# PONDEROSA Tutorial

Henn lab meeting
13 May 2021

# Resources

- Github repo (with manual): https://github.com/williamscole/PONDEROSA

- Any questions (or bugs), please please shoot me an email: cole_williams@brown.edu

# When should I use PONDEROSA?

1. I need to generate a .fam file

2. I know which pairs are parent-offspring, but I’m not sure who the parent is, who the child is

3. I’m not sure whether two people are half-siblings or full siblings

4. I want to identify all the relationships in the dataset

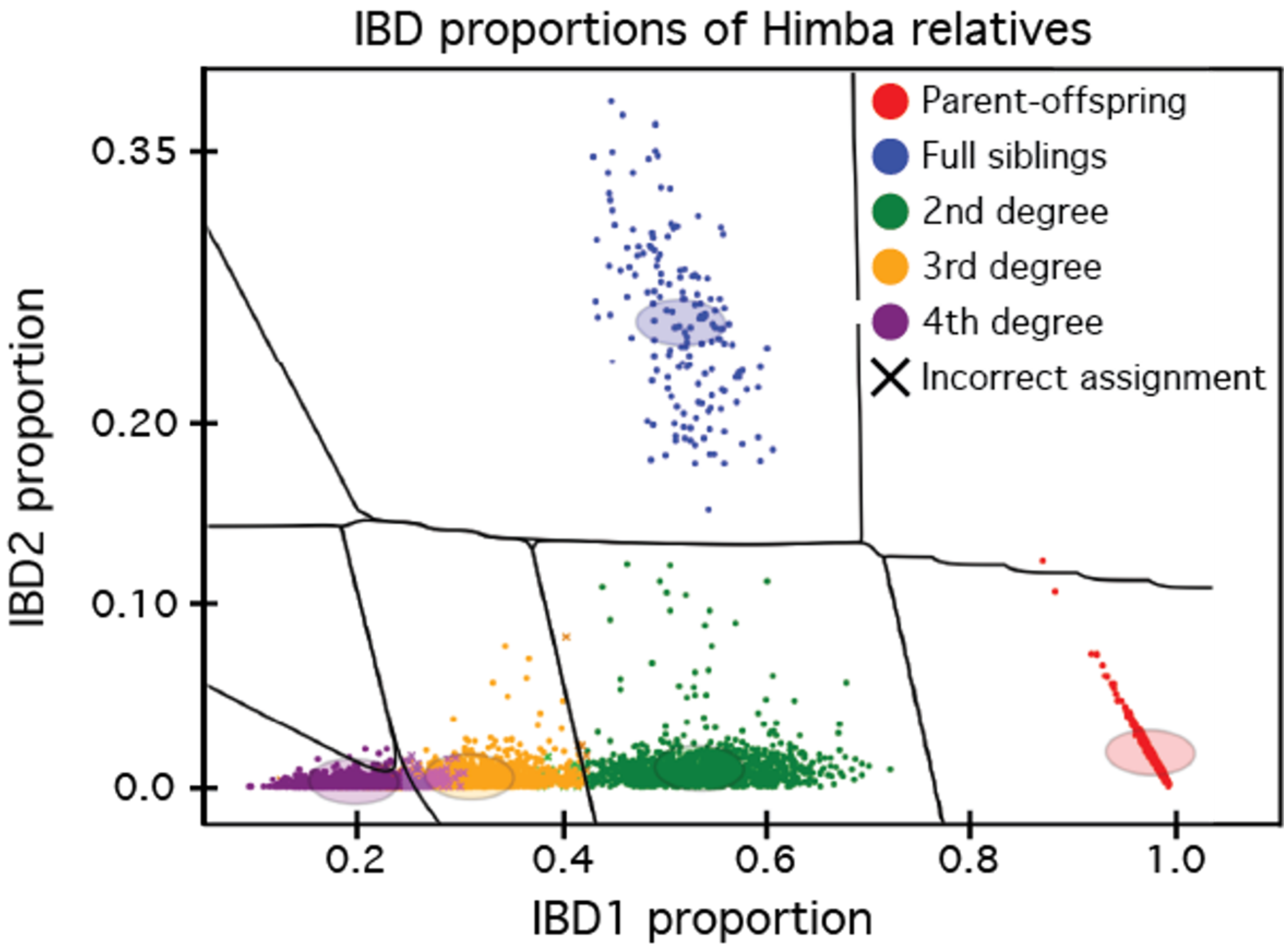
5. There are 2nd degree pairs, but I’m not sure whether they are half-siblings, grandparent-grandchildren or avuncular

6. I want to identify a set of unrelated individuals

# Motivation behind PONDEROSA

- Himba relatives share more IBD than outbred populations
- Existing algorithms overestimated relatedness in the Himba
    - E.g. Himba cousins being misclassified as half-siblings
- PONDEROSA mines the dataset for high-confidence relationships and uses these to train a machine learning classifier

# LDA classifier used by PONDEROSA



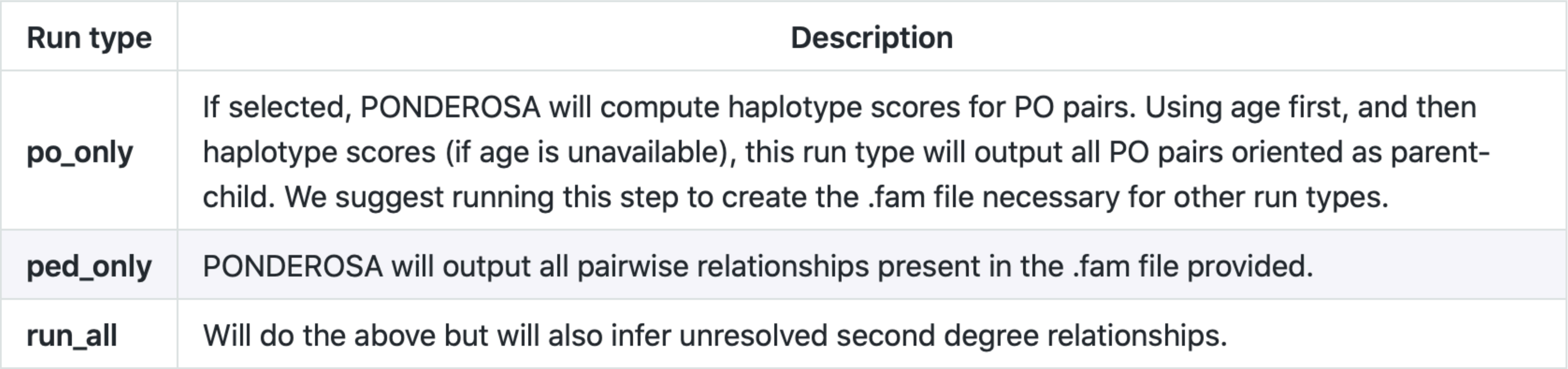
# Running PONDEROSA

- PONDEROSA was written in python and requires python 3.6 or higher
- PONDEROSA requires the sci-kit learn python package, numpy, pandas
    - Either create a virtual environment or use anaconda3's python
    - On augrabies: /software/anaconda3/4.5.12/lssc0-linux/bin/python3.6
- PONDEROSA's github repo is found here: /share/hennlab/lab_scripts/PONDEROSA
    - Good practice to git pull before using it, just in case I've made any changes

# Running PONDEROSA

- PONDEROSA has **three** different run types that should be run sequentially for a new dataset

- For a complete manual, see https://github.com/williamscole/PONDEROSA



| Run type | Description |
|---|---|
| **po_only** | If selected, PONDEROSA will compute haplotype scores for PO pairs. Using age first, and then haplotype scores (if age is unavailable), this run type will output all PO pairs oriented as parent-child. We suggest running this step to create the .fam file necessary for other run types. |
| **ped_only** | PONDEROSA will output all pairwise relationships present in the .fam file provided. |
| **run_all** | Will do the above but will also infer unresolved second degree relationships. |

# Quick set up

- git clone https://github.com/williamscole/PONDEROSA_tutorial
- cd PONDEROSA_tutorial
- source /share/hennlab/projects/himba_pedigree/PONDEROSA_tutorial/bin/activate
- bash setup.sh

# A note on calling IBD

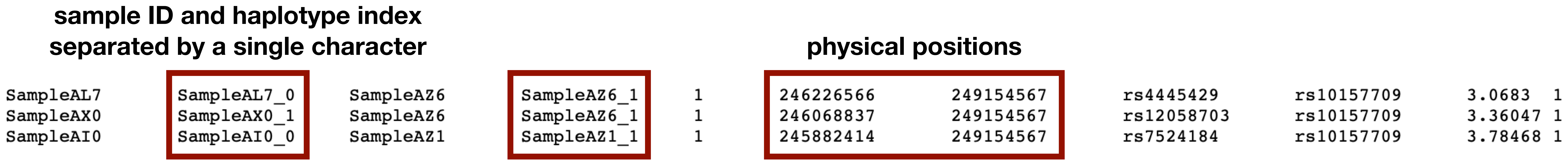
- Many, many IBD callers out there

- Doesn't really matter which you use (in terms of format), as long as it can be run on haploid genomes

  - Germline v1.5 (—**haploid** flag)

  - iLASH (haploid method; no flag)

  - phasedibd: 23andMe's new IBD caller. use_phase correction **must be False.** Must rearrange columns to be in Germline or iLASH format



**sample ID and haplotype index separated by a single character**

**physical positions**

```
SampleAL7     SampleAL7_0     SampleAZ6     SampleAZ6_1     1     246226566     249154567     rs4445429     rs10157709     3.0683  1
SampleAX0     SampleAX0_1     SampleAZ6     SampleAZ6_1     1     246068837     249154567     rs12058703    rs10157709     3.36047 1
SampleAI0     SampleAI0_0     SampleAZ1     SampleAZ1_1     1     245882414     249154567     rs7524184     rs10157709     3.78468 1
```

# ***po_only***

- You’ve just received a new dataset; all you have is the genotype data and (maybe) some age data

- Pre-PONDEROSA steps

  - Run GERMLINE (**—haploid** flag) or iLASH

  - Run KING (**—kinship** flag)

- That’s it! Now PONDEROSA will…

  1. Use KING to identify parent-offspring pairs

  2. Use age to identify who is the parent, who is the child

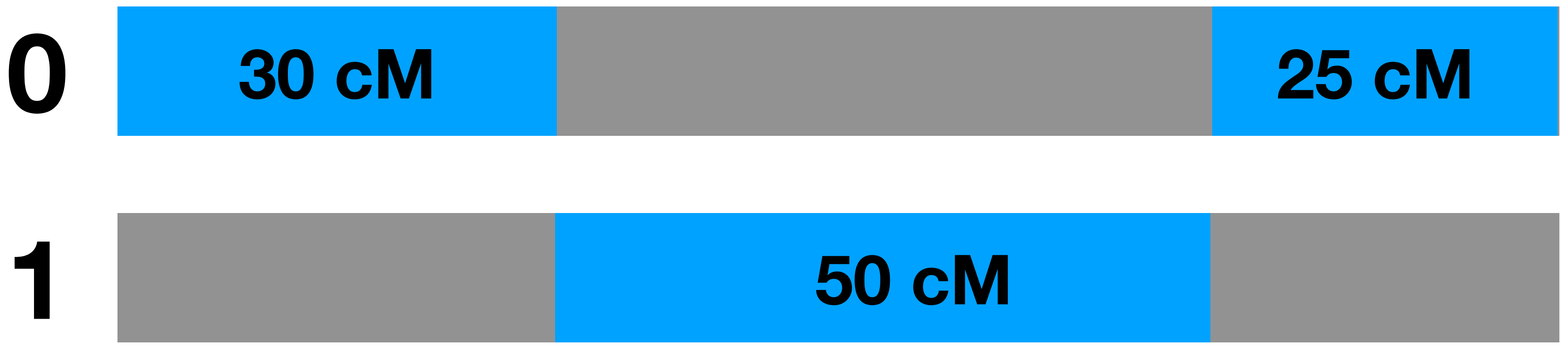
  3. If no/incomplete age data, uses **haplotype scores** to do this

# Haplotype scores

- Haplotype scores are calculated for **each individual** in **each pair**
  - e.g. the haplotype score of individual $i$ in the pair $(i, j)$ will likely be different than $i$'s haplotype score in the pair $(i, k)$
- Haplotype scores reflect how much IBD is shared on one haplotype vs. the other haplotype

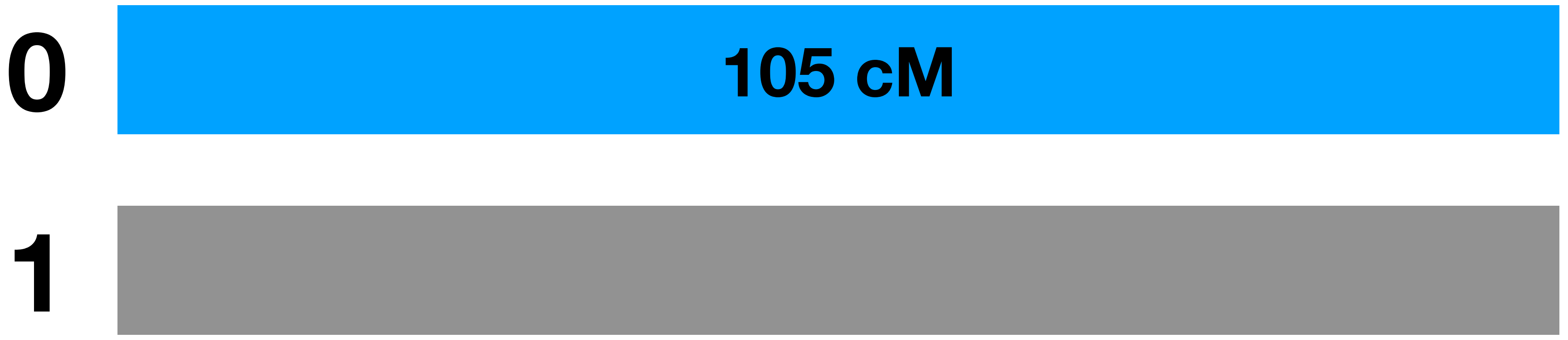
# More on haplotype scores

**Take individuals $i$ and $j$ who are parent-offspring**



**Individual $i$**

$$h_i = \frac{max(50{,}55)}{55 + 50} = 0.52$$



**Individual $j$**

$$h_j = \frac{max(105{,}0)}{105 + 0} = 1$$

# More on haplotype scores

- Haplotype scores allow PONDEROSA to infer the genetically older individual in a pair
    - Parent in a parent-offspring pair
    - Grandparent in a grandparent-grandchild pair
    - Aunt/uncle in an avuncular pair
- The genetically **older** relative should have a **lower** haplotype score than the **younger** relative
- Individuals that are **genetically** the same age (e.g. the same generation) should have similar haplotype scores

# [prefix]_PO.txt file

- The script **make_fam.py** converts [prefix]_PO.txt file to .fam file
- Easy to run: python make_fam.py par_file.txt
  - Don’t need to make any changes to the par_file
- It does several things
  1. If there are siblings, makes sure they have the same parent
  2. Checks for sex errors (e.g. an individual has two parents of the same sex)
  3. If sex is missing, infers the missing sex of the parents
  4. Checks age gaps of parents for possible age errors
  5. Several checks to make sure the parent-offspring orientation is correct

# ***ped_only***

- Now that we have a full .fam file, we can explore what relatives are present in our dataset

- ***ped_only*** uses the parent-offspring pairs from the .fam file to identify 2nd, 3rd, and 4th degree relatives

- Only requires a complete **.fam file** and a KING **.seg file**

- If we want to **run_all**, we must have enough training pairs (at least one of each PO, FS, 2nd, 3rd, 4th and at least one of GP, AV, PHS, and MHS)

    - More training pairs is better though. I recommend more than 10 pairs of each category

# Identifying sets of unrelated individuals

- Run the **remove_relatives.py** script
    - I created a bash script remove_relatives.sh that makes this easy to run; just change the parameters in the script
- The script gives **many** options for (1) specifying the maximum relatedness tolerated in the set and (2) how relatedness is determined
- 3 modes:
    - LDA: uses the PONDEROSA [prefix]_pairs.txt file to train an LDA classifier. The classifier is used to determine relatedness
    - KING: uses the KING **.seg** file's relatedness inference.
    - float: the user specifies the maximum kinship coefficient to be tolerated

# remove_relatives.py

- LDA mode

```
python remove_relatives.py Himba_pairs.txt Himba.seg Himba.fam 3rd
```

- KING mode

```
python remove_relatives.py None Himba.seg Himba.fam 3rd
```

- float mode

```
python remove_relatives.py 0.18 Himba.seg Himba.fam 5th
```

# ***run_all***

- ***run_all*** does the previous steps *and* infers the relationship of 2nd degree relatives
    - 2nd degree relatives include half-siblings, avuncular (e.g. aunt/niece), and grandparent-grandchild
- ***run_all*** is only useful if there are enough training pairs
    - Running ***ped_only*** will give you an idea of how many training pairs you have; if you don’t PONDEROSA will (gracefully) stop running

# How does PONDEROSA distinguish 2nd degree relatives?

- 2nd degree relatives are hard to distinguish because they all (on average) share the same proportion of the genome IBD

- PONDEROSA trains another LDA classifier with two statistics:

  - **The number of IBD segments shared,** $n$

  - **The haplotype score ratio,** $HSR$

    - For a given pair $(i, j)$, $HSR_{ij} = \dfrac{h_i}{h_j}$ where $h_j > h_i$

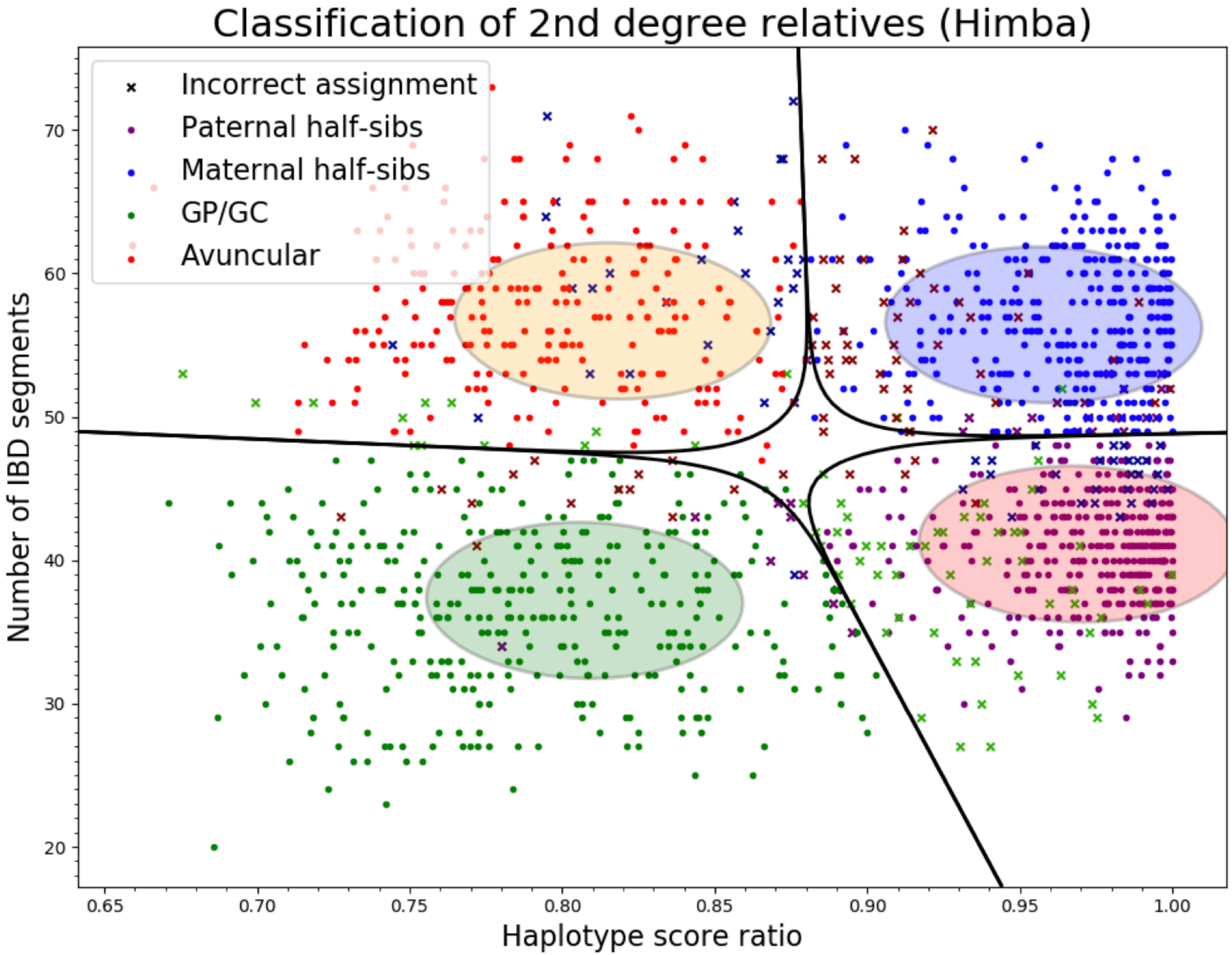
  - Because half-siblings are of the same generation, we expect $h_j = h_i$ so $HSR_{ij} = 1$

  - This is not the case in avuncular/grandparent-grandchild pairs, where $h_j > h_i$ so $HSR_{ij} < 1$

# How does PONDEROSA distinguish 2nd degree relatives?

- PONDEROSA will look for other information to help distinguish relationships
  - Parameter *mhs_gap* is the maximum age gap two maternal half-siblings can be; if a pair exceeds this then $P(MHS) = 0$
  - *gp_gap*: the minimum age gap for two individuals to be considered grandparent-grandchildren

# LDA classifier for 2nd degree relatives



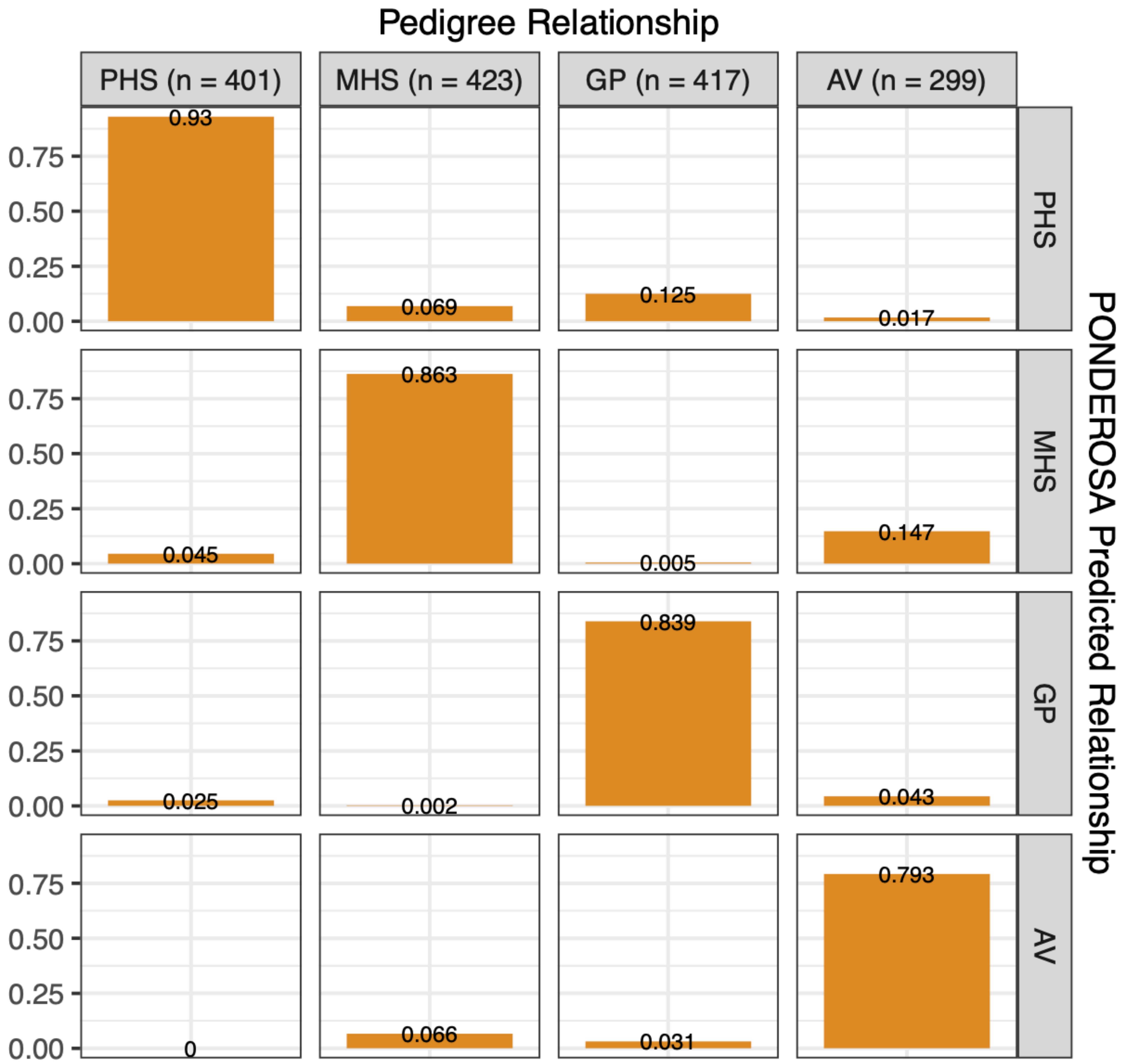
# Evaluating [prefix]_second.txt

- This step of PONDEROSA works well when the data are phased well
- PONDEROSA stitches together IBD segments, so that $n$ (the number of IBD segments) is robust to phase errors
- The HSR is more sensitive to phase errors
- When phase quality is poor, PONDEROSA will have a hard time distinguishing half-siblings from grandparent-grandchild/avuncular
    - But it can still do a good job distinguishing MHS/avuncular from PHS/grandparent-grandchild

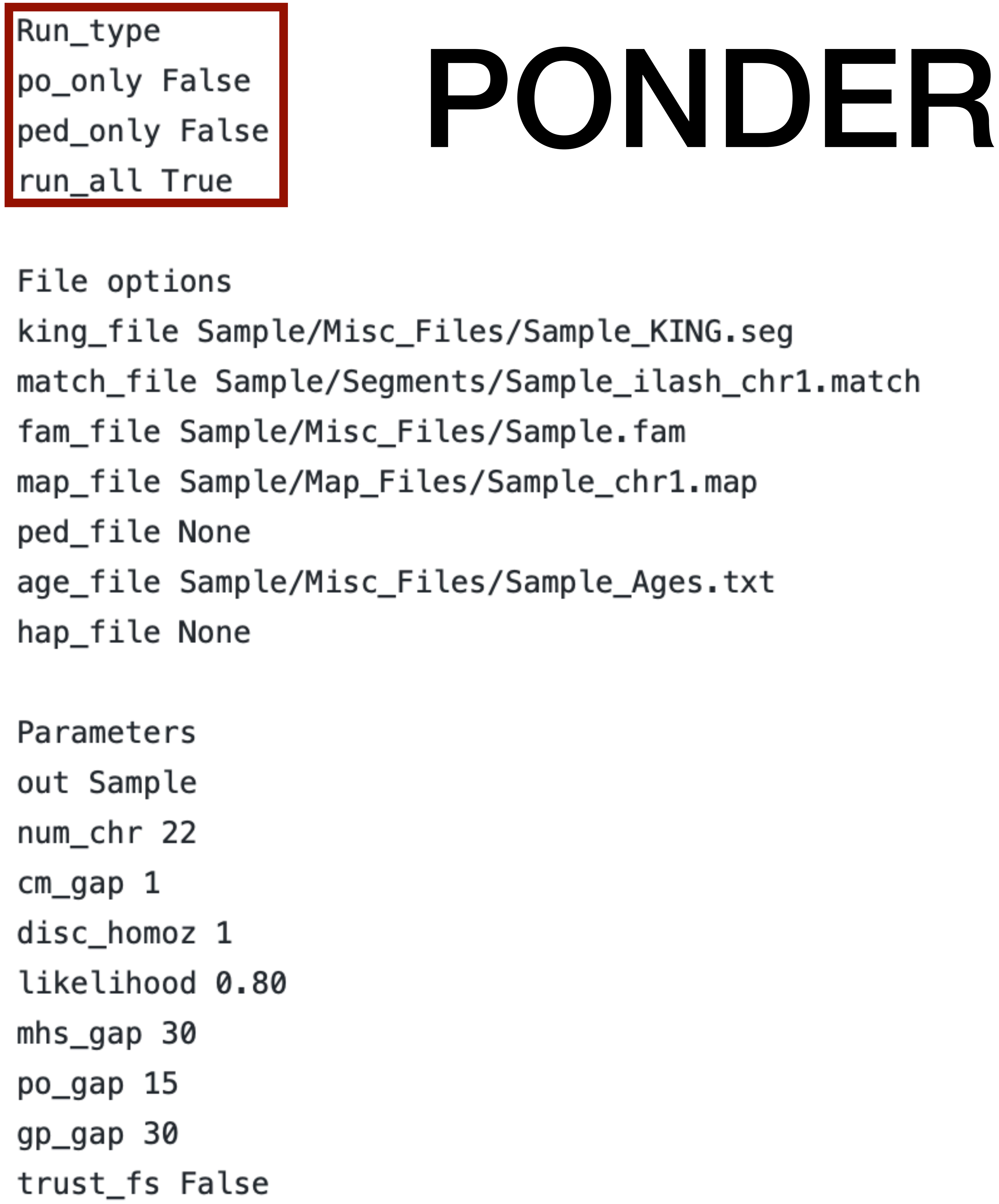
# Common PONDEROSA errors

- GP and AV tend to be misclassified as HS

- Half-siblings misclassifications *tend* to be sex misclassifications of the missing parent



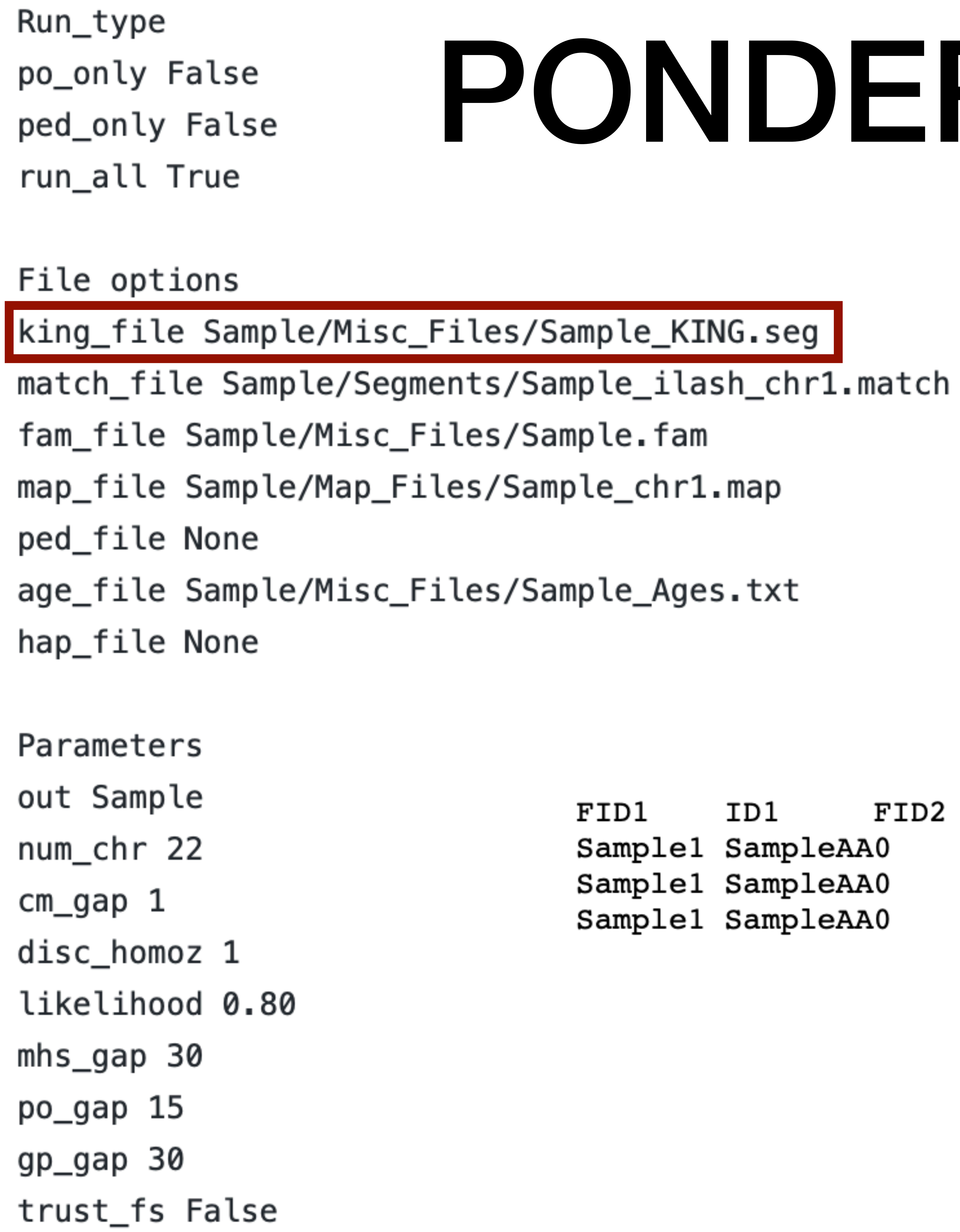
# PONDEROSA par_file



```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

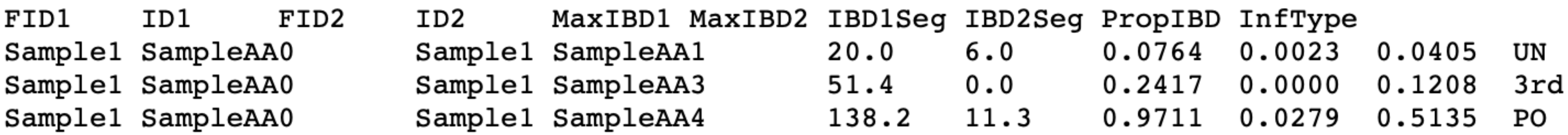
- Run_type specifies the run type; only one can be true

# PONDEROSA par_file



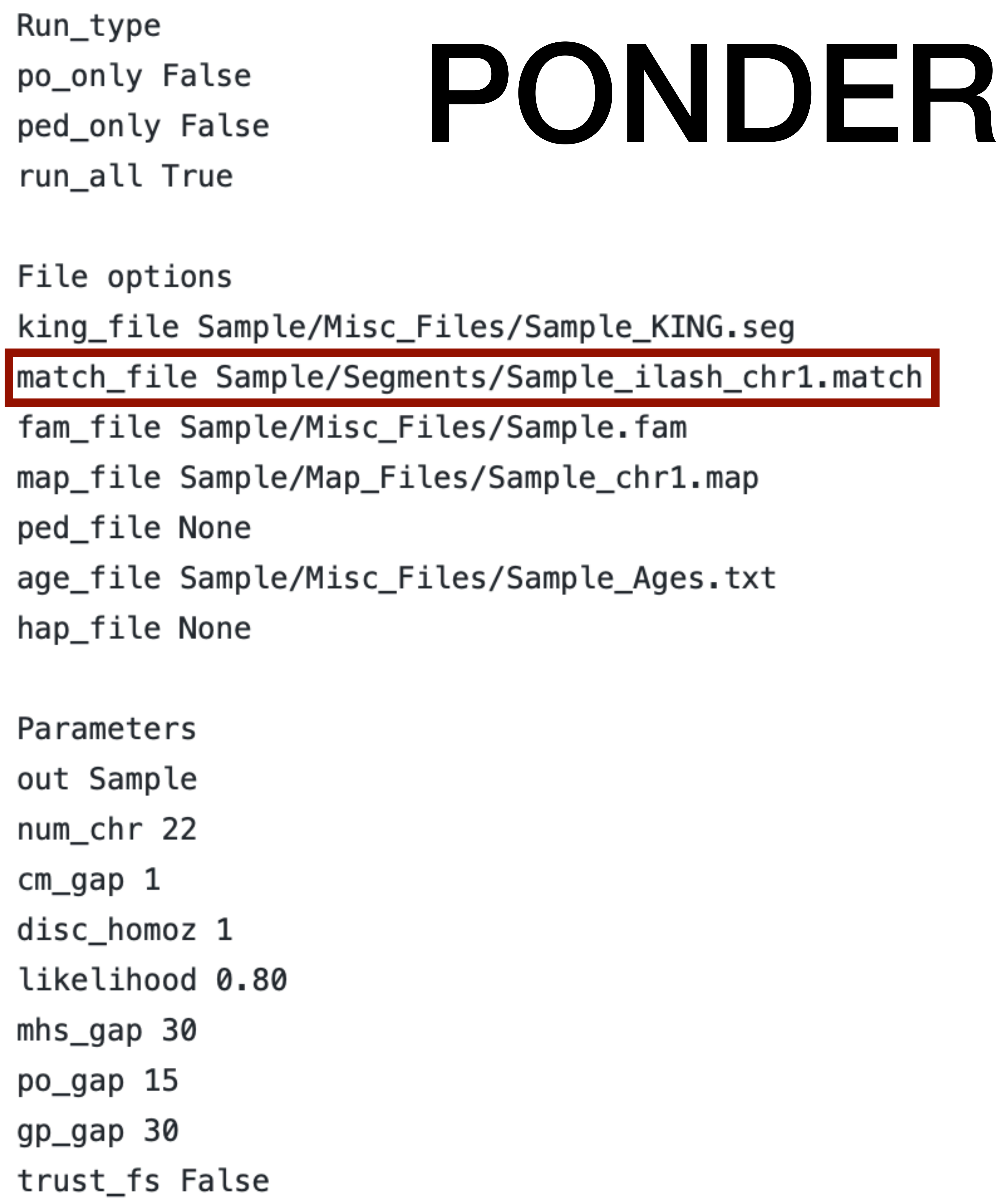
```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

- king_file is the path to the KING generated .seg file
- Generated using KING's —**kinship** flag
- Required for **all** run types



```
FID1    ID1      FID2    ID2      MaxIBD1 MaxIBD2 IBD1Seg IBD2Seg PropIBD InfType
Sample1 SampleAA0        Sample1 SampleAA1        20.0    6.0     0.0764  0.0023  0.0405  UN
Sample1 SampleAA0        Sample1 SampleAA3        51.4    0.0     0.2417  0.0000  0.1208  3rd
Sample1 SampleAA0        Sample1 SampleAA4        138.2   11.3    0.9711  0.0279  0.5135  PO
```

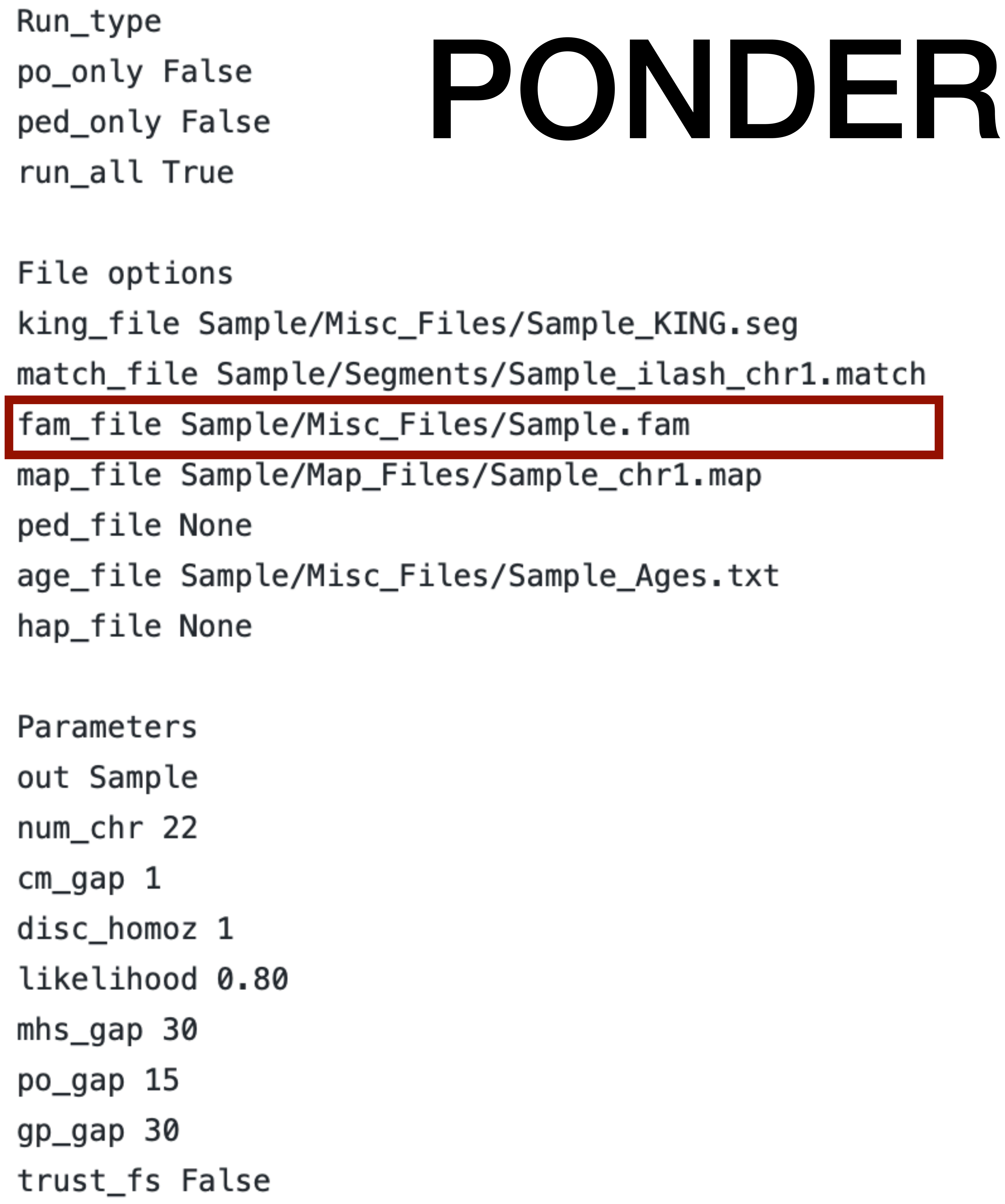
# PONDEROSA par_file



```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

- IBD segments generated from Germline or iLASH
- Germline: **must** use **—haploid** flag
- Required for **po_only** and **run_all** run types

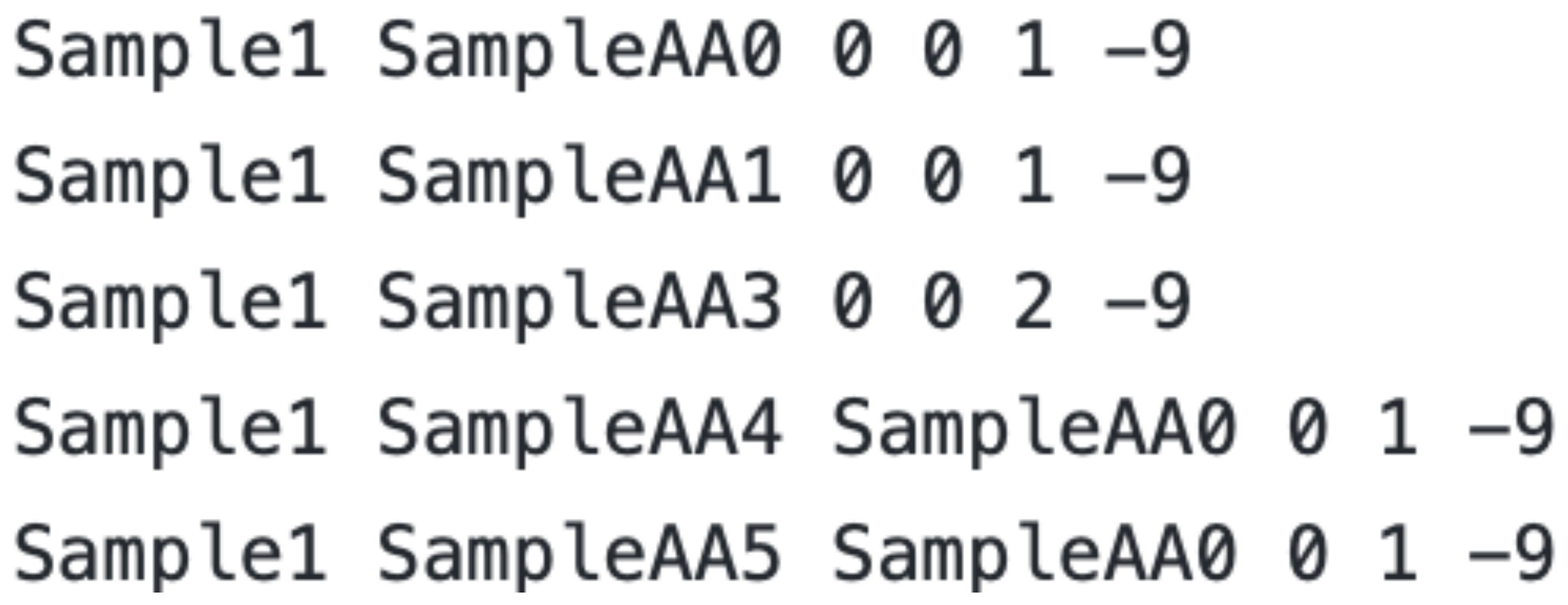
# PONDEROSA par_file



```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

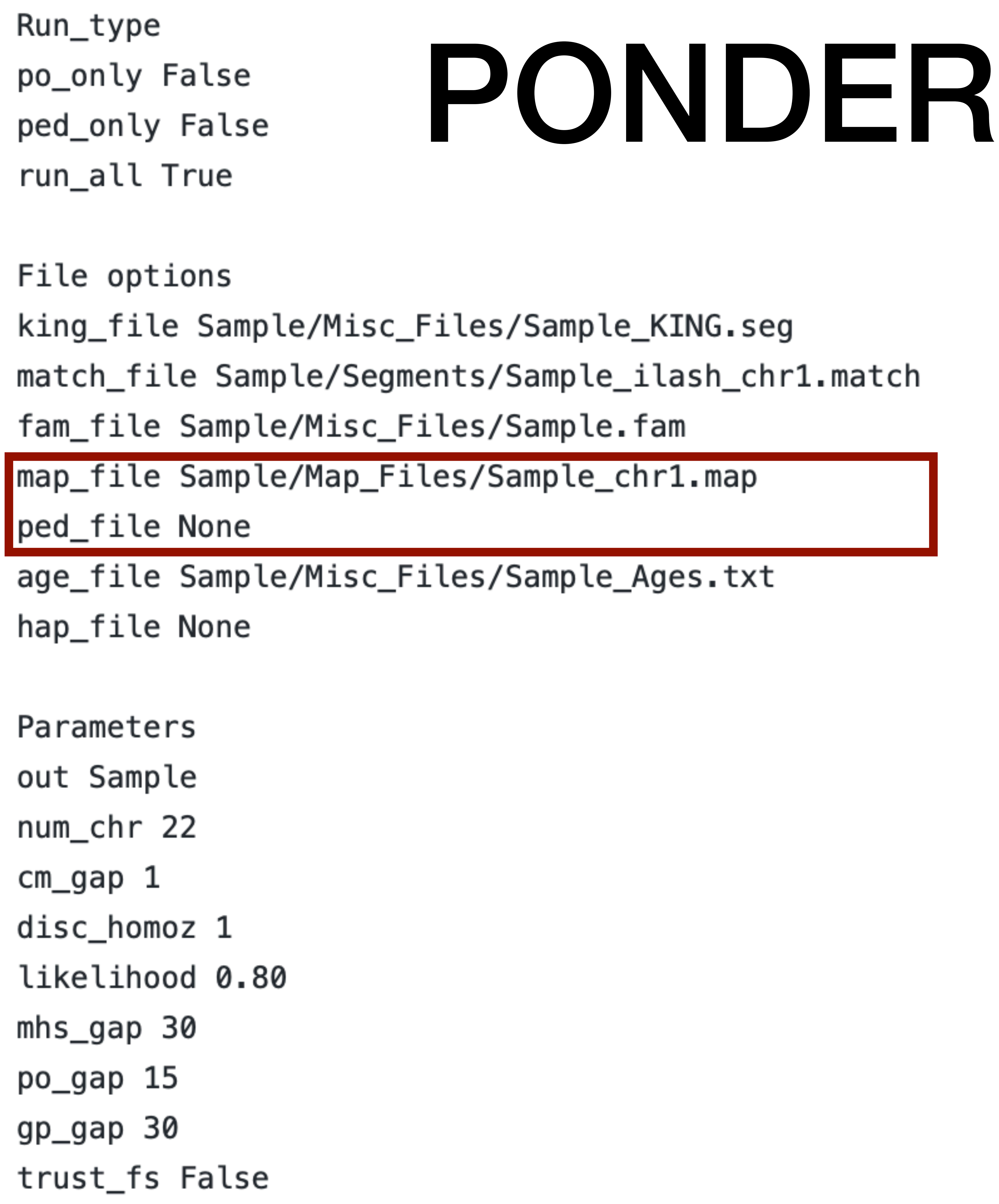
- .fam file contains the individual ID, father ID, mother ID, sex

- Required for **ped_only** and **run_all**



```
Sample1 SampleAA0 0 0 1 -9
Sample1 SampleAA1 0 0 1 -9
Sample1 SampleAA3 0 0 2 -9
Sample1 SampleAA4 SampleAA0 0 1 -9
Sample1 SampleAA5 SampleAA0 0 1 -9
```

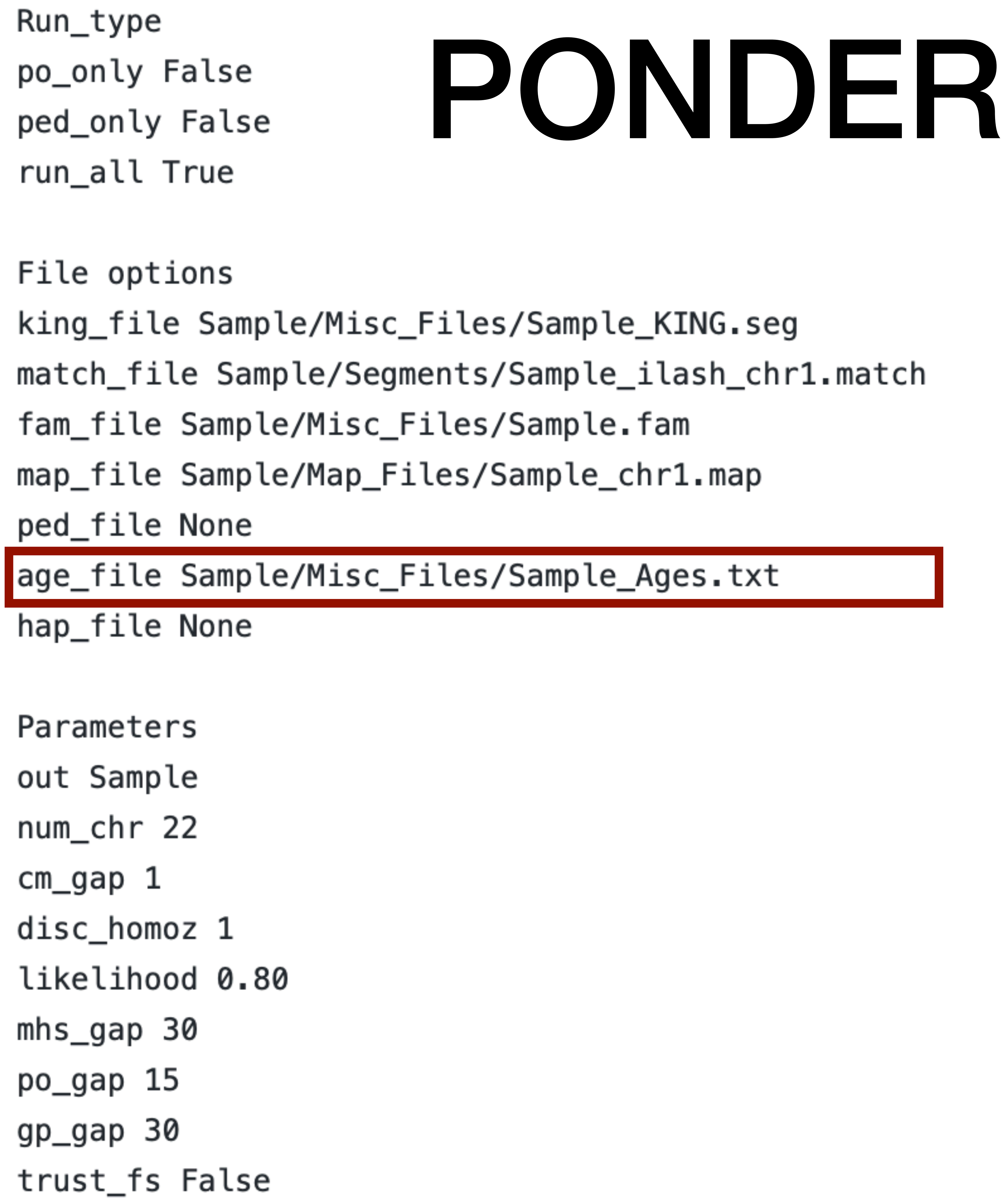
# PONDEROSA par_file



```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

- PLINK .map and .ped files
- .map file required for **po_only** and **run_all**
- .ped file optional for **po_only** and **run_all**

# PONDEROSA par_file

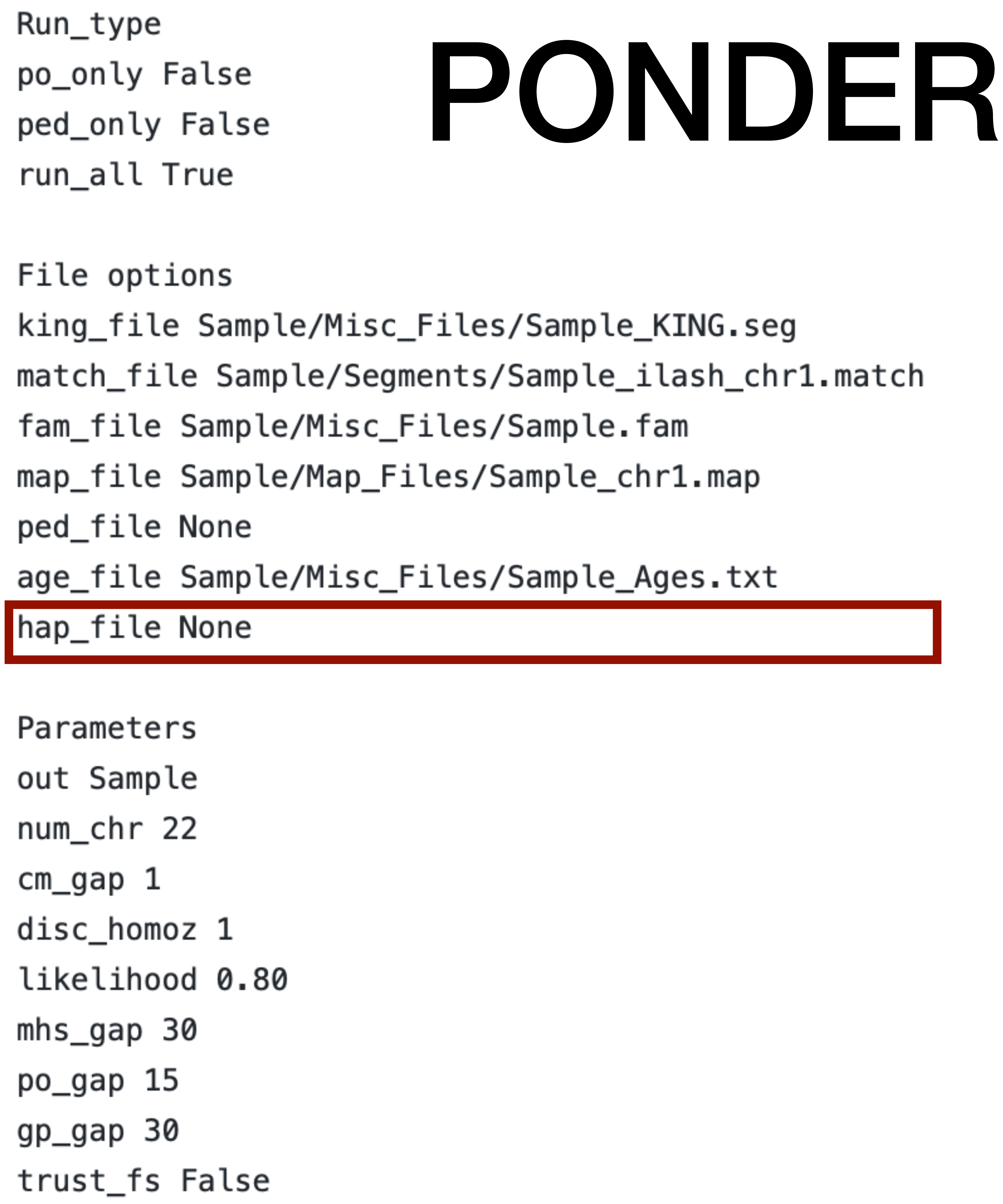


```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

- Age file. Column 1 is the IID; column 2 is the age (int or float)

- Optional for **po_only** and **run_all**

# PONDEROSA par_file



```
Run_type
po_only False
ped_only False
run_all True

File options
king_file Sample/Misc_Files/Sample_KING.seg
match_file Sample/Segments/Sample_ilash_chr1.match
fam_file Sample/Misc_Files/Sample.fam
map_file Sample/Map_Files/Sample_chr1.map
ped_file None
age_file Sample/Misc_Files/Sample_Ages.txt
hap_file None

Parameters
out Sample
num_chr 22
cm_gap 1
disc_homoz 1
likelihood 0.80
mhs_gap 30
po_gap 15
gp_gap 30
trust_fs False
```

- The calculation of haplotype scores is the most computationally intensive step
- If PONDEROSA has already been run, the .haps file can be supplied so that it skips this step