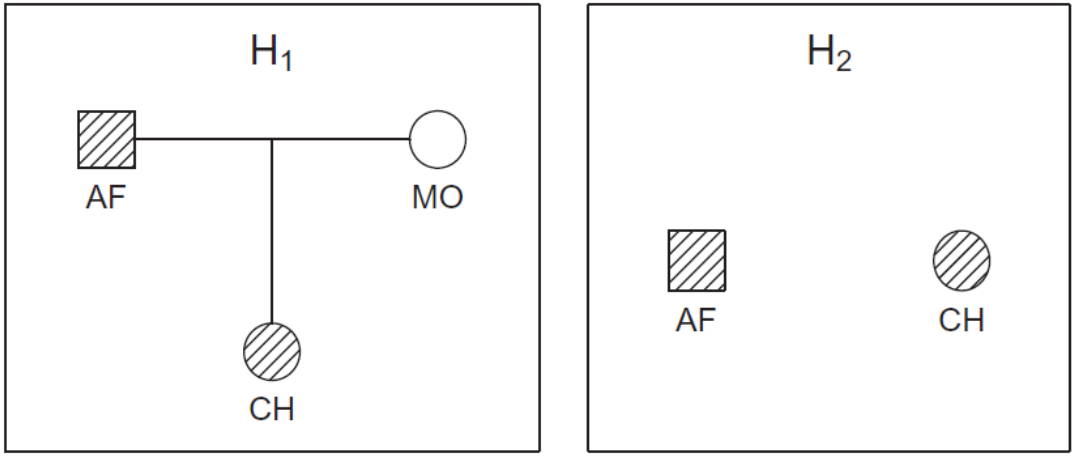
ISFG Summer School - Virtual Edition 2021

Lecture 3. Kinship testing.
LR: paternity cases and complex cases



Thore Egeland
Norwegian University of Life Sciences

Motivating examples

- Paternity testing
- Identification of missing persons
- Disaster victim identification
- Legal cases, like inheritance disputes and applications for immigration
- Medical genetics, for example, quality control in family-based analysis
- Historic cases, like the Romanov family
- Plant and wildlife research

- We distinguish between
  - *kinship testing*, current topic, where a specific set of alternatives are compared, and
  - *relatedness inference* aiming to find the most probable relationship without restrictions

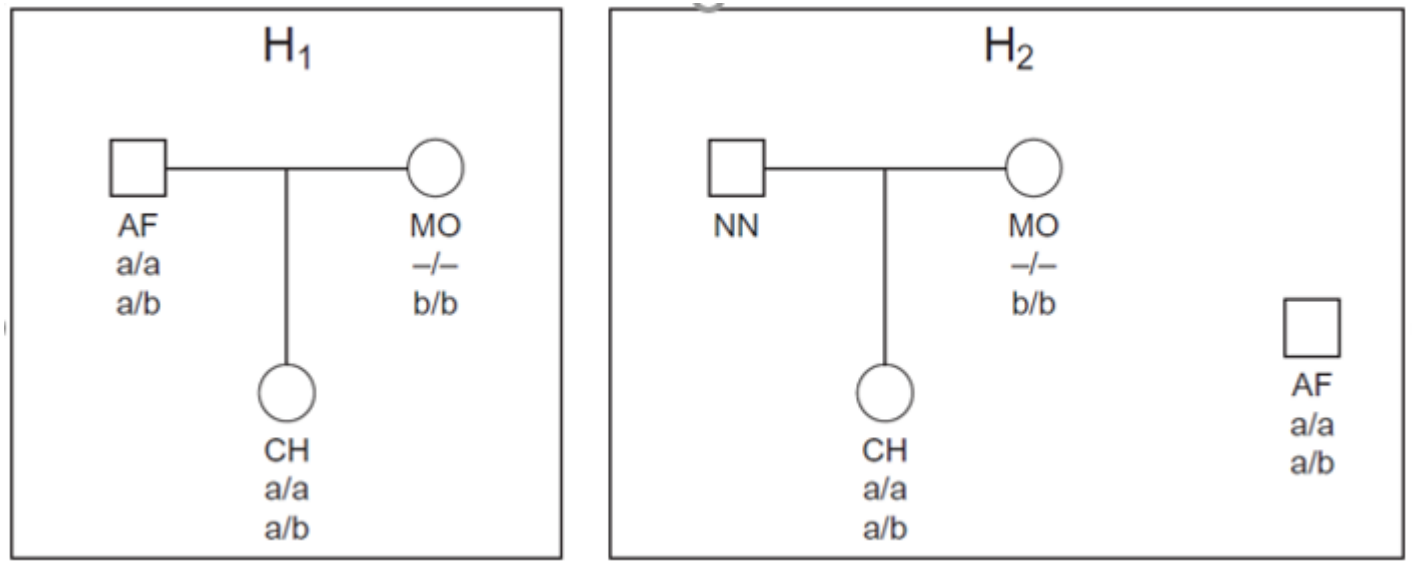
The Likelihood Ratio (LR)

- H_1: The individuals are related according to some pedigree $\mathcal{P}_1$.
- H_2: The individuals are related according to a different pedigree $\mathcal{P}_2$.

$$\mathrm{LR} = \frac{P(\text{data} \mid H_1, \Theta)}{P(\text{data} \mid H_2, \Theta)}.$$

- data: available genotypes
- Θ fixed model parameters common to both hypotheses. Omitted in notation
- Interpretation: The LR says how much better the explanation for the data offered by H_1 is, compared to the explanation offered by H_2

- Default assumptions (can be relaxed):
  - ✓ Hardy Weinberg Equilibrium
  - ✓ No mutations
  - ✓ No artefacts (drop out, drop in, genotyping error)
  - ✓ Independence between markers

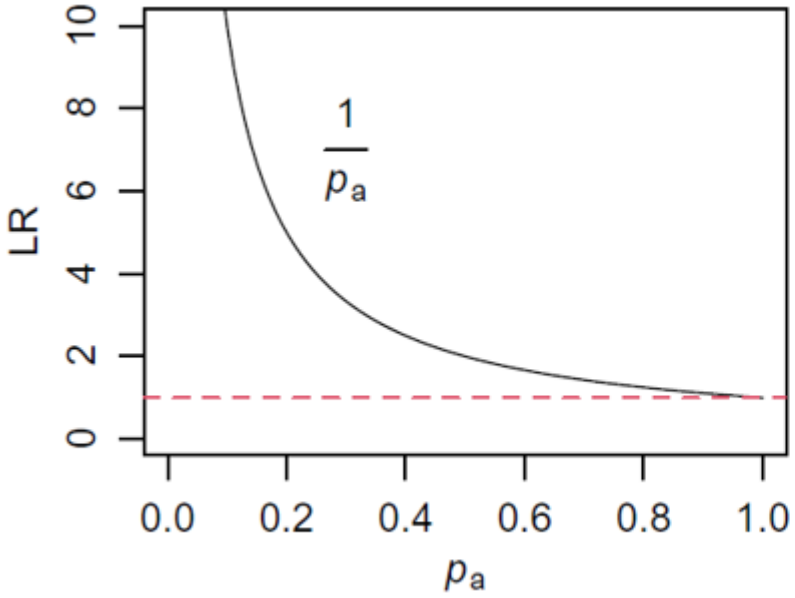


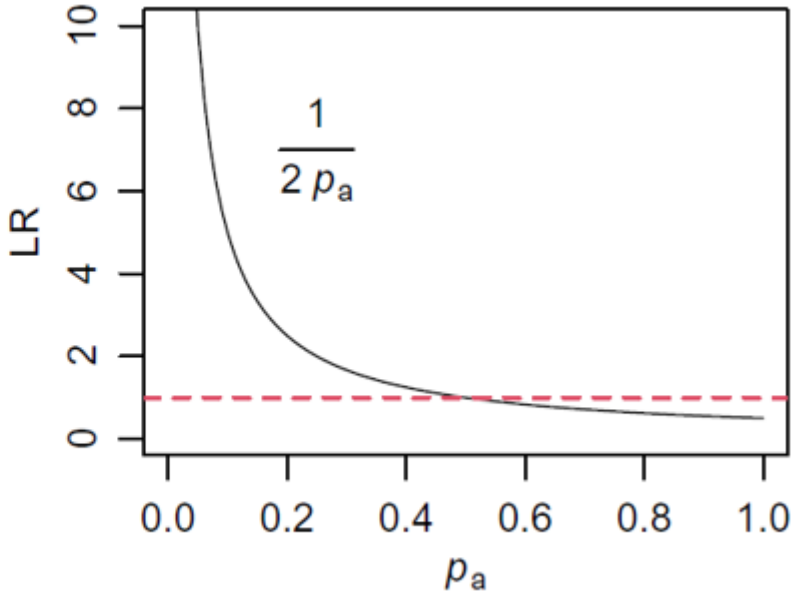
- LR for first and second marker

$$\mathrm{LR} = \frac{P(\mathrm{AF} = \mathrm{a/a},\ \mathrm{CH} = \mathrm{a/a} \mid H_1)}{P(\mathrm{AF} = \mathrm{a/a},\ \mathrm{CH} = \mathrm{a/a} \mid H_2)} = \frac{p_{\mathrm{a}}^2 \cdot p_{\mathrm{a}}}{p_{\mathrm{a}}^2 \cdot p_{\mathrm{a}}^2} = \frac{1}{p_{\mathrm{a}}}$$

$$\mathrm{LR} = \frac{P(\mathrm{AF} = \mathrm{a/b},\ \mathrm{MO} = \mathrm{b/b},\ \mathrm{CH} = \mathrm{a/b} \mid H_1)}{P(\mathrm{AF} = \mathrm{a/b},\ \mathrm{MO} = \mathrm{b/b},\ \mathrm{CH} = \mathrm{a/b} \mid H_2)} = \frac{2p_{\mathrm{a}}p_{\mathrm{b}} \cdot p_{\mathrm{b}}^2 \cdot \frac{1}{2}}{2p_{\mathrm{a}}p_{\mathrm{b}} \cdot p_{\mathrm{b}}^2 \cdot p_{\mathrm{a}}} = \frac{1}{2p_{\mathrm{a}}}.$$

- $p_a = 0.05$ gives a total LR of (1/0.05)*(1/(2*0.05)) = 200.
  Conclusion: Data 200 times more likely assuming H_1 rather than H_2





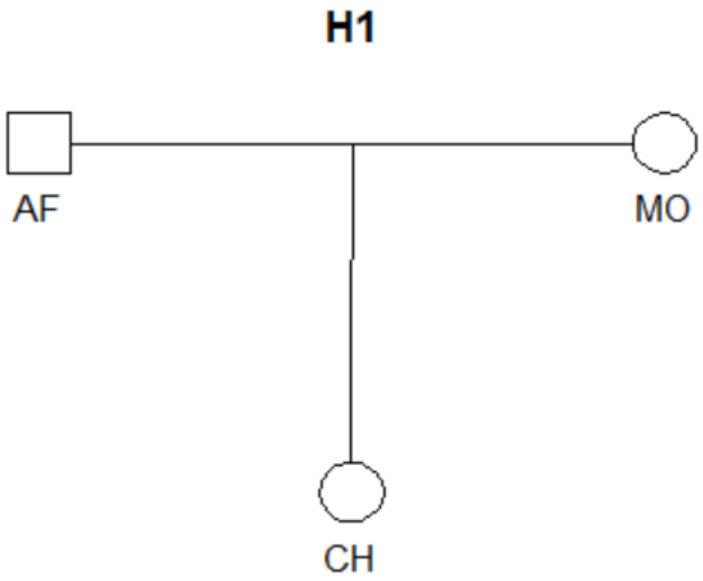
- Observe
  - ✓ LR < 1 if p_a > 0.5 in right panel! Why?

Kinship testing using forrel

i. Create pedigrees representing the hypotheses.
ii. Attach the given genotype data to one of the pedigrees.
iii. Invoke the function kinshipLR() to calculate LRs.

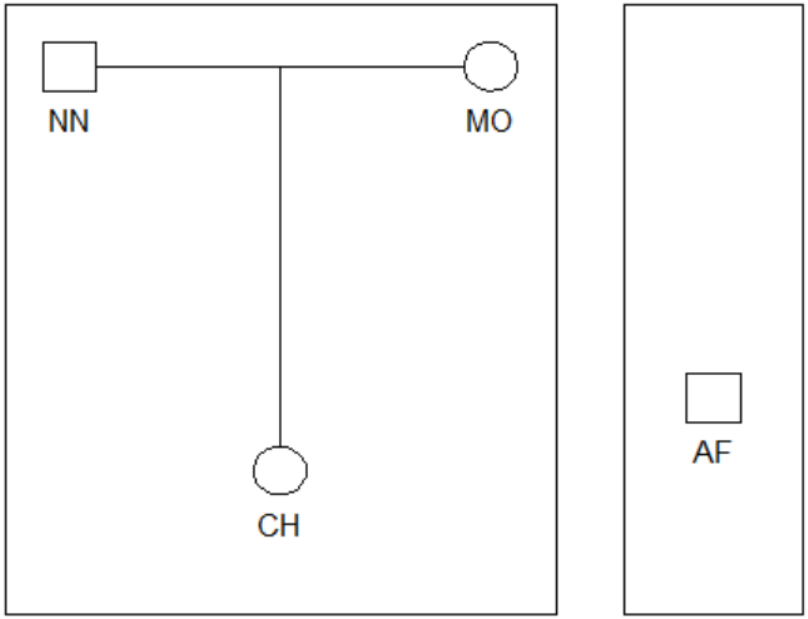
i. Create pedigrees. H1

```
> library(forrel)
> H1 = nuclearPed(fa = "AF", mo = "MO", child = "CH", sex = 2)
> plot(H1, title = "H1")
```



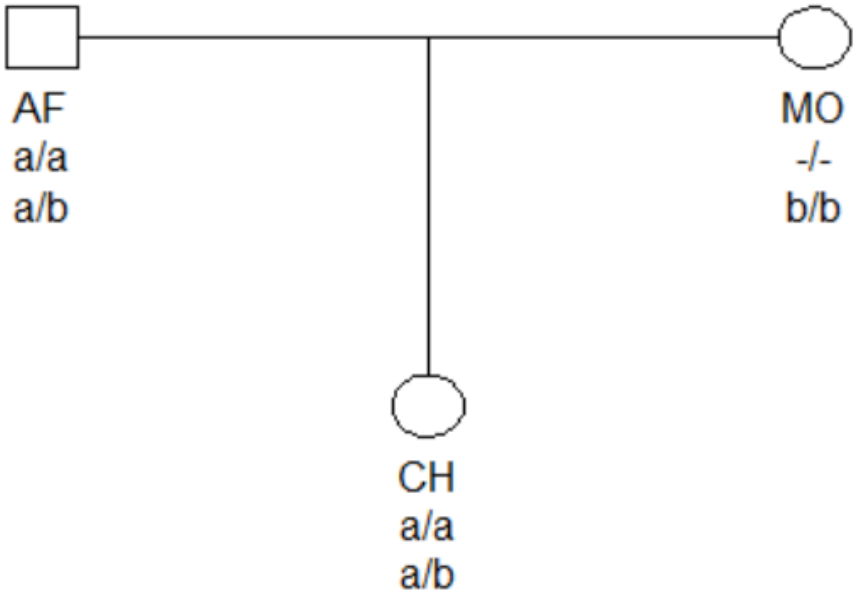
i. Create pedigrees. H2

```
> H2 = list(nuclearPed(fa = "NN", mo = "MO", child = "CH", sex = 2),
>           singleton("AF"))
> plotPedList(H2)
```



ii. Attach genotype data to one of the pedigrees.

```
> afr = c(a = 0.05, b = 0.95)
> m1 = marker(H1, AF = "a/a", CH = "a/a", afreq = afr)
> m2 = marker(H1, AF = "a/b", MO = "b/b", CH = "a/b", afreq = afr)
> H1 = setMarkers(H1, list(m1, m2))
> plot(H1, marker = 1:2)
```



iii. kinshipLR() documentation

```
> ?kinshipLR
```

kinshipLR {forrel} R Documentation

Likelihood ratios for kinship testing

Description

This function computes likelihood ratios (LRs) for a list of pedigrees. One of the pedigrees (the last one, by default) is designated as 'reference', to be used in the denominator in all LR calculations. To ensure that all pedigrees use the same data set, one of the pedigrees may be chosen as 'source', from which data is transferred to all the other pedigrees.

Usage

```
kinshipLR(
  ...,
  ref = NULL,
  source = NULL,
  markers = NULL,
  linkageMap = NULL,
  verbose = FALSE
)
```

Not discussed

iii. Invoke the function kinshipLR() to calculate LRs

```
> lr = kinshipLR(H1, H2, source = 1)
> lr
```

```
H1:H2 H2:H2
  200     1
```

```
> lr$LRperMarker
```

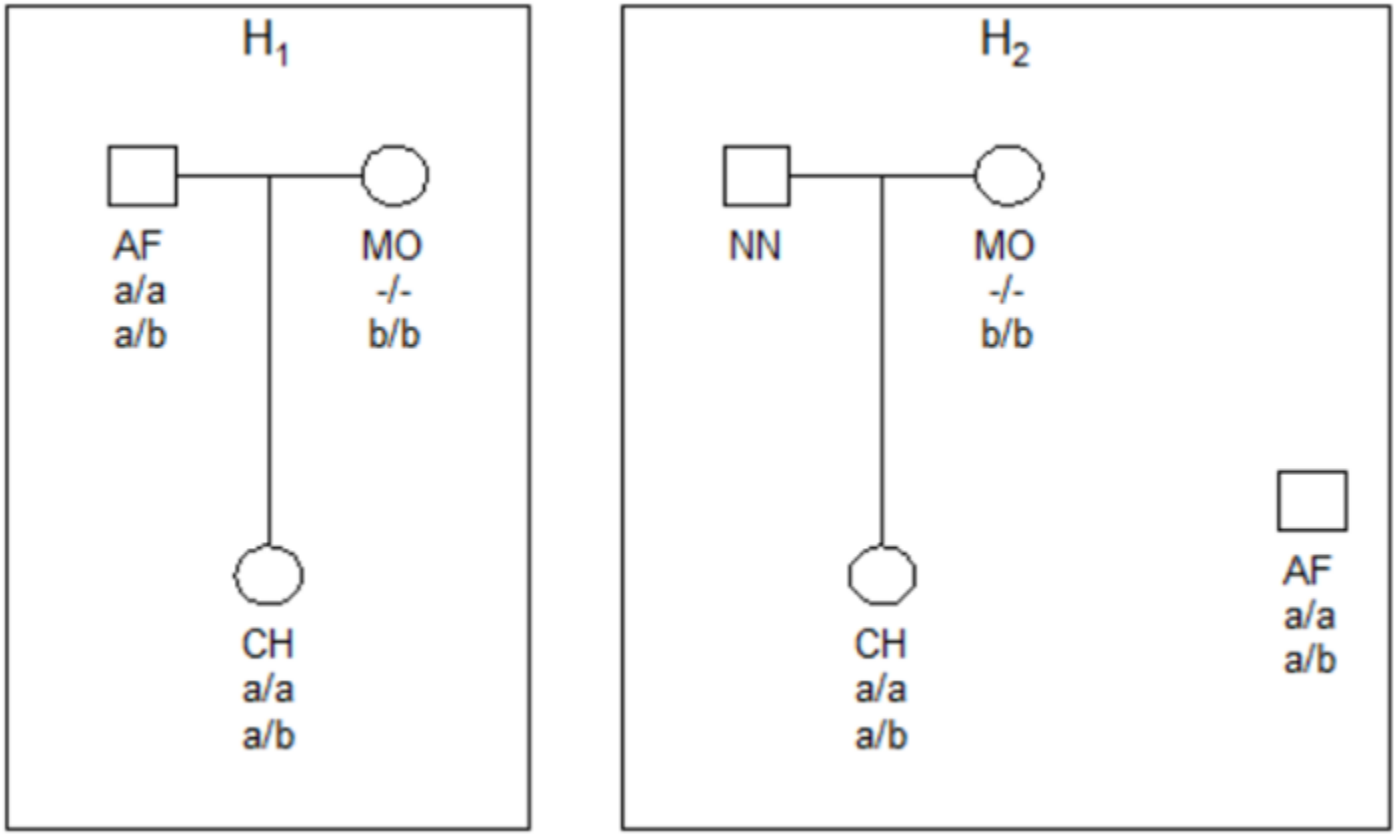
```
    H1:H2 H2:H2
<1>    20     1
<2>    10     1
```

Case code summarised

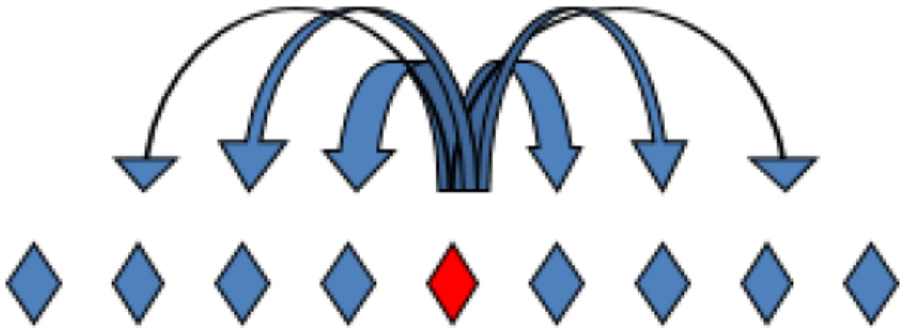
```
> library(forrel)

> H1 = nuclearPed(fa = "AF", mo = "MO", child = "CH", sex = 2)
> H2 = list(nuclearPed(fa = "NN", mo = "MO", child = "CH", sex = 2),
            singleton("AF"))

> afr = c(a = 0.05, b = 0.95)
> m1 = marker(H1, AF = "a/a", CH = "a/a", afreq = afr)
> m2 = marker(H1, AF = "a/b", MO = "b/b", CH = "a/b", afreq = afr)
> H1 = setMarkers(H1, list(m1, m2))

> kinshipLR(H1, H2, source = 1)
```

Case summarised in figure

```
> titles = c(expression(H[1]), expression(H[2]))
> plotPedList(list(H1, H2), titles = titles, marker = 1:2,
>             source = 1, cex = 1.2)
```

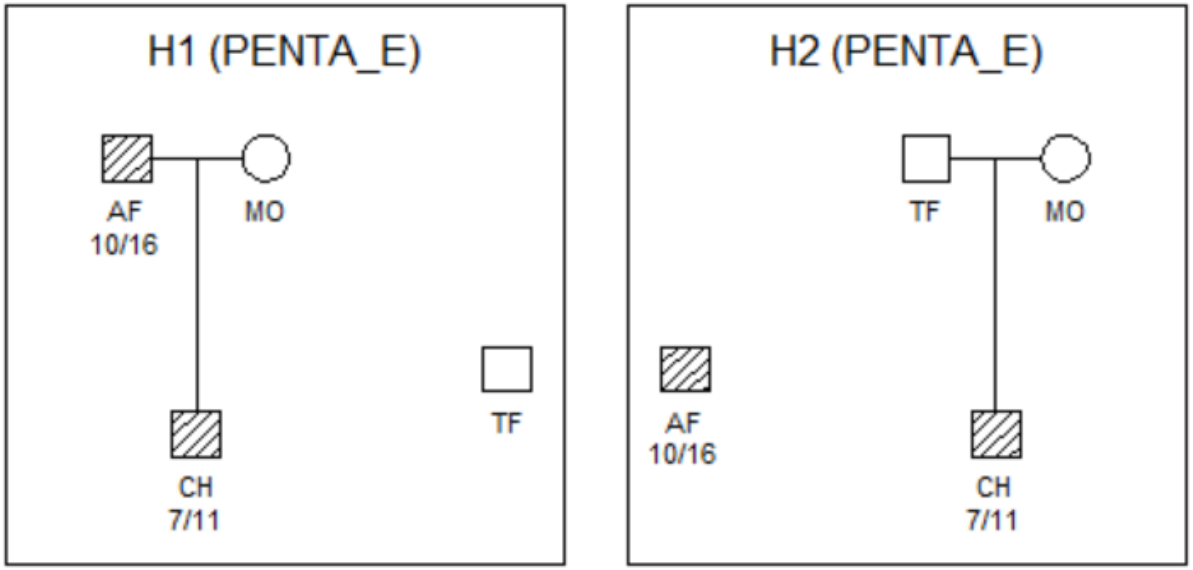


Mutations. Models



- Mutation rates higher in males.
- Short mutations more likely: One step mutation more likely than two steps and so on.
- Mutation rates:
  `http://www.cstl.nist.gov/strbase/mutation.htm`

Dealing with mutations

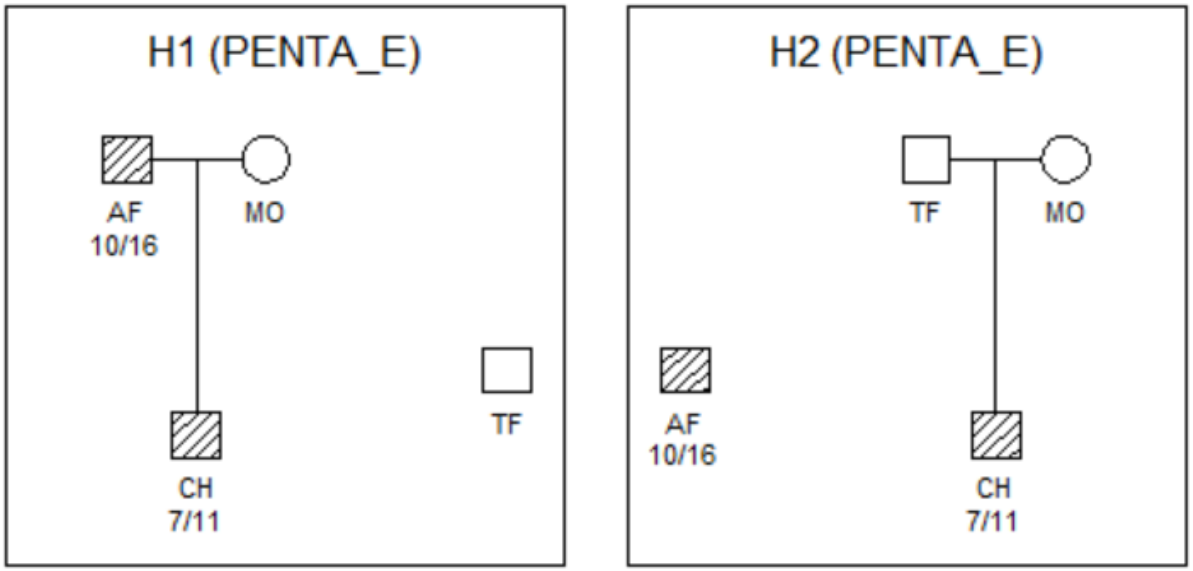


Strategies for handling mutations

i. Exclude inconsistent markers from the analysis. Not recommended
ii. Apply mutation modelling only to inconsistent markers
iii. Apply mutation modelling to *all* markers . Recommended

Read data and compute LR

```
> ?readFam
```




```
# Read data from Familias file and find LR
> filename = "http://familias.name/norbisRelatedness/paternityCase.fam"
> dat = readFam(filename)
> H1  = dat[[1]]
> H2  = dat[[2]]
> lr1 = kinshipLR(H1, H2)
> lr1

H1:H2 H2:H2
    0     1
```

Inspect each marker

```
> lr1$LRperMarker
```

	H1:H2
D3S1358	2.466752
TH01	1.194605
D21S11	1.095934
D18S51	2.153261
PENTA_E	0.000000
D5S818	1.406127
D13S317	4.041611
D7S820	1.433570
D16S539	8.312297
CSF1PO	2.024678
PENTA_D	11.989252
VWA	5.565000
D8S1179	9.650567
TPOX	1.787652
FGA	2.956394
D12S391	2.183522
D1S1656	3.333333
D2S1338	3.147060
D22S1045	26.748152
D2S441	1.445948
D19S433	3.343766

Mutation models

```
> ?setMutationModel
```

setMutationModel {pedprobr} R Documentation

Set a mutation model

Description

This function offers a convenient way to attach mutation models to a pedigree with marker data. It wraps `pedmut::mutationModel()`, which does the main work of creating the models, but relieves the user from having to loop through the markers in order to supply the correct alleles and frequencies for each marker.

Usage

```
setMutationModel(x, model, markers = NULL, ...)
```

Details

Currently, the following models are implemented in the `pedmut` package:

- `equal` : All mutations equally likely; probability *1-rate* of no mutation
- `proportional` : Mutation probabilities are proportional to the target allele frequencies
- `onestep`: A mutation model for microsatellite markers, allowing mutations only to the nearest neighbours in the allelic ladder. For example, '10' may mutate to either '9' or '11', unless '10' is the lowest allele, in which case '11' is the only option. This model is not applicable to loci with non-integral microvariants.
- `stepwise`: A common model in forensic genetics, allowing different mutation rates between integer alleles (like '16') and non-integer "microvariants" like '9.3'). Mutations also depend on the size of the mutation if the parameter 'range' differs from 1.
- `custom` : Allows any mutation matrix to be provided by the user, in the `matrix` parameter

Recompute with mutation model

```
> library(pedsuite)
> H2 = setMutationModel(H2, model = "proportional",
       rate = 0.00001)
> lr2 = kinshipLR(H1, H2, ref = 2, source = 2)
> lr2
```

```
   H1:H2    H2:H2
107132.1      1.0
```

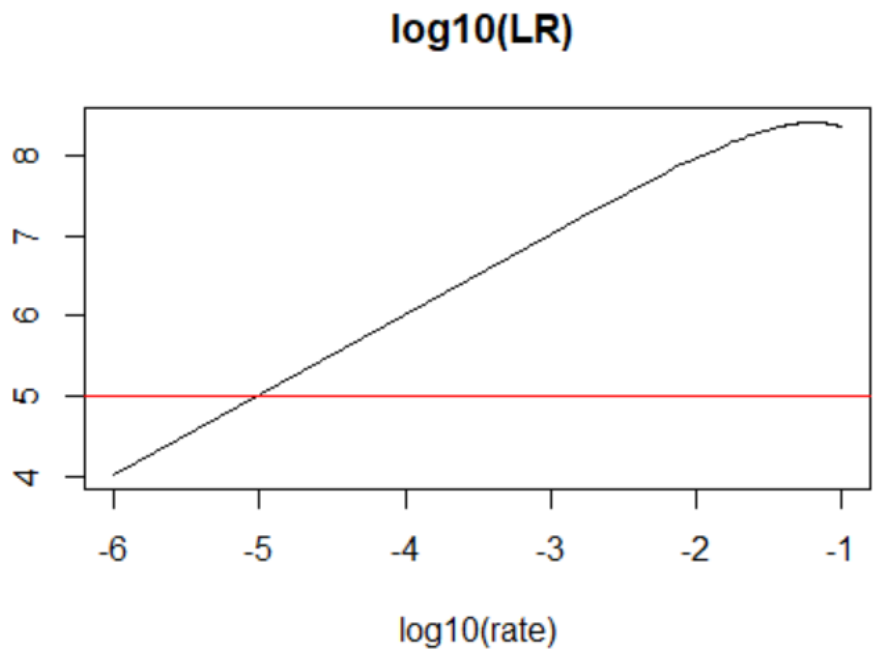
Inspect each marker again

```
> lr2$LRperMarker
```

	H1:H2
D3S1358	2.466733e+00
TH01	1.194603e+00
D21S11	1.095933e+00
D18S51	2.153248e+00
PENTA_E	1.114807e-05
D5S818	1.406121e+00
D13S317	4.041573e+00
D7S820	1.433564e+00
D16S539	8.312203e+00
CSF1PO	2.024664e+00
PENTA_D	1.198912e+01
VWA	5.564943e+00
D8S1179	9.650459e+00
TPOX	1.787639e+00
FGA	2.956371e+00
D12S391	2.183508e+00
D1S1656	3.333307e+00
D2S1338	3.147035e+00
D22S1045	2.674780e+01
D2S441	1.445941e+00
D19S433	3.343736e+00

Sensitivity plot. More advanced code

```
>  rvec = seq(0.000001, 0.1, length = 100)
>  lrvec = sapply(rvec, function(r) {
>    H2 = setMutationModel(H2, model = "prop", rate = r)
>    lr = kinshipLR(H1, H2, source = 2)
>    lr$LRtotal[1]
     })
>  plot(log10(rvec), log10(lrvec), type = "l",
       xlab = "log10(rate)", main = " log10(LR)")
>  abline(h = 5, col = "red")
```



- If mutation rate > 0.00001 (log10(0.00001) = -5), LR > 10^5

A closer look at the impact of mutation

```
> lrNoMut = lr1$LRperMarker[,1]
> lrMut = lr2$LRperMarker[,1]
> ratio = lrMut/lrNoMut
> tab = data.frame(lrNoMut, lrMut, ratio)
> round(tab,5)
```

	lrNoMut	lrMut	ratio
D3S1358	2.46675	2.46673	0.99999
TH01	1.19461	1.19460	1.00000
D21S11	1.09593	1.09593	1.00000
D18S51	2.15326	2.15325	0.99999
PENTA_E	0.00000	0.00001	Inf
D5S818	1.40613	1.40612	1.00000
D13S317	4.04161	4.04157	0.99999
D7S820	1.43357	1.43356	1.00000
D16S539	8.31230	8.31220	0.99999
CSF1PO	2.02468	2.02466	0.99999
PENTA_D	11.98925	11.98912	0.99999
VWA	5.56500	5.56494	0.99999
D8S1179	9.65057	9.65046	0.99999
TPOX	1.78765	1.78764	0.99999
FGA	2.95639	2.95637	0.99999
D12S391	2.18352	2.18351	0.99999
D1S1656	3.33333	3.33331	0.99999
D2S1338	3.14706	3.14703	0.99999
D22S1045	26.74815	26.74780	0.99999
D2S441	1.44595	1.44594	1.00000
D19S433	3.34377	3.34374	0.99999

- ratio = lrMut/lrNoMut ≈ 1 for consistent markers if mutation rate is close to 0. Why?
- Dubious practice:
  modify model to fit data

  Hence
- Recommendation:
  - Apply mutation model to all markers

A Relationship Riddle

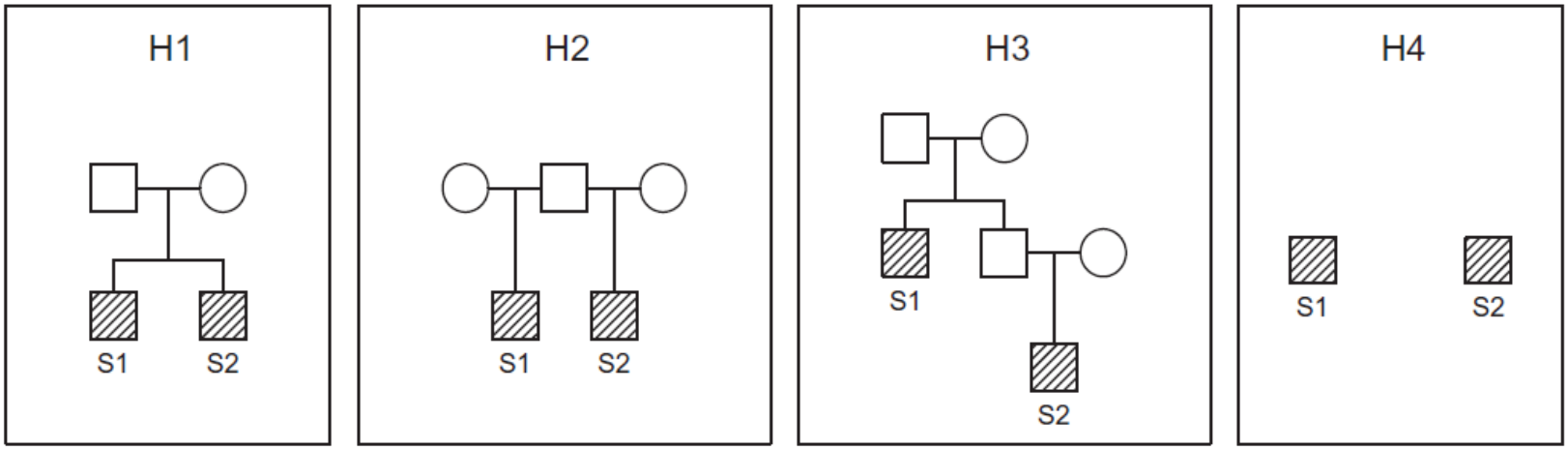


Fig. 6.4 A relationship riddle: Four hypothesised relationships between S1 and S2.

- H_1: Full brothers
- H_2: Half-brothers
- H_3: Uncle and nephew
- H_4: Unrelated

Data sets

- Next example and some exercises need data files
- Data are made available by running

```
> url = "https://magnusdv.github.io/pedinr/datasets/data.zip"
> download.file(url, destfile = "data.zip")
> unzip("data.zip")
```

R code

i. Define first three pedigrees:

```
> ids = c("S1", "S2")
> H1 = nuclearPed(children = ids)
> H2 = relabel(halfSibPed(), old = 4:5, new = ids)
> H3 = relabel(cousinPed(deg = 0, rem = 1),
                old = c(3,6), new = ids)
```

ii. Read marker data, also adds H4, assign database

```
> H4 = readPed("data/kinship-riddle.ped")
> H4 = setFreqDatabase(H4,
     database = NorwegianFrequencies)
```

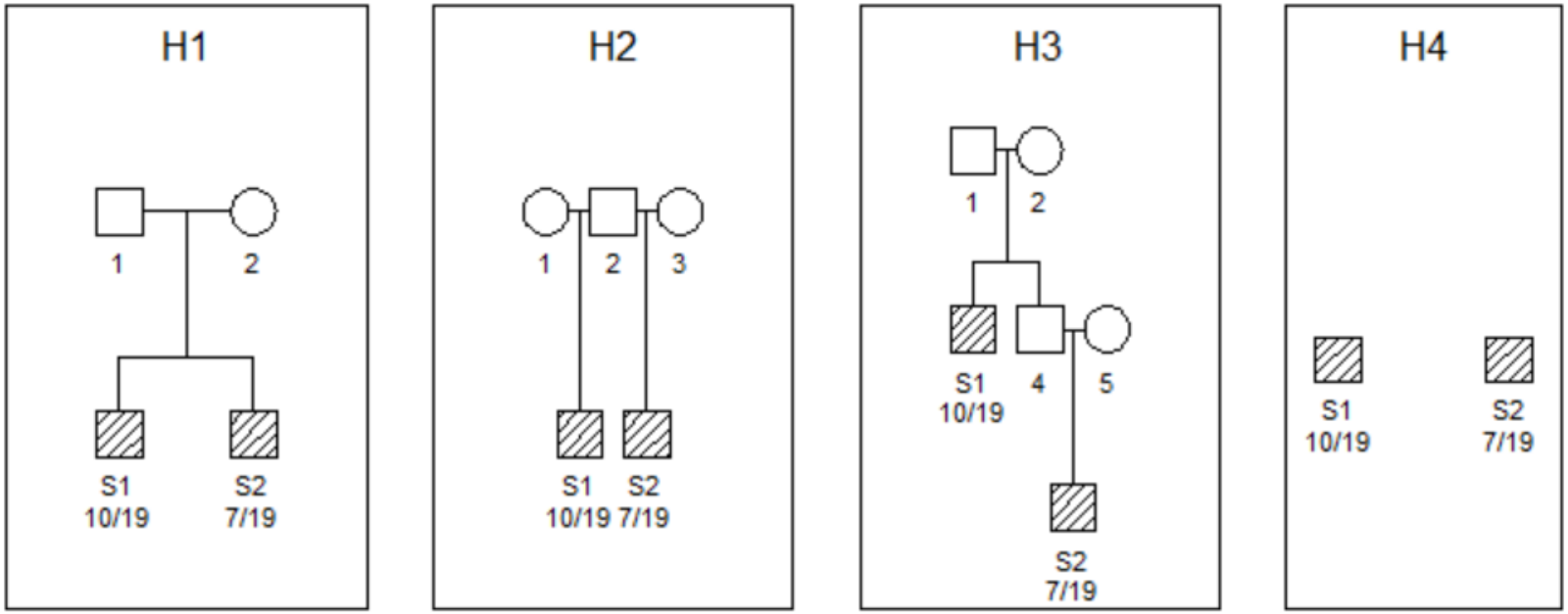
iii. Find LR

```
> lr = kinshipLR(H1, H2, H3, H4, verbose = TRUE, ref = 4)
```

```
Reference pedigree:  4
Source pedigree:  4
Number of markers:  15
   H1:H4    H2:H4    H3:H4    H4:H4
569.3989 805.1175 805.1175   1.0000
```

```
> lr$LRperMarker
```

```
##              H1:H4      H2:H4      H3:H4 H4:H4
## D3S1358   0.8626008  1.1126008  1.1126008     1
## TH01      0.6171107  0.8671107  0.8671107     1
## D21S11    3.4238376  3.6738376  3.6738376     1
## D18S51    0.2500000  0.5000000  0.5000000     1
## PENTA_E 13.2568407 13.5068407 13.5068407     1
## D5S818    0.8941534  1.1441534  1.1441534     1
## D13S317   1.3073685  1.5573685  1.5573685     1
## D7S820    6.2033366  2.4906498  2.4906498     1
## D16S539   0.2500000  0.5000000  0.5000000     1
## CSF1PO    8.5203211  2.5445854  2.5445854     1
## PENTA_D   3.8396894  1.6137480  1.6137480     1
## VWA       1.4611460  1.7111460  1.7111460     1
## D8S1179   0.8293633  1.0793633  1.0793633     1
## TPOX      0.7502601  1.0002601  1.0002601     1
## FGA       1.7481264  1.9981264  1.9981264     1
```

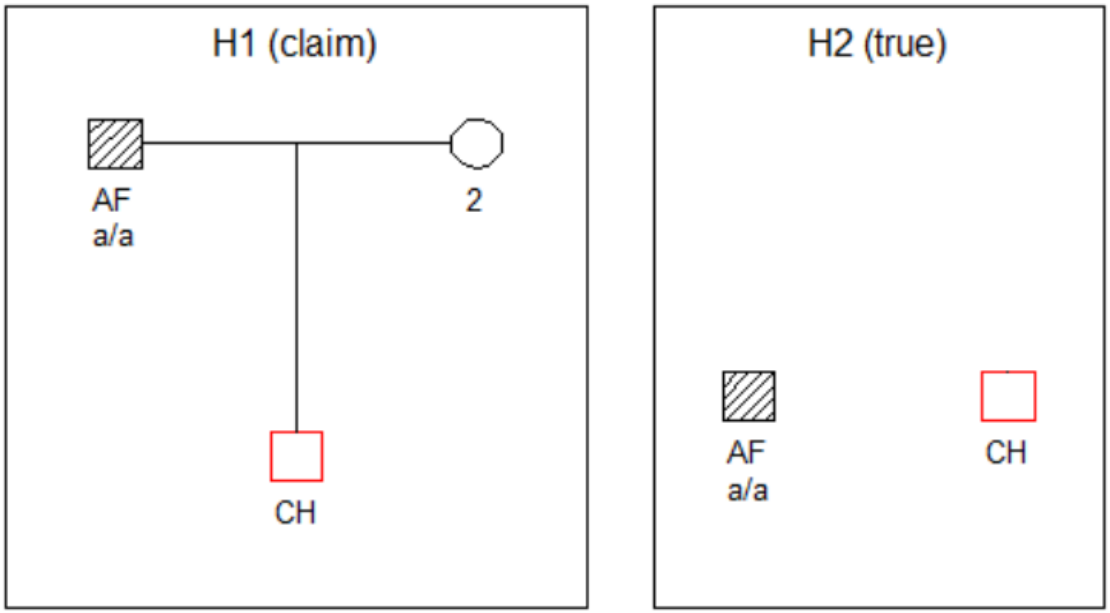



```
> afreq(H4, marker = "PENTA_E")["19"]

          19
0.009610328
```

- Large LR for PENTA_E: Frequency of allele 19 low and so unlikely to be shared by unrelated individuals

Exclusion Power

- The *exclusion power* (EP) of a kinship test is the probability that H_1 ('claim') can be excluded, given that H_2 is true



$$\text{EP} = P(\text{data incompatible with } H_1 \mid H_2)$$

$$= (1 - p_a)^2 = (1 - 0.1)^2 = 0.81$$

Exclusion power using forrel

- The general function is
  - exclusionPower(claimPed, truePed, ids)
- If H_2 (true) is 'unrelated', we can use the simpler
  - randomPersonEP(claimPed, id)

```
> H1  = nuclearPed(fa = "AF", child = "CH")
> afr = c(a = 0.1, b = 0.9)
> m1  = marker(H1, AF = "a/a", afreq = afr)
> H1  = setMarkers(H1, m1)
> randomPersonEP(H1, "CH", verbose = F)
```

```
Potential mismatches: 1 (1)
Expected mismatches: 0.81
P(at least 1 mismatch): 0.81
```

Summary

- Some omitted topics
  - Simulation: markerSim, profileSim,…(Lecture 6)
  - Power: LRpower
  - Theta correction
  - Linkage, linkage equilibrium, X-chromosomal markers
  - Posterior probalities (Lecture 6)
- Advantages of R
  - Plotting. Familias also uses R for plotting
  - Loops
  - Extensions