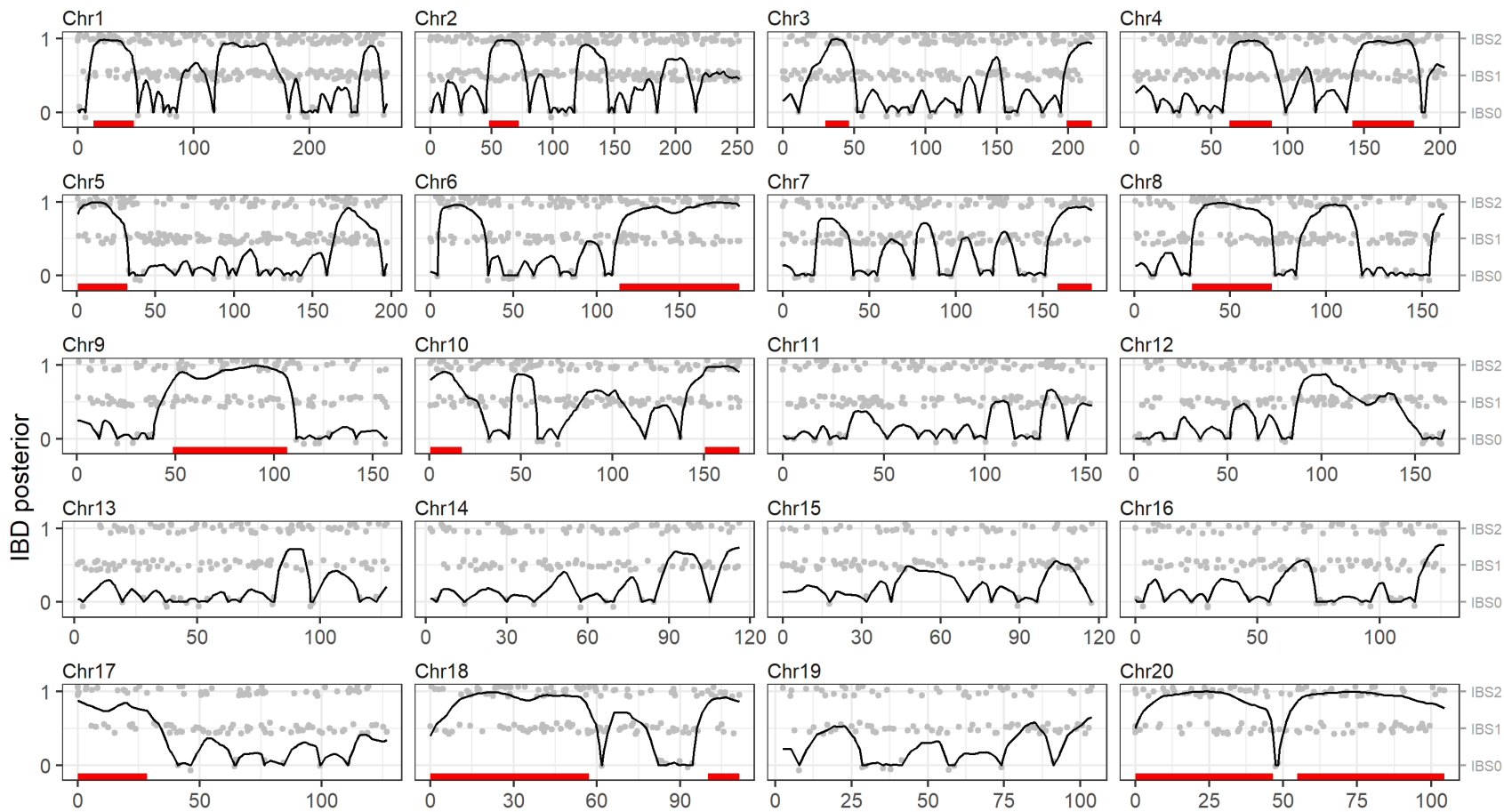
R code

```
library(ibdfindr)
ibd = findIBD(data)
plotIBD(ibd, chrom = c(6,9))
```



- Fits a continuous-time **Hidden Markov Model**
- Predicts IBD segments (Viterbi algorithm)
- Finds posterior IBD probs (forward-backward algorithm)

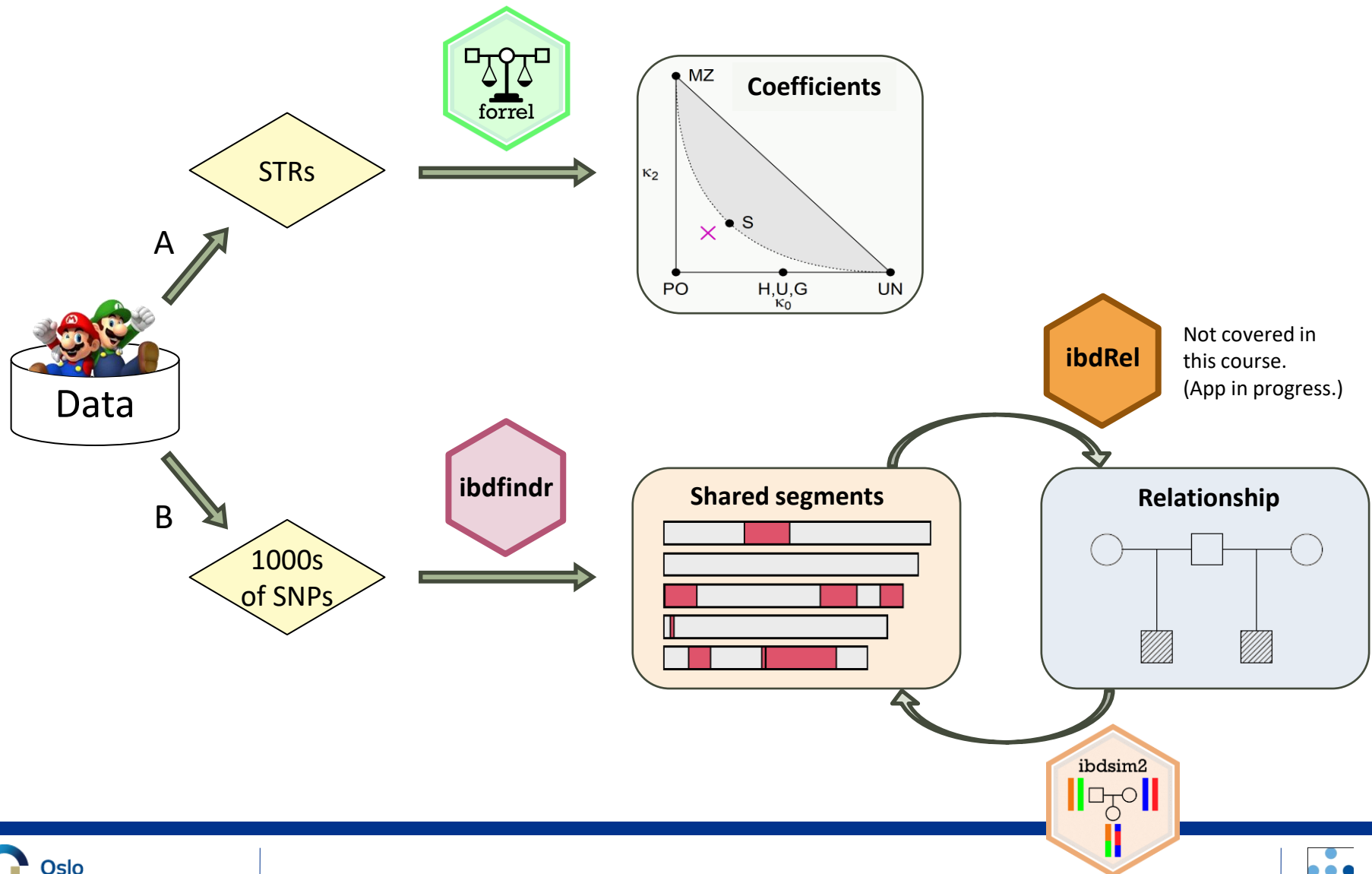




+ chrom 21, 22

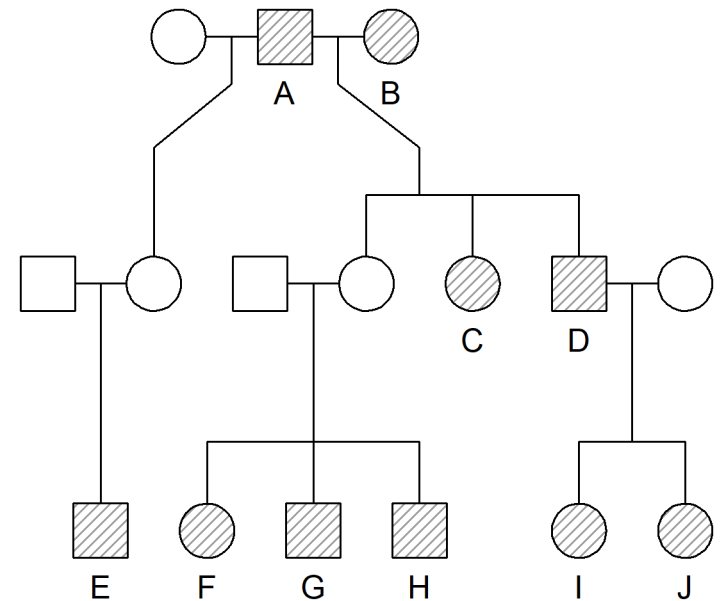
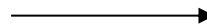
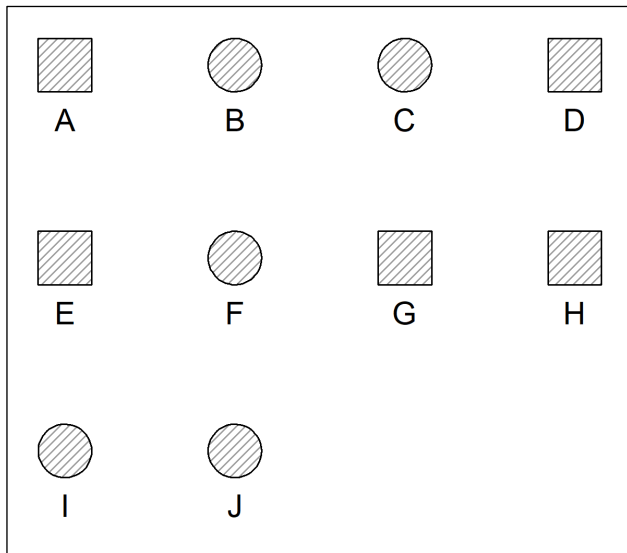


# Summary: Tools for pairwise kinship inference



## Part II: Pedigree reconstruction

# Pedigree reconstruction: Ultimate goal



# Generally impossible - even in theory!

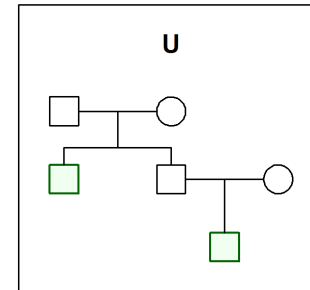


Data

A/A	A/C
G/A	A/A
C/C	C/C
C/G	G/G
...	...

?

True pedigree

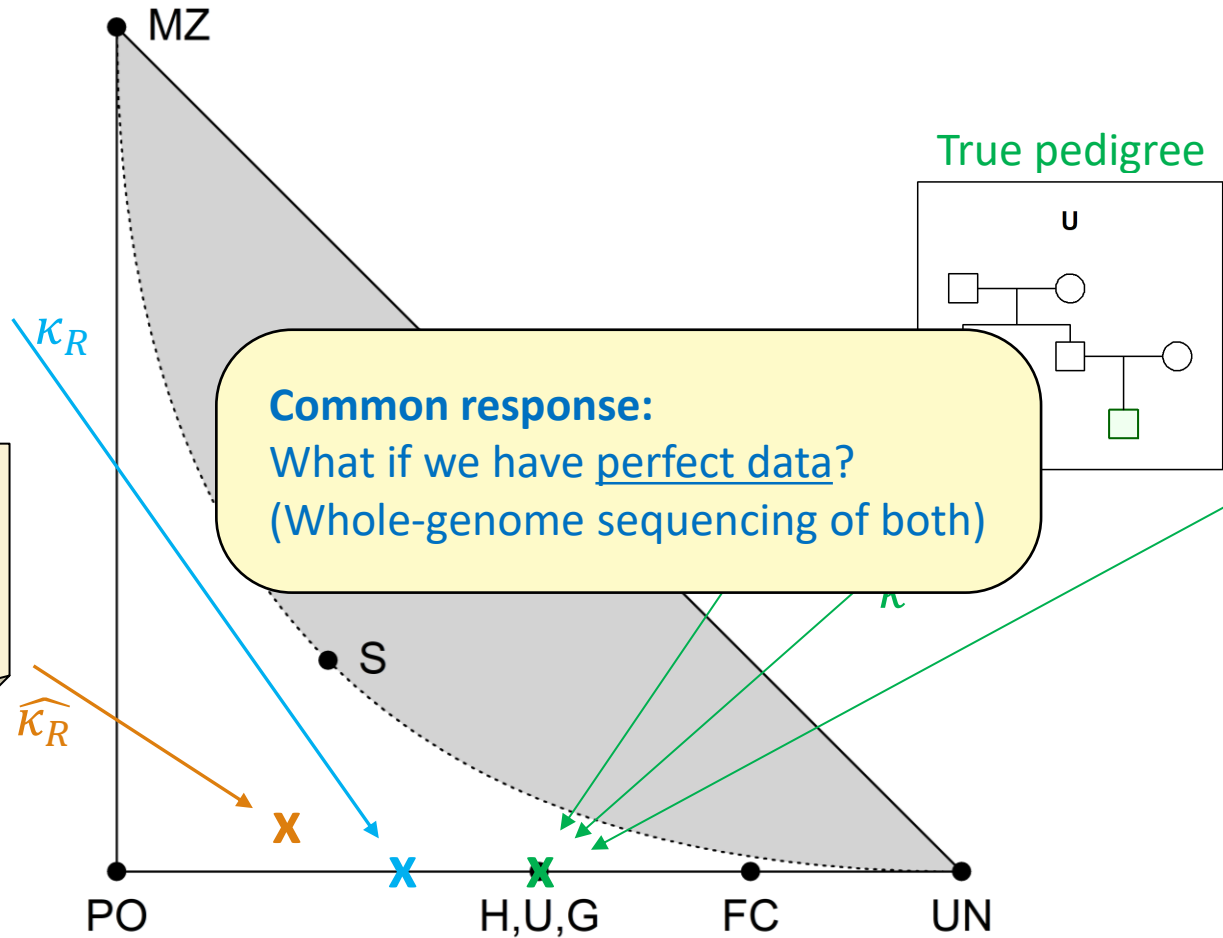


# Generally impossible - even in theory!



Data

A/A	A/C
G/A	A/A
C/C	C/C
C/G	G/G
...	...

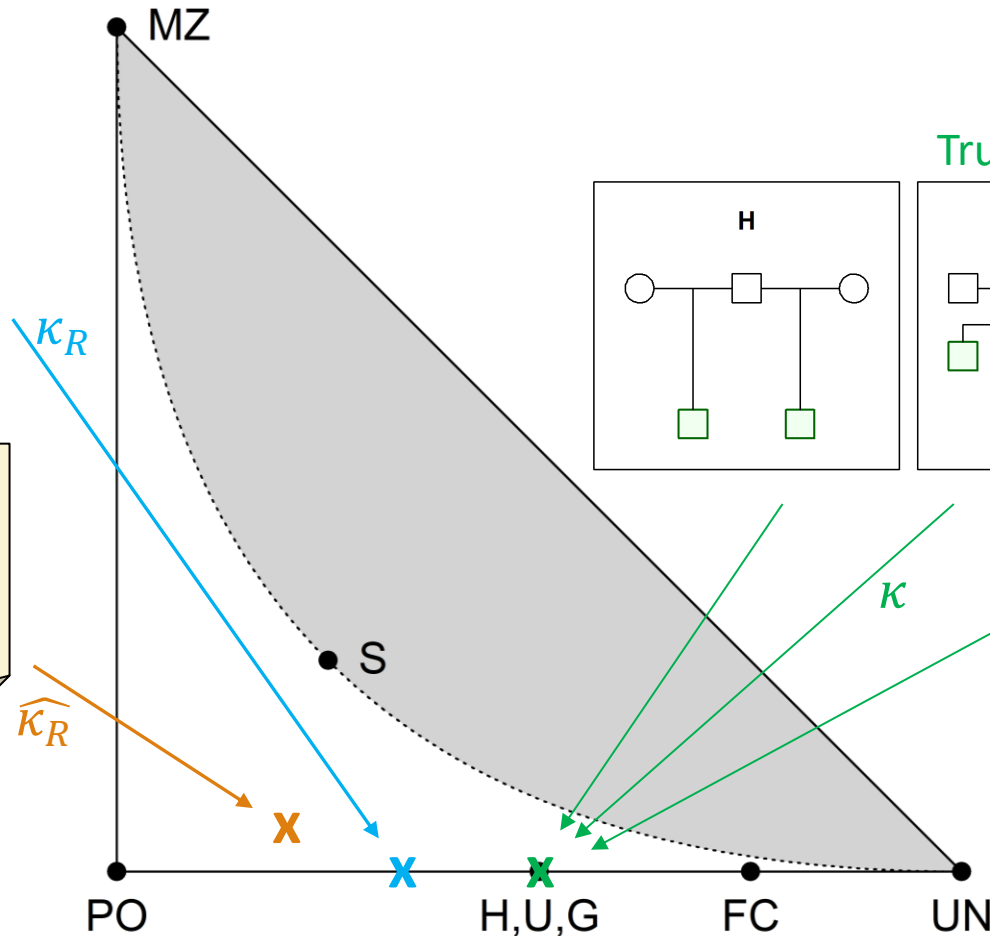


# Generally impossible - even in theory!

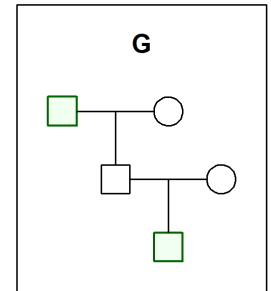
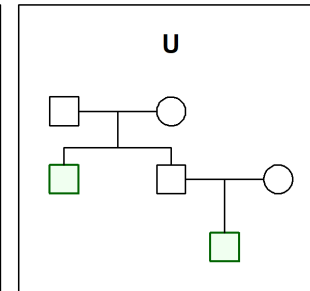
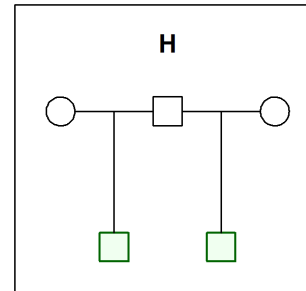


Data

A/A	A/C
G/A	A/A
C/C	C/C
C/G	G/G
...	...



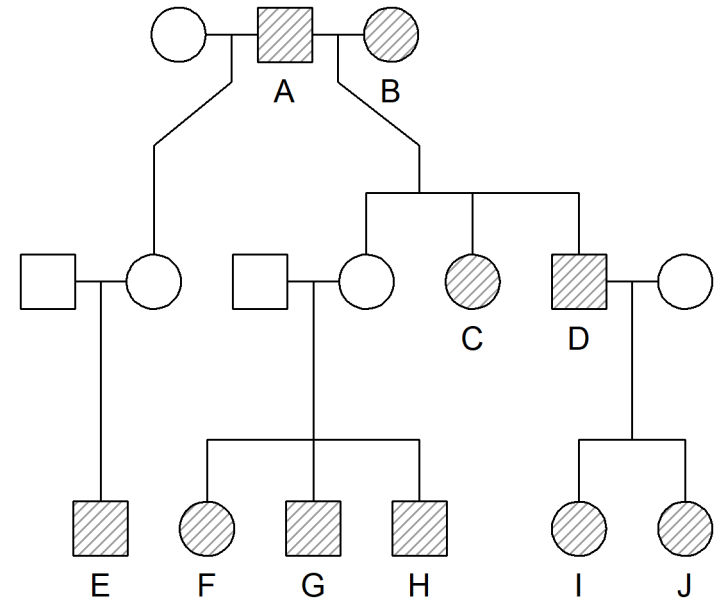
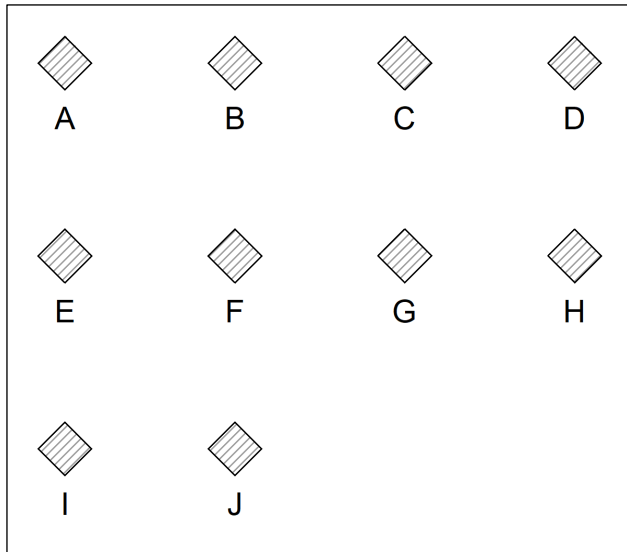
True pedigree



**What *can* we trust?**

- Parent-offspring
- Full sibs if good data

# Pedigree reconstruction: Naive approach



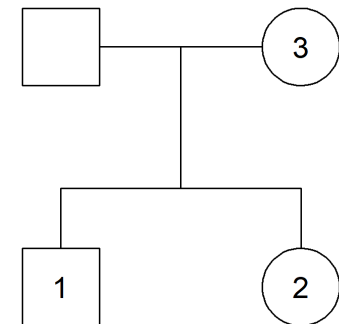
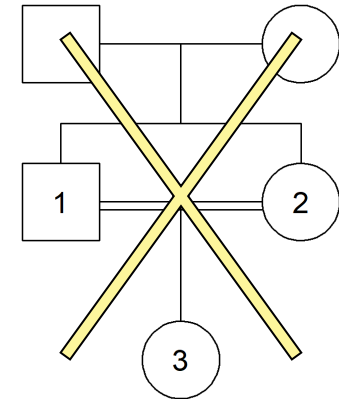
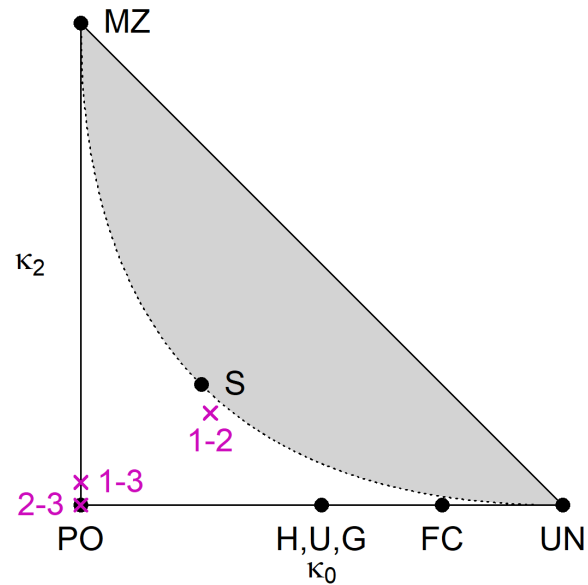
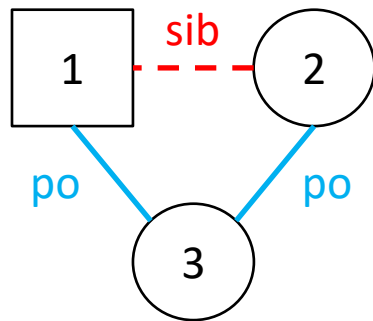
Step 1: Genders

Step 2: Estimate **pairwise** relationships

- Connect parent-child
- Exploit siblings

Step 3: **Solve the puzzle!**

# Example





# Better approach: Maximum likelihood

## Idea:

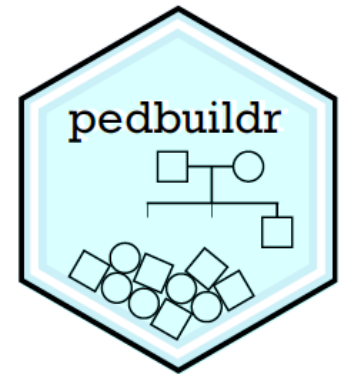
- Generate a list of "all possible" pedigrees connecting the individuals
- Compute the likelihood of each pedigree
- Sort and output the best pedigrees

## 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(pedsuite)

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

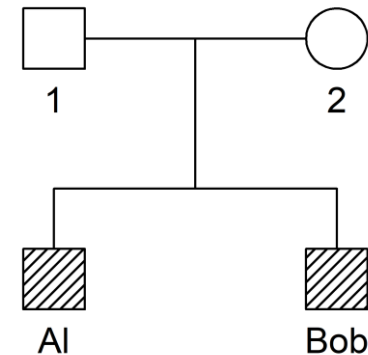
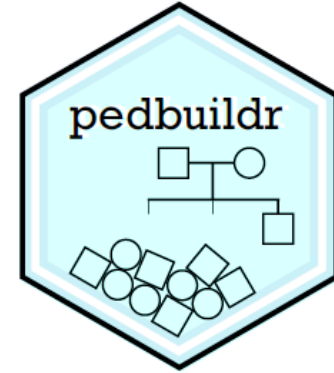
> x = markerSim(x, N = 100, ids = ids,
               alleles = 1:2, seed = 1234)

> x
```

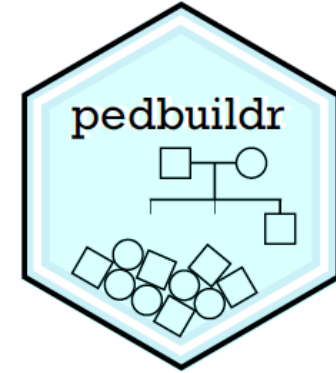
id	fid	mid	sex	<1>	<2>	<3>	<4>	<5>
1	*	*	1	-/-	-/-	-/-	-/-	-/-
2	*	*	2	-/-	-/-	-/-	-/-	-/-
Al	1	2	1	1/1	1/2	1/1	1/2	2/2
Bob	1	2	1	1/1	1/2	1/1	1/2	2/2

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(dat)
```

Pedigree parameters:

ID labels: Al, Bob

Sex: 1, 1

Extra: parents

Age info: -

Known PO: -

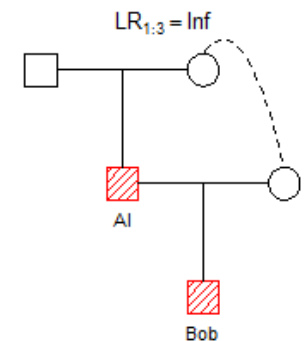
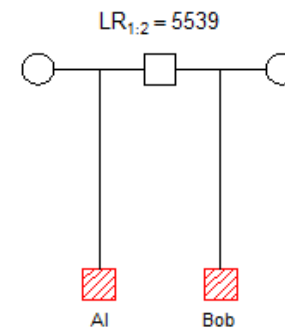
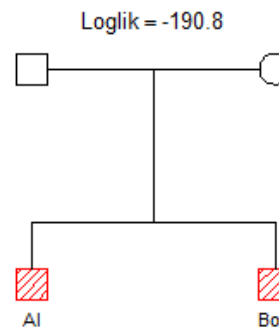
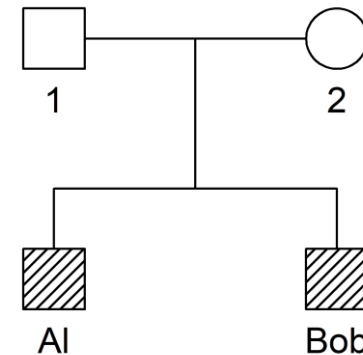
...

Building pedigree list:

...

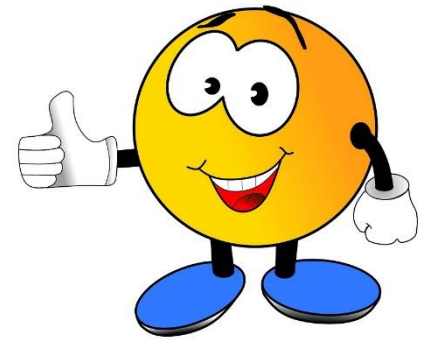
Computing the likelihood of 6 pedigrees.

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

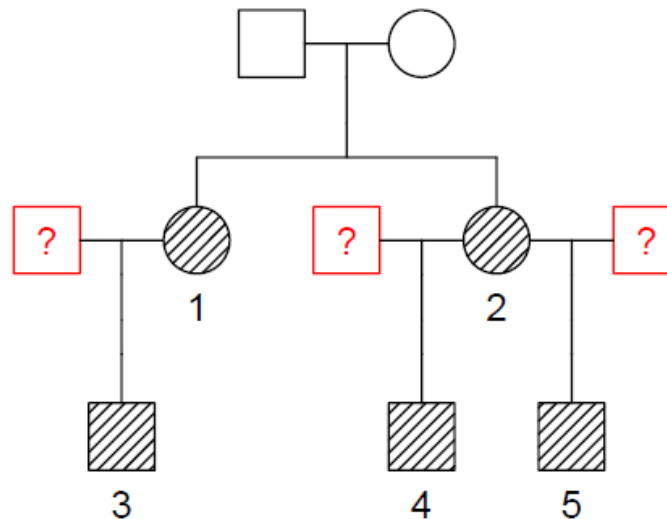


# Optional parameters for restricting the search space

- **extra**: The max number of connecting individuals
  - default: **extra** = "parents" (suitable for small datasets)
- **maxInbreeding**: Default: 1/16 (e.g., first cousins)
- **age**: A vector of (relative) ages OR age inequalities, e.g. "Al > Bob"
- **inferPO**: If TRUE, an initial stage of pairwise IBD estimation is done
- **knownPO**: Known parent–offspring pairs
- **allKnown**: Is **knownPO** the *complete* list of POs?
- **notPO**: Pairs known not to be parent–offspring
- **noChildren**: Individuals known to have no children
- **linearInb**: Max incestuous generation gap (default: 0)
- **connected**: Set to FALSE to allow disconnected pedigrees
- **sexSymmetry**: Remove 'symmetric' versions. Default: TRUE



## Your turn: Exercises!



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