Statistics of forensic lineage DNA markers

Mikkel Meyer Andersen (mikl@math.aau.dk)

Dept of Mathematical Sciences
Aalborg University, Denmark

Apr 18, 2018 @ Dept of Stats, U of Auckland, NZ

Outline

- Introduction
  - Forensic statistics
  - DNA and lineage markers
- Statistical models w/ computational aspects
  - Classical model with parameter estimation
  - Simulation based model

Forensic statistics

Example

- Crime committed
- Victim: "Culprit has blue eyes"
- Suspect apprehended
  - Brown eyes? Not him.
  - Blue eyes? What now?

Evidential weight

Weight of the evidence (likelihood ratio):

$$LR = \frac{P(E \mid H_p)}{P(E \mid H_d)},$$

- E is the evidence (e.g. DNA profile from crime scene)
- H_p (prosecutor's hypothesis) is "the suspect is the donor of the evidence from the crime scene"
- H_d (defence attorney's hypothesis) is "the suspect is unconnected to the crime"

Simplifications:

- $P(E \mid H_p)$ often assumed equal to 1
- $P(E \mid H_d)$: Match probability $\approx$ match by chance $\approx$ "How probable it is that some random man matches the evidence found at the crime scene?" (population frequency)

Example (cont.)

- $LR = \frac{P(E|H_p)}{P(E|H_d)}$
- Victim: "Culprit has blue eyes"
- Suspect apprehended: Blue eyes.
- $P(E \mid H_p) = P(E = \text{'blue eyes'} \mid H_p = \text{'culprit = suspect'})$
  - $P(E \mid H_p) = 1$ (for the sake of this example)
  - Suspect non blue eyes: $P(E \mid H_p) = 0$
- $P(E \mid H_d) = P(E = \text{'blue eyes'} \mid H_d = \text{'culprit} \neq \text{suspect'})$:
  - $P(E \mid H_d)$? "suspect did not do it" $\approx$ "a random man from the population did it"
  - $p_{\text{blue}} = 0.2$

Thus:

$$LR = \frac{P(E \mid H_p)}{P(E \mid H_d)} = \frac{1}{p_{\text{blue}}} = \frac{1}{0.2} = 5.$$

Example (cont.)

$$LR = 5.$$

"It is 5 times more likely to observe the evidence from the crime scene if the suspect left it than if a random individual from the population left it."

- Conditional probabilities are difficult: Prosecutor's fallacy ("given the evidence, it is 5 times more likely that the suspect did it than a random individual did it")
- Posterior odds can be obtained:

$$\underbrace{\frac{P(H_p \mid E)}{P(H_d \mid E)}}_{\text{Posterior odds}} = \underbrace{\frac{P(E \mid H_p)}{P(E \mid H_d)}}_{LR} \times \underbrace{\frac{P(H_p)}{P(H_d)}}_{\text{Prior odds}}$$

DNA profiles

Biological traces

- Example: eye color; often not available nor scientific (eye witnesses)
- Biological trace often found (hair, blood, saliva, . . .)
  - "Non-coding" part
    - Forensic DNA profiles
  - Coding part
    - (Research on physical appearance from biological traces)

DNA

- Bases: A-T and C-G
- 3.3 billion (10^9) base pairs
  - 23 chromosome pairs
    - One inherited from mother and one from father
  - 22 autosomes (non-sex chromosomes): recombination
  - 1 allosome (sex chromosome)
    - Girl: 23rd pair: XX
    - Boy: 23rd pair: XY
    - 23rd: X always inherited from mother, father determines sex (pass on X or Y)
- Unique: no other humans have the same (monozygotic/identical twins are special, ignore these)

Forensic DNA profiles:

- Subset of the DNA

Short tandem repeats (STRs)

Terminology of short tandem repeats (STR):

- **Locus** (**loci** in plural): Location at a certain chromosome (e.g. D3S1358, DYS391)
- **Allele**: The number of times a *motif* (short sequence of 3-5 base pairs) repeats itself
  - An example of an allele of 3:

$$\underbrace{AGAT}_{motif}\ AGAT\ AGAT = [AGAT]_3$$

  - STR's can mutate ($\mu \approx 0.003$) during meiosis causing variation (e.g. $11 \to 10$)
- Allele: An integer (for simplicity)
- DNA profile consists of 10-30 STR loci

Types of DNA profiles

- Traditional DNA profiles
  - Based on autosomal (non-sex) chromosomes
  - Example of autosomal STR DNA profile (only three loci shown):

$$\text{D3S1358} = \{15, 18\},\ \text{D5S818} = \{12, 12\},\ \text{D7S820} = \{10, 11\}$$

- Lineage marker DNA profiles
  - Y-STR **haplotype** (*as a whole*): DNA profile from the Y chromosome using STR
    - Inherited from father to boys; hence paternal lineage
    - Example of Y-STR DNA profile (only three loci shown):

$$\text{DYS391} = 10,\ \text{DYS437} = 15,\ \text{DYS635} = 22$$

  - Mitochondrial DNA profiles
    - Independent genome with approx. 16,500 bases (ATGC)
    - Inherited from mother to children; hence maternal lineage
    - "Pick somebody as reference individual" (revised Cambridge Reference Sequence, rCRS), list variations in relation to this sequence

Why?

Why bother using anything else than traditional autosomal STR DNA profiles?

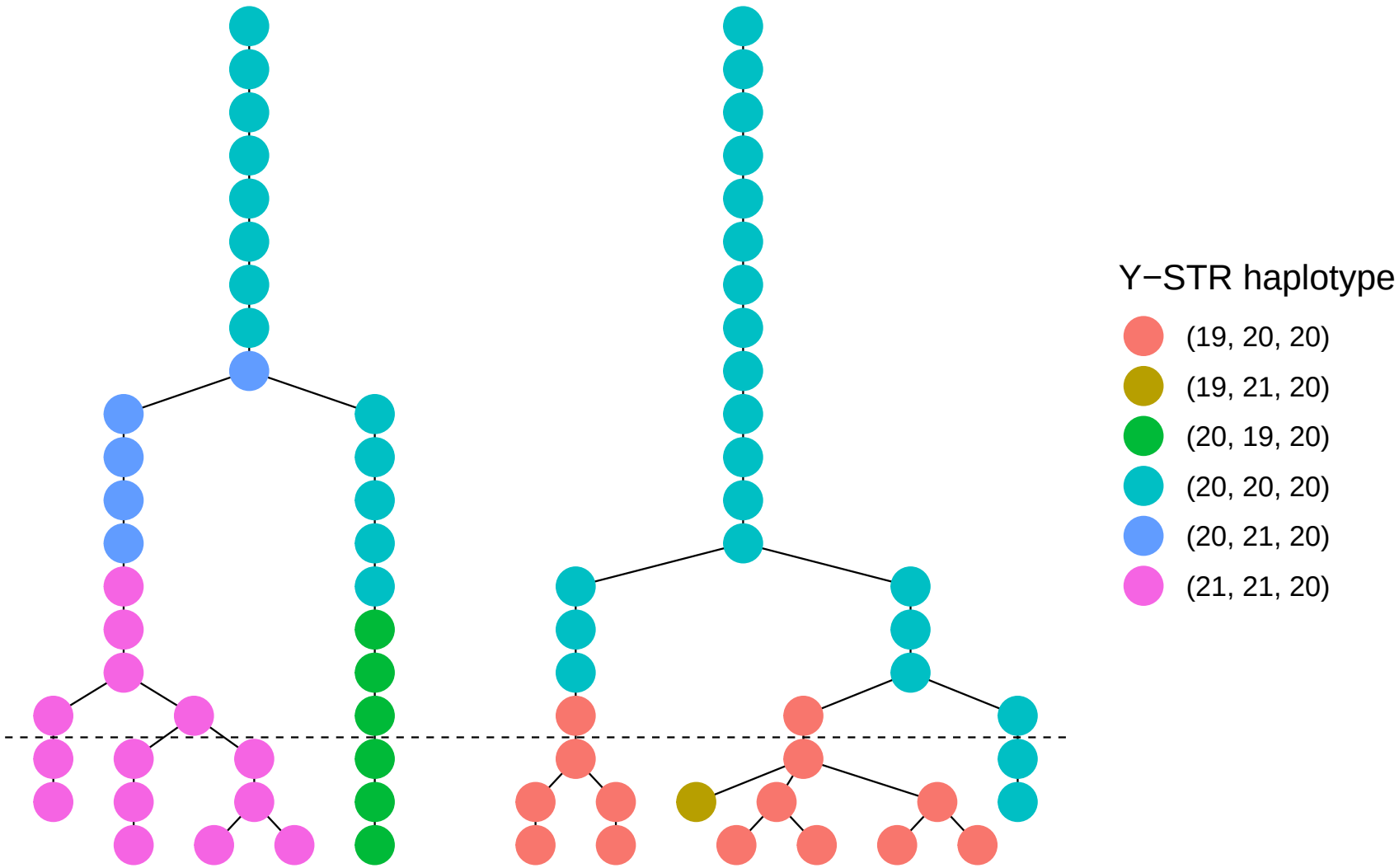
- Y chromosomal DNA profiles
  - Unbalanced mixture of female/male DNA (minor male component masked), e.g. sexual assault cases:
    - Touch DNA / male DNA under the fingernails of a victim
    - Sexual assult without ejaculation or by a vasectomised male
    - Extract (biochemically) Y chromosomal DNA to obtain Y chromosomal DNA profile
- Mitochondrial DNA profiles
  - Degraded DNA (time, weather, environment, . . .)
  - No nuclei: hair shafts

Statistical properties

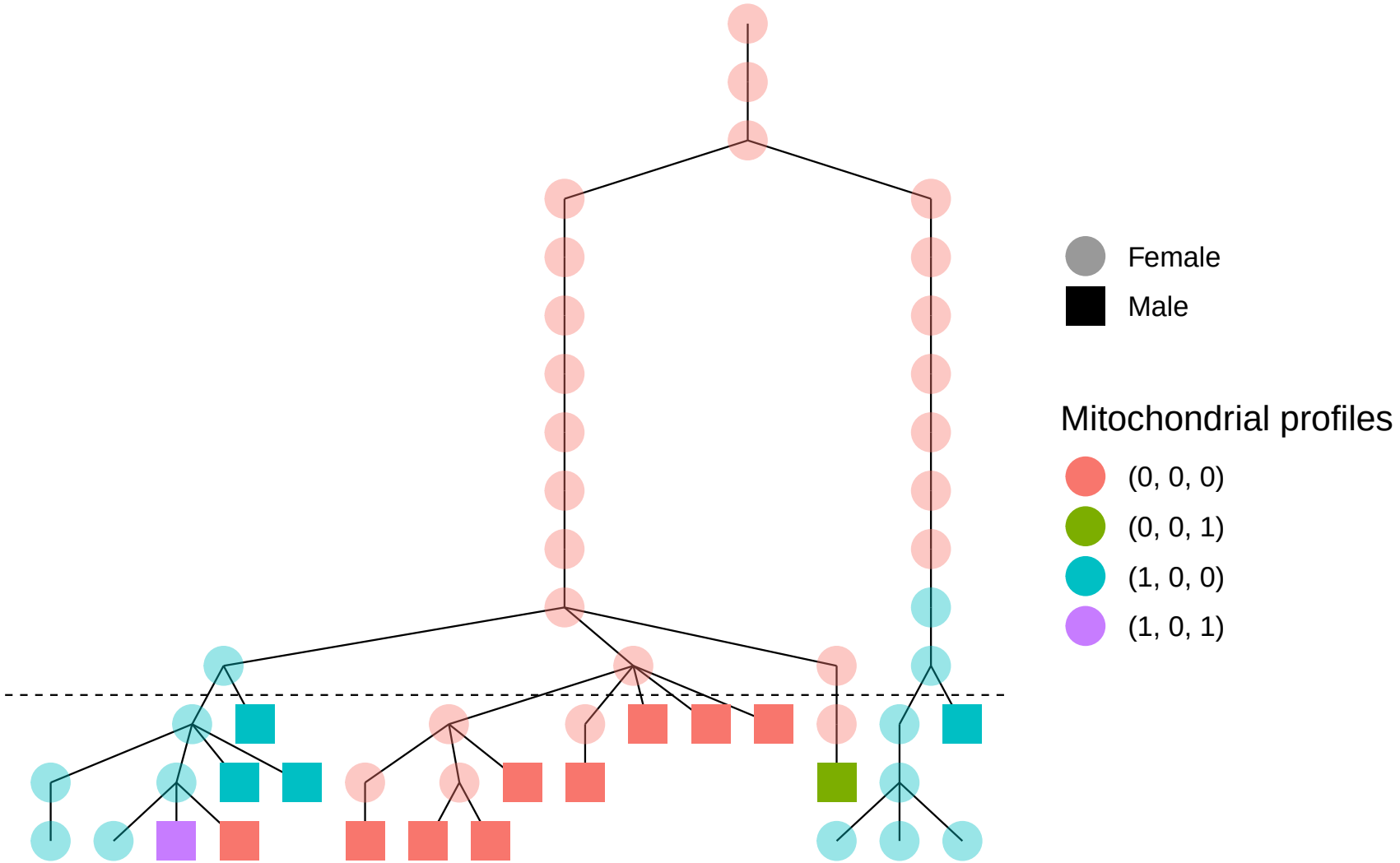
Statistical properties (due to genetic inheritance):

- **Autosomal**: Two alleles per locus inherited independently between and within loci from each parent
  - Widely used and a lot of statistics for that area exist
  - Match probability (grossly simplified) of DNA profile: Product of the allele frequencies at each locus (due to independence)
- **Y chromosomal**/**Mitochondrial**: Inherited as a whole from the father (Ychr)/mother (mito):
  - Strong dependency between loci
  - Match probability of DNA profile: Very different than for autosomal DNA profiles

Y-STR profiles



Mitochondrial profiles



Statistical models

Lineage marker DNA profiles

- Match probability $\approx$ DNA profile frequency
- Count method (works for any trait, e.g. blood type)
  - n: Database (DB) size
  - n_x: Number of times x is observed in the database
  - $P(X = x) = n_x/n$ such that $LR = n/n_x$
- Problem: Singletons (haplotypes only observed once) are common (a lot of rare variants), $> 90\%$ of observed haplotypes are singletons
  - $\sum_{x \in \text{DB}} n_x/n = 1$, hence $P(X = x) = 0$ for $x \notin$ DB
  - $1/n$ over-estimates the match probability for singletons
- Many suggestions
  - A single match probability
  - Population frequencies (probability distribution on **all** haplotypes)

Modelling **all** haplotypes

- A single match probability is "easy"
- All haplotypes must get a probability
  - Other calculations (not shown)
  - Many possibilities: kernel smoothing etc.
  - Support: $\mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N} = \mathbb{N}^L$
  - Even $L = 10$ is a large space (and we now have $L \approx$ 20-30); difficult to learn structure
  - Exploit genetic knowledge
  - Finite population size in simulation studies

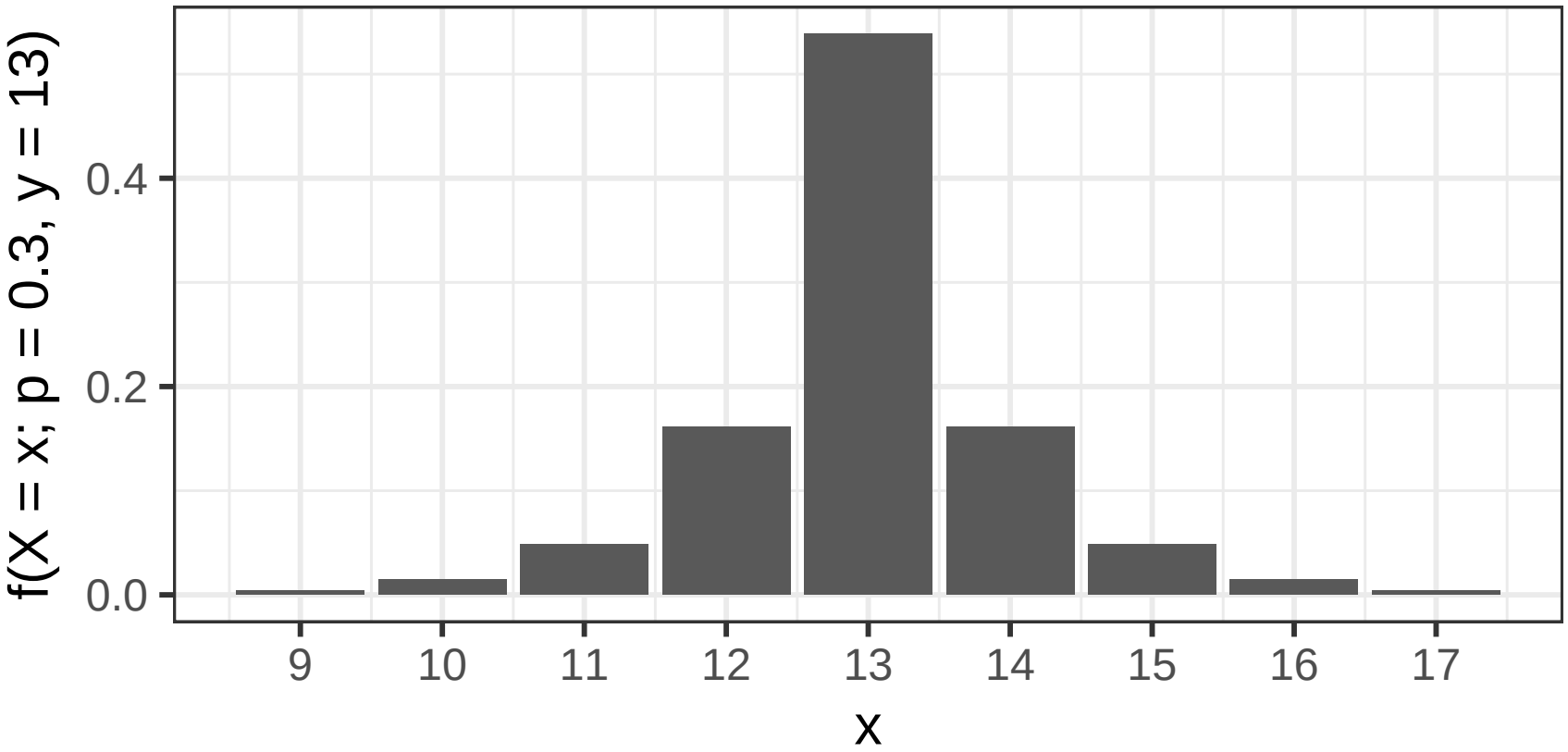
Classical statistical model

The discrete Laplace distribution

- Discrete Laplace distributed $X \sim DL(p, y)$:
  - Dispersion parameter $0 < p < 1$ and
  - Location parameter $y \in \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
- Probability mass function:

$$f(X = x; p, y) = \frac{1-p}{1+p} \cdot p^{|x-y|} \quad \text{for } x \in \mathbb{Z}.$$

The discrete Laplace distribution



- Perfectly homogeneous population with 1-locus haplotypes:

$$P(X = x) = f(X = x; p, y)$$

- 2 parameters instead of (#alleles - 1)
- Support on all integers vs categorical distribution (high dimensions)

The discrete Laplace exponential family

- MM Andersen, PS Eriksen, N Morling. "The discrete Laplace exponential family and estimation of Y-STR haplotype frequencies", 2013.
  - AFAIK: Discrete Laplace exponential family (for fixed, known location parameter, y)
  - $\theta = \log p$ and $d = x - y$:

$$f(d;\theta) = \exp\left(\theta|d| - A(\theta)\right) \quad \text{with } A(\theta) = \log\left(\frac{1+e^{\theta}}{1-e^{\theta}}\right).$$

  - Advantages in parameter estimation
  - `R` library `disclap`
    - {`d`, `p`, `r`}`disclap`
    - `glm(d ~ 1, data, family = DiscreteLaplace())`

Statistical model for Y-STR haplotypes

Multiple loci – r-locus haplotypes (homogeneous population):

$$P(X = (x_1, x_2, \ldots, x_r)) = \prod_{k=1}^{r} f\left(x_k - y_k; p_k\right)$$

- Mutations, $x_k - y_k$, happen independent relative to y; loci, x_k, are not independent
- $y = (y_1, y_2, \ldots, y_r)$: central haplotype (location parameter)
- $p = (p_1, p_2, \ldots, p_r)$: discrete Laplace parameters (one for each locus)

Statistical model for Y-STR haplotypes

Populations are not perfectly homogeneous, multiple subpopulations. Assume c subpopulations and r-locus haplotypes, for $X = (x_1, \ldots, x_r)$:

$$P(X) = \sum_{j=1}^{c} P(X \mid z = j) P(z = j) = \sum_{j=1}^{c} \tau_j \prod_{k=1}^{r} f\left(x_k - y_{jk}; p_{jk}\right)$$

- τ_j: a priori probability for originating from the j'th subpopulation ($\sum_{j=1}^{c} \tau_j = 1$)
- Finite mixture
- EM algorithm
- For known location parameters, `FlexMix` could be used to estimate parameters, but:
  - Family function (`DiscreteLaplace()`) not supported
  - Location parameters are not known, updates necessary
  - More efficient estimation can be done

Discrete Laplace model

$$P(X = (x_1, x_2, \ldots, x_r)) = \sum_{j=1}^{c} \tau_j \prod_{k=1}^{r} f\left(x_k - y_{jk}; p_{jk}\right)$$

$$L_{\text{full}} = \prod_{i=1}^{n} \prod_{j=1}^{c} \prod_{k=1}^{r} \left(\tau_j{}^{1/r} f(x_{ik} - y_{jk}; p_{jk})\right)^{v_{ij}},$$

- Estimate y_{jk}
  - **E**stimate v_{ij} and τ_j
  - **M**aximise L_{full} to get estimate of $\theta_{jk} = \log p_{jk} = \alpha_j + \beta_k$:
    - `glm(d ~ cluster + locus, data, family = DiscreteLaplace(), weights =` v_{ij}`)`

Discrete Laplace model

$$L_{\text{full}} = \prod_{i=1}^{n} \prod_{j=1}^{c} \prod_{k=1}^{r} \left({\tau_j}^{1/r} f(x_{ik} - y_{jk}; p_{jk}) \right)^{v_{ij}},$$

`glm(d ~ cluster + locus, ..., weights =` v_{ij}`)`
Balanced design. One individual:

```
  cluster1 cluster2 locus2 locus3
1        1        .      .      .
2        .        1      .      .
3        1        .      1      .
4        .        1      1      .
5        1        .      .      1
6        .        1      .      1
```

- Design matrix of dimension $(n \cdot c \cdot r) \times (c + r - 1)$

Discrete Laplace model

- GLM/IRLS: $\hat{\beta}^{(t+1)} = \left(X^\top W^{(t)} X\right)^{-1} X^\top W^{(t)} z^{(t)}$ must be calculated
- Calculate $\left(X^\top W^{(t)} X\right)^{-1}$ without constructing X
- Weighted 2-way ANOVA sums + inversion of smaller matrices
- Speed-up: 10x-20x (deviance); 10x-100x ($\hat{\beta}^{(t+1)}$)

Discrete Laplace model

- Statistical model for haplotypes
- Estimates population frequencies
  - Large support vs underestimation
- Finding central haplotypes, y_{jk}, are difficult
- Larger kits (more loci)

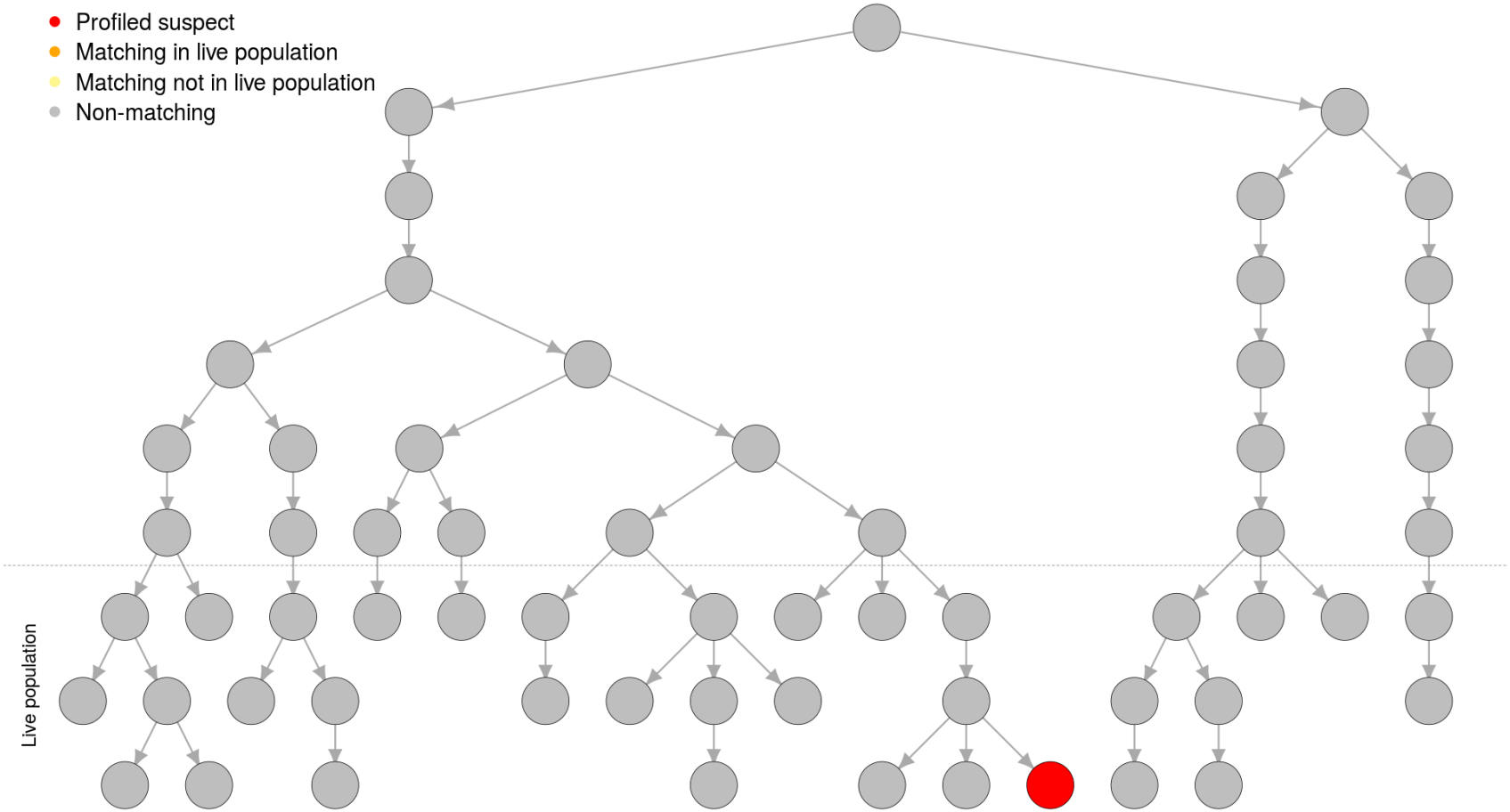
Simulation based model

Simulation based model

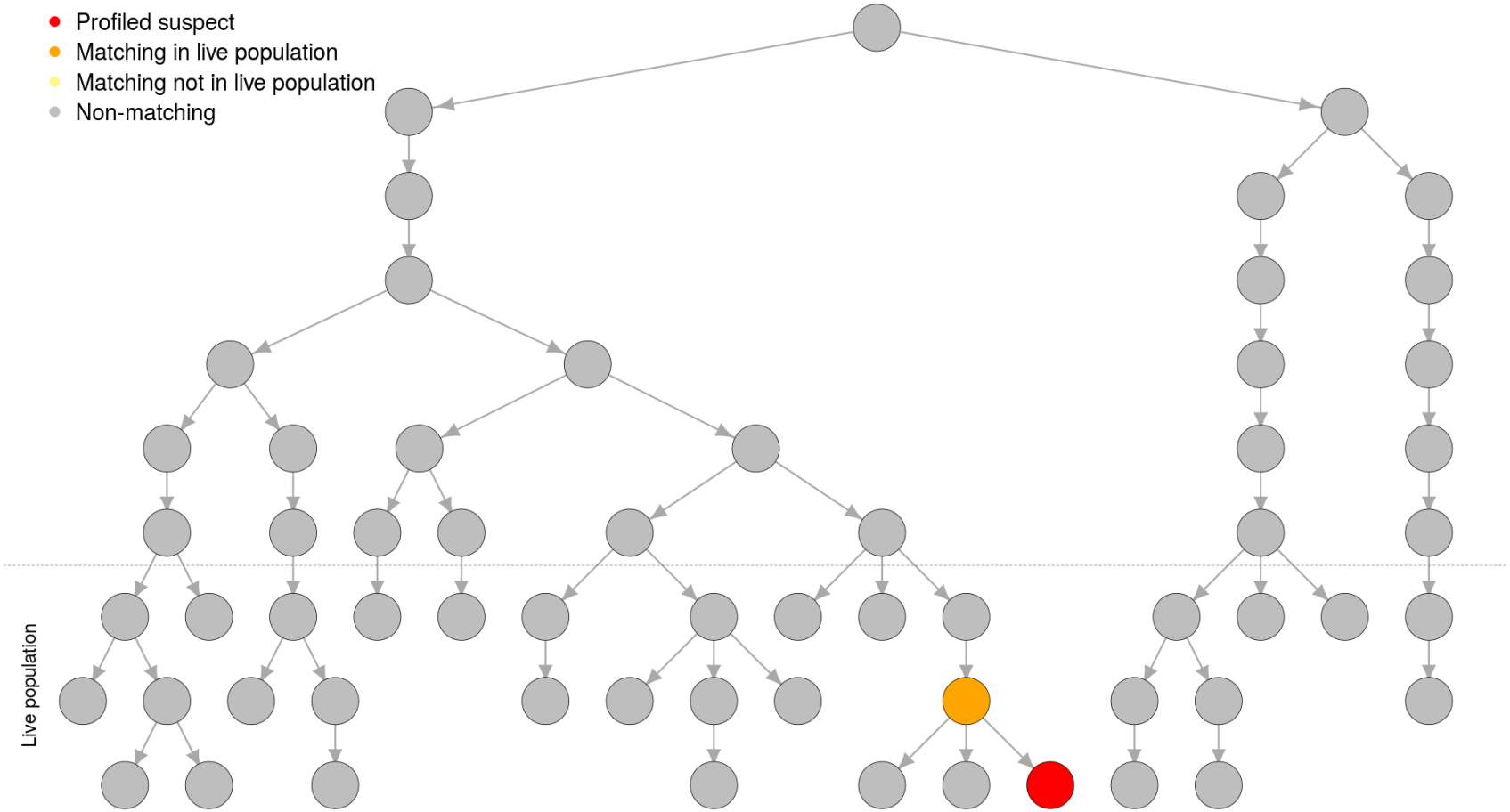
MM Andersen and DJ Balding. "How convincing is a matching Y-chromosome profile?", PLOS Gen, 2017.

- Modern kits, many loci
- Almost all profiles are rare
- "Population frequency" – what population?

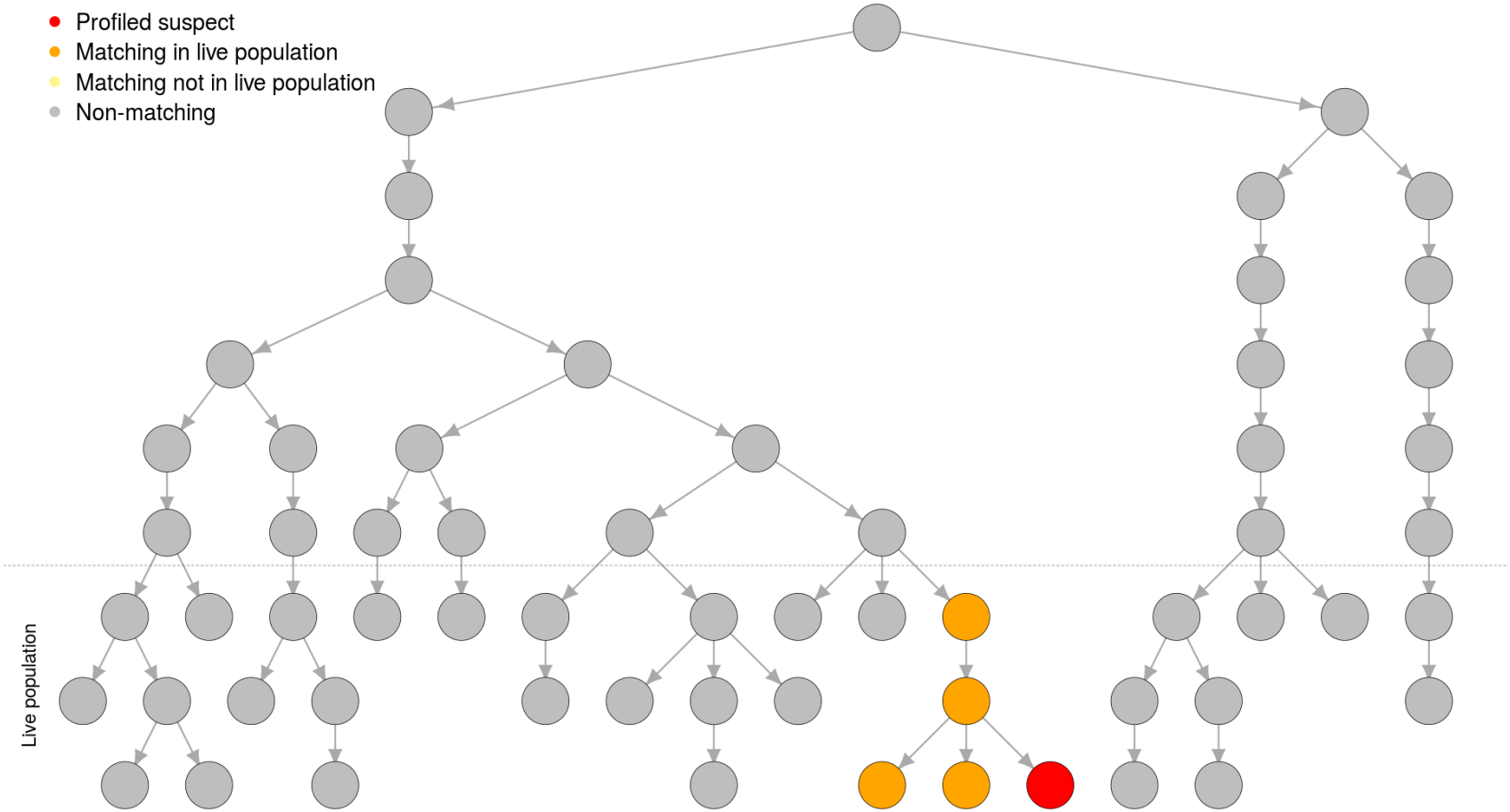
Animation


Profiled suspect
Matching in live population
Matching not in live population
Non-matching
Live population

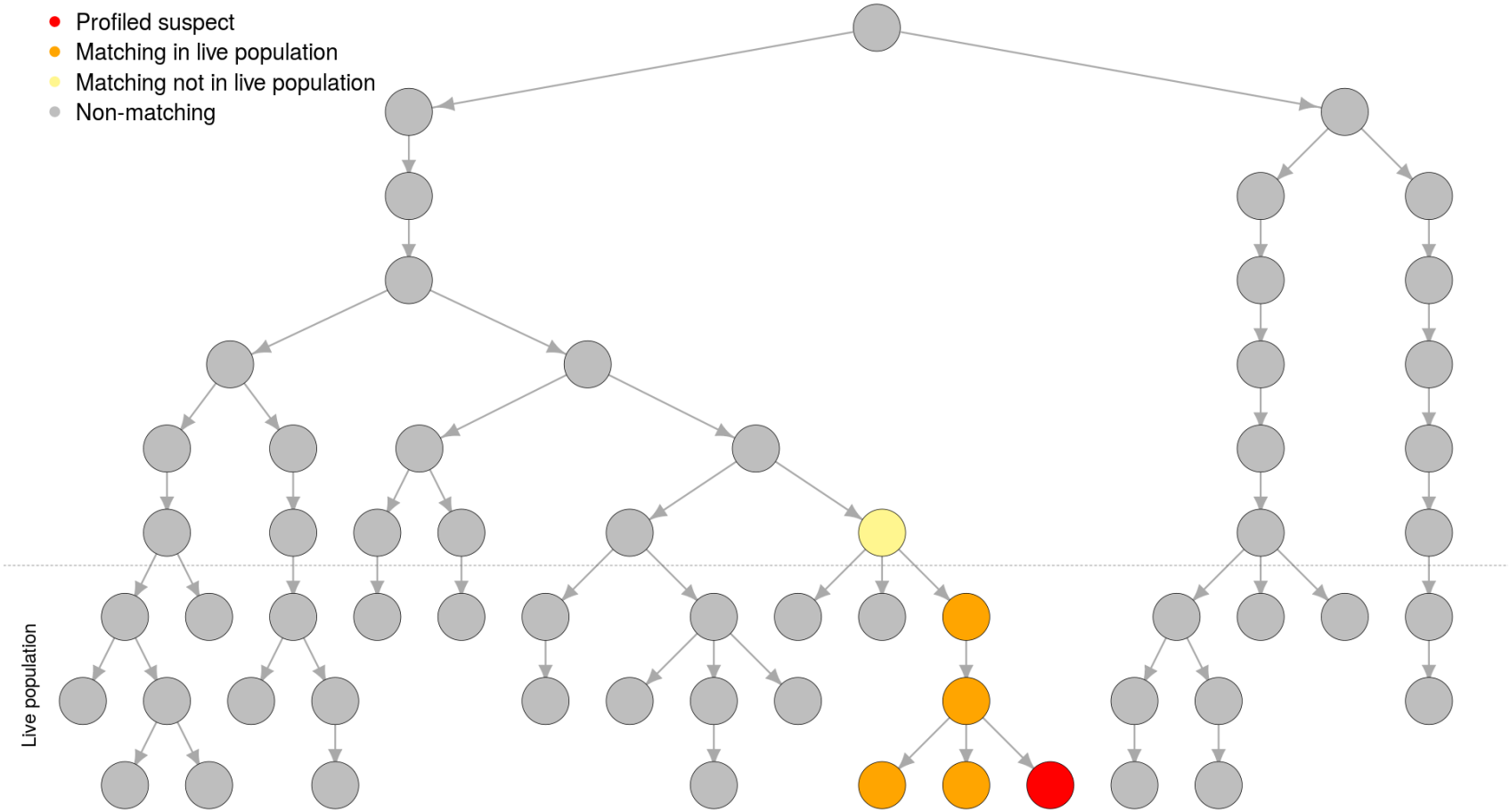
Animation


Profiled suspect
Matching in live population
Matching not in live population
Non-matching
Live population

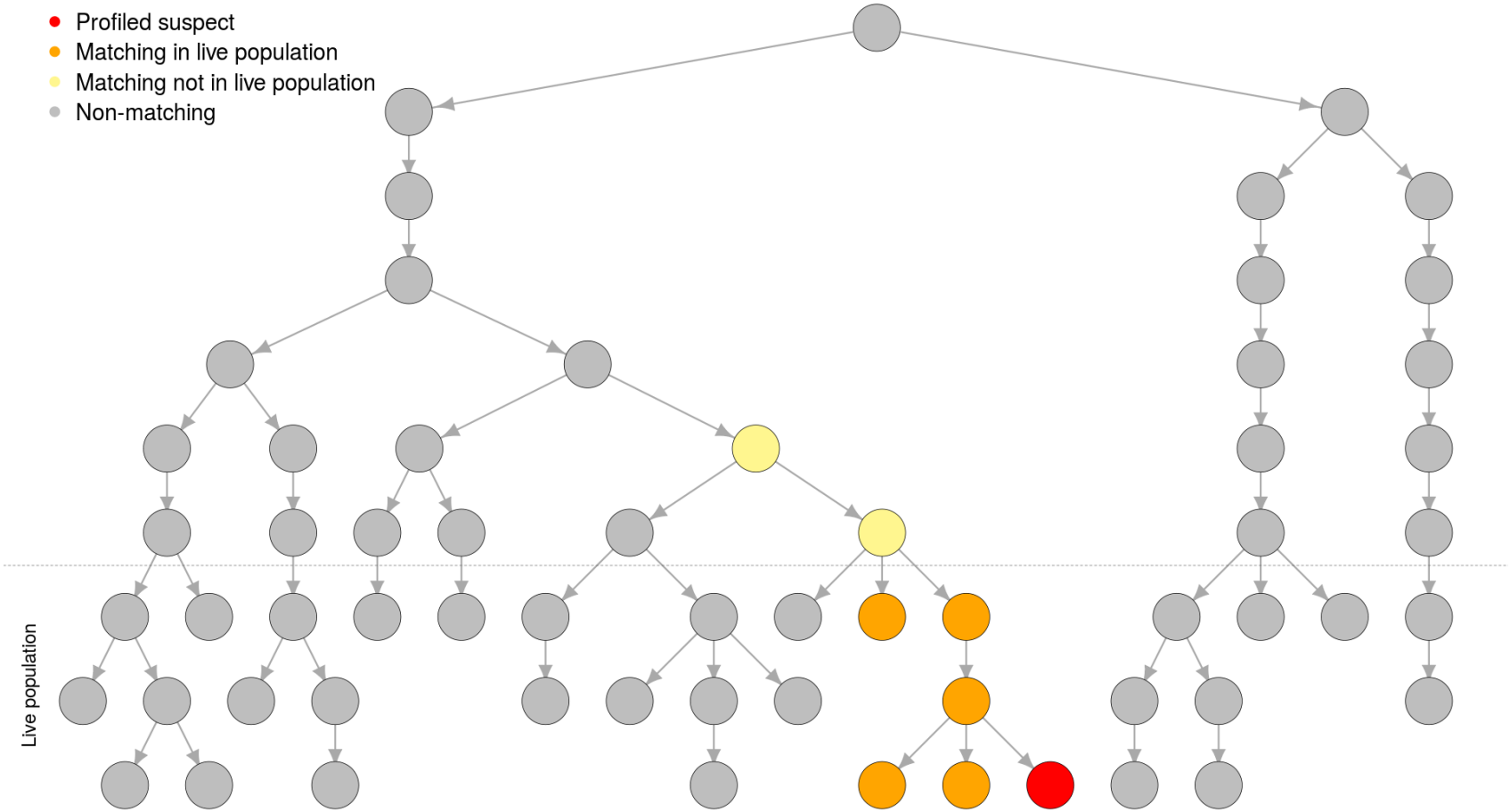
Animation


Profiled suspect
Matching in live population
Matching not in live population
Non-matching
Live population

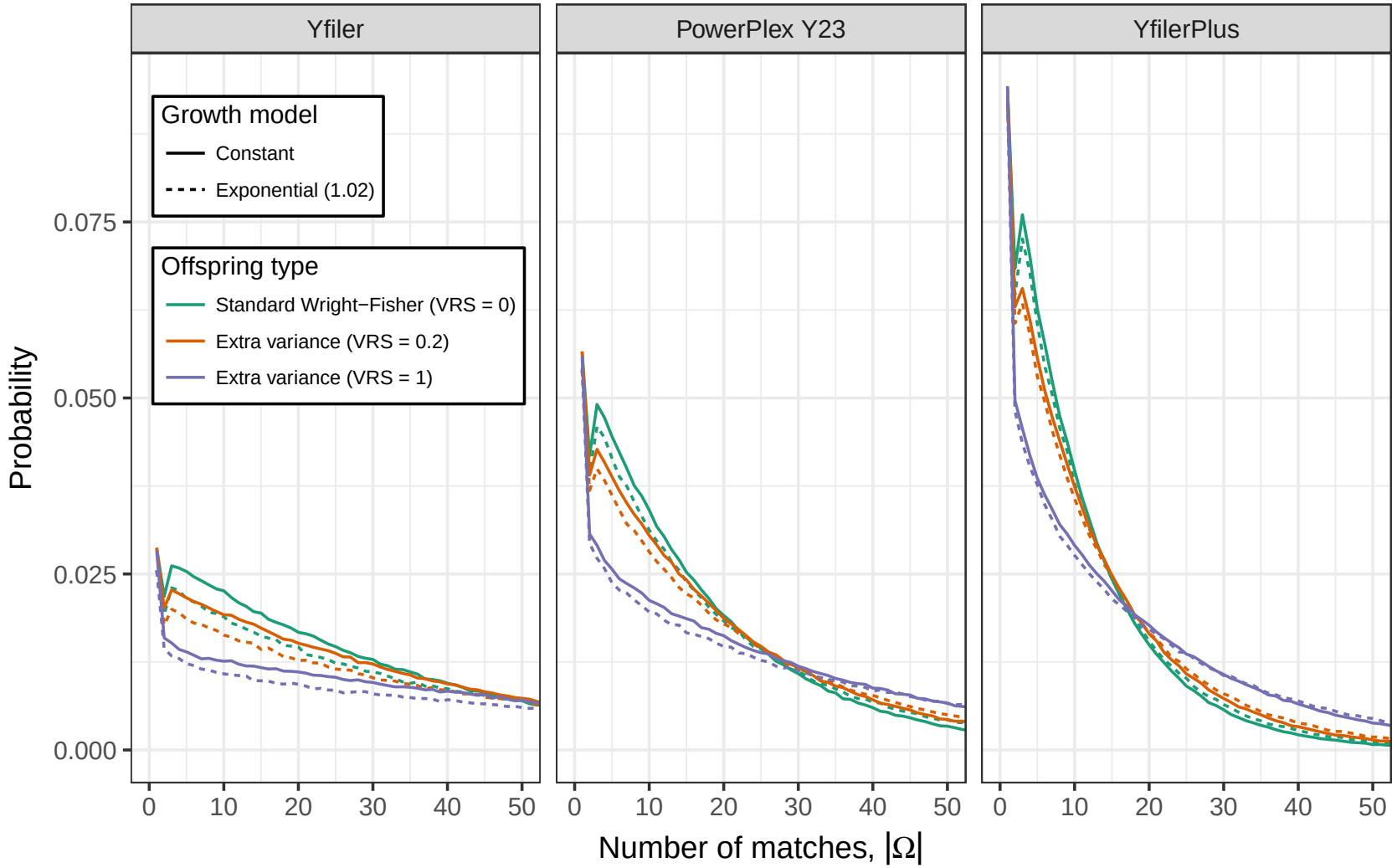
Animation


Profiled suspect
Matching in live population
Matching not in live population
Non-matching
Live population

Animation

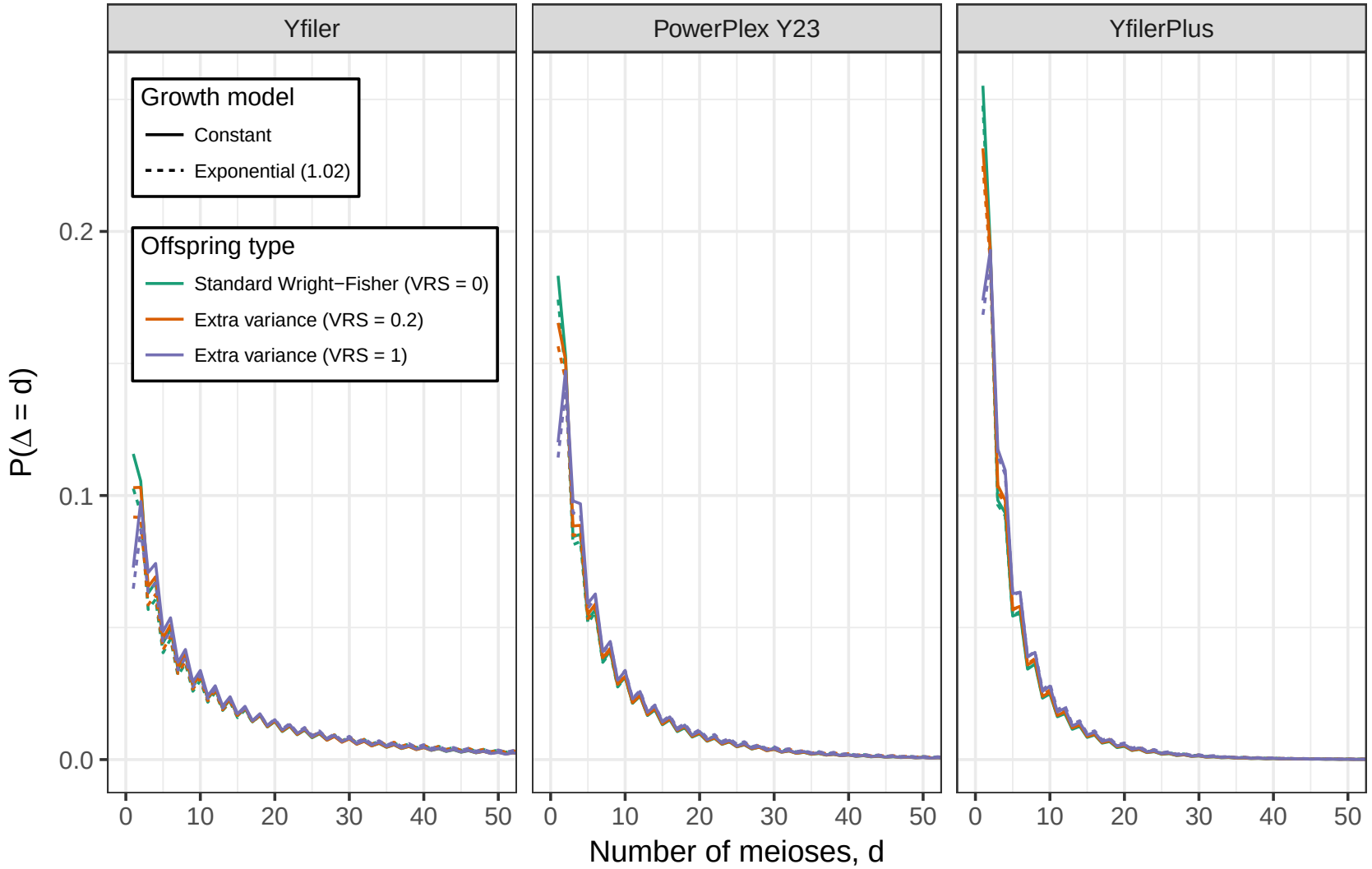

Profiled suspect
Matching in live population
Matching not in live population
Non-matching
Live population

Results



$P(\#\text{ of matches} \leq 37) \geq 0.95$ (Yfiler Plus)

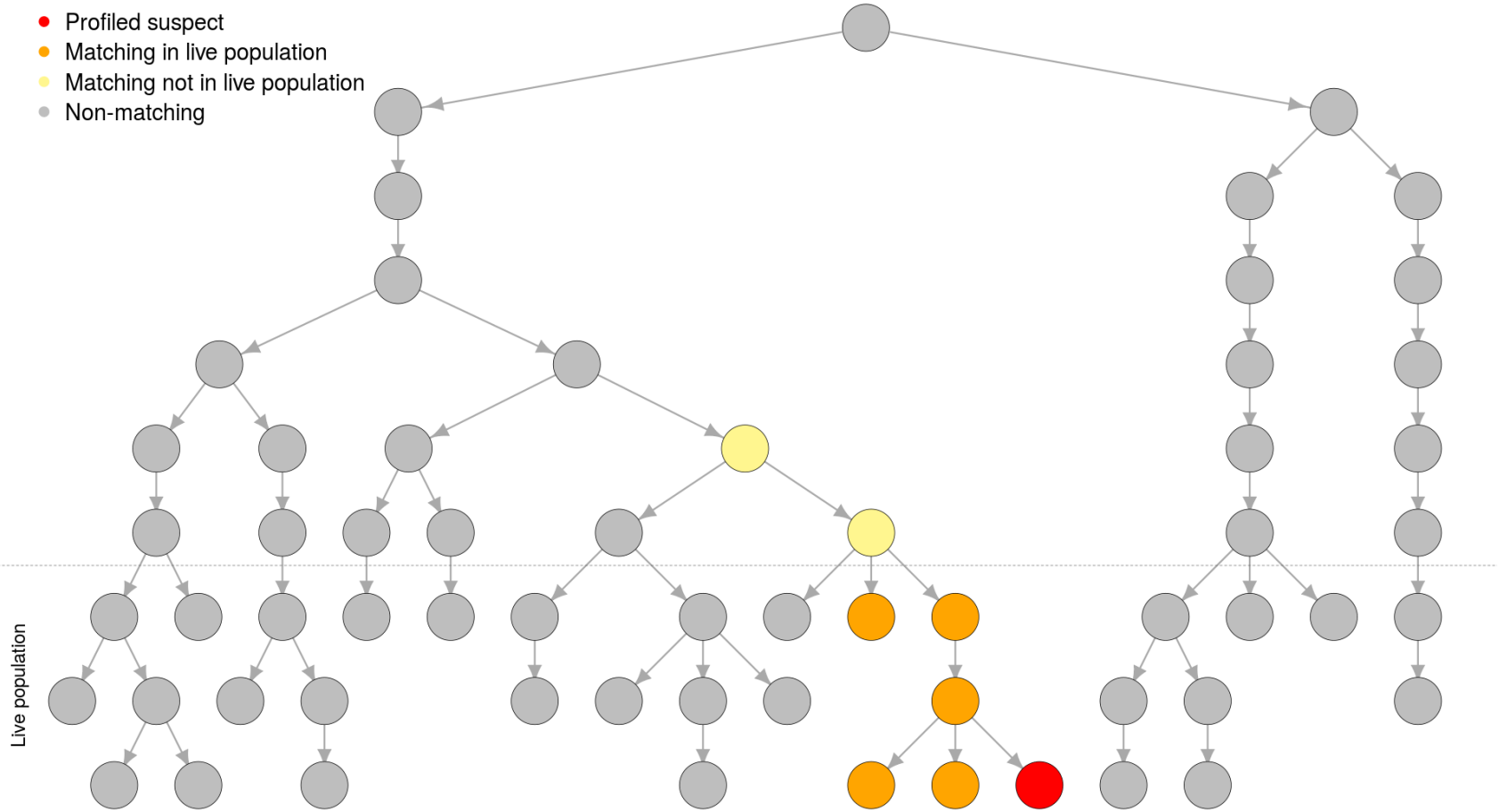
Results



$P(\#\text{ meioses to matches} \leq 18) \geq 0.95$ (Yfiler Plus)

Results

- Suspect: Closer related to culprit than "random man" from population (what population?)
- Number of meioses between suspect and "random man"



Results

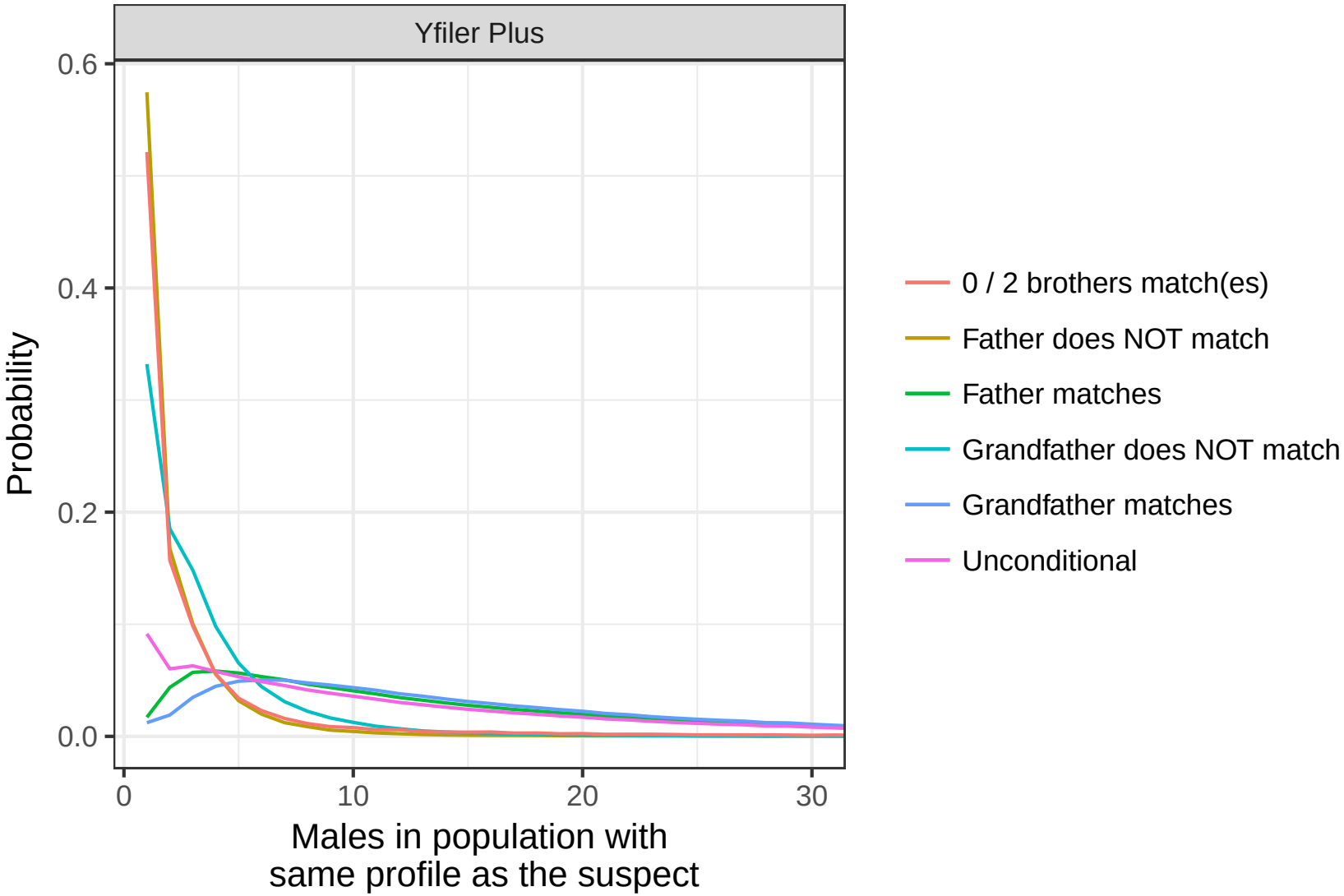
Database information:

- Representative sample?
- Importance sampling reweighting: number of matching males conditional on a database frequency

Yfiler Plus:

$$P(\#\text{ matches} \leq 37) \geq 0.95$$
$$P(\#\text{ matches} \leq 36 \mid \text{db size 1,000 has 0 copies}) \geq 0.95$$
$$P(\#\text{ matches} \leq 56 \mid \text{db size 1,000 has 1 copies}) \geq 0.95$$
$$P(\#\text{ matches} \leq 74 \mid \text{db size 1,000 has 2 copies}) \geq 0.95$$

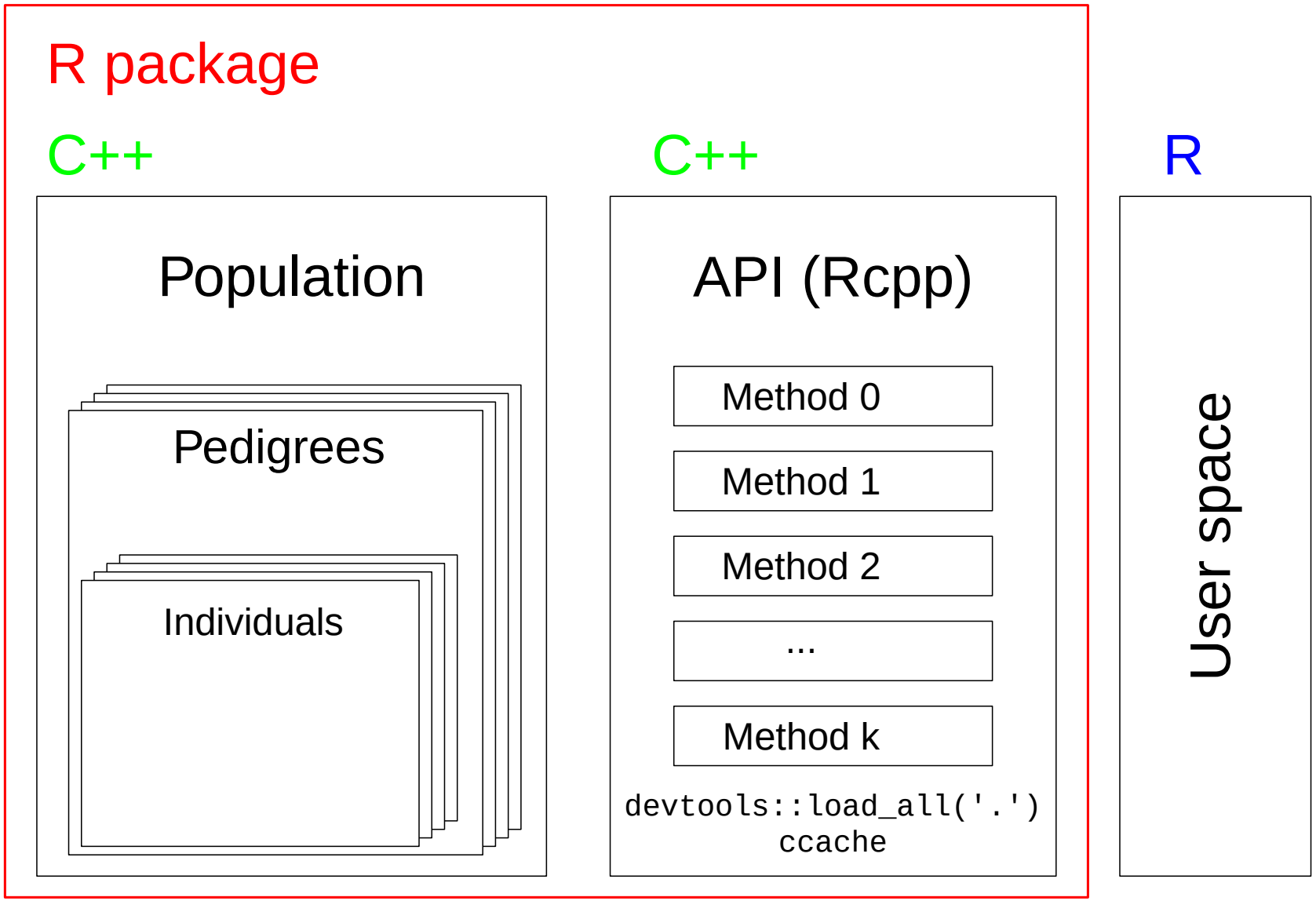
Results


Yfiler Plus
Probability
0.6
0.4
0.2
0.0
0
10
20
30
Males in population with
same profile as the suspect
0 / 2 brothers match(es)
Father does NOT match
Father matches
Grandfather does NOT match
Grandfather matches
Unconditional

Implementation

- References (10^6 individuals)
- R's reference classes / environments
  - Multiple entry points (population, pedigree, individual)
- Pure `igraph` (graph algorithms), no population simulation
- Compromise between run time and development time
- Easily extensible (population parameters, number of meioses, fathers matches, number of brothers, autosomal alleles, . . .)
- Fun and educational

Rcpp / "REPL C++" (Read-eval-print loop)



Rcpp / "REPL C++" (Read-eval-print loop)

```
#include <Rcpp.h>

class MyClass {
private:
  int secret_value;
public:
  MyClass(int value) {
    secret_value = value;
  }
  void do_something() {
    Rcpp::Rcout << "Secret = " <<
      secret_value << std::endl;
  }
};
```

Rcpp / "REPL C++" (Read-eval-print loop)

```
// [[Rcpp::export]]
Rcpp::XPtr<MyClass> create_object(const int value) {
  MyClass* m = new MyClass(value);
  Rcpp::XPtr<MyClass> p(m);
  p.attr("class") =
    Rcpp::CharacterVector::create("myclass_xptr",
                                  "externalptr");
  return p;
}

// [[Rcpp::export]]
void print_myclass_xptr(const Rcpp::XPtr<MyClass> p) {
  p->do_something();
}
```

Rcpp / "REPL C++" (Read-eval-print loop)

```
print.myclass_xptr <- function(x, ...) {
  print_myclass_xptr(x)
}

x <- create_object(2018)
x
```

```
## Secret = 2018
```

Rcpp / "REPL C++" (Read-eval-print loop)

```
x <- create_object(2018)
x
```

```
## Secret = 2018
```

(Implement `plot(x)`, ...)

```
devtools::load_all('.')

# No need to restart R nor run create_object() again!

plot(x)
```

Rcpp / "REPL C++" (Read-eval-print loop)