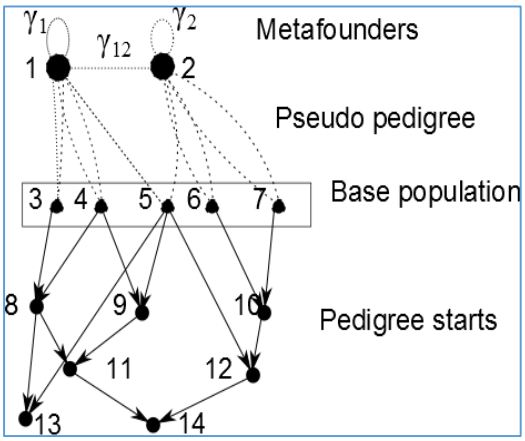
Use of `createHmf` to compute H matrices with metafounders

Software `/save/alegarra/createHmetafounders/createHmf` makes two different tasks.
On input:

a recoded pedigree and genotypes are needed.

- Genotypes must be coded as {0,1,2} and the reference allele must be taken at random so that, in the observed data, the average p across loci is 0.5.
- Pedigree must be renumbered so that parents precede progeny, and at the beginning there are the metafounders. Example from the Genetics paper:




```
Pedigree
1  0  0
2  0  0
3  1  1
4  1  1
5  1  2
```

•

A file with gamma coefficients
This file is required unless Γ is estimated from the data. This file contains the gamma coefficients, then the s parameter. For instance:

```
0.70  0.52
0.52  0.61
23876.4
```

Estimate the parameters $\boldsymbol{\Gamma}$

Three methods are used. In all cases $\boldsymbol{\Gamma} = 8\boldsymbol{\Omega} = 8Cov(\boldsymbol{P})$ where $\boldsymbol{\Omega} = Cov(\boldsymbol{P})$ is the covariance of allelic frequencies across loci in the base populations. $\boldsymbol{P}$ is a matrix with as many rows as markers and as many columns as populations, p_{ij} is the base allelic frequence at locus i and population j. Ω_{ii} is the variance of allelic frequencies in base i and , Ω_{ij} is the covariance of allelic frequencies in population i with allele frequencies in population j. The virtue of using $\boldsymbol{\Gamma}$ instead of using $\boldsymbol{P}$ explicitly is that whereas $\boldsymbol{P}$ is poorly estimated $\boldsymbol{\Gamma}$ is much better estimated. Methods are:

- Naif (crude estimate of marker allelic frequencies ignoring pedigree structure)
- Generalized least squares (GLS), although this is actually Gengler et al. (2007, 2008) BLUP method for prediction of gene content. This method, extended to several populations and their crosses, is, for one locus:

$$\begin{pmatrix} \boldsymbol{Q'Q} & \boldsymbol{Q'W} \\ \boldsymbol{W'Q} & \boldsymbol{W'W + A^{-1}\lambda} \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q'm} \\ \boldsymbol{W'm} \end{pmatrix}$$

- This can be rewritten as:

$$\widehat{\boldsymbol{\mu}}_i = (\boldsymbol{Q'A_{22}^{-1}Q})^{-1}\boldsymbol{Q'A_{22}^{-1}m}_i$$

Where $\boldsymbol{\mu}_i$ is the mean of the gene content, $\boldsymbol{p}_i = \boldsymbol{\mu}_i/2$ is a vector containing allelic frequencies across the different populations, $\boldsymbol{m}_i$ is a vector containing genotypes coded as $\{-1,0,1\}$ and $\boldsymbol{Q}$ is a matrix of population fractions for each individual. Use of $\boldsymbol{A}_{22}^{-1}$ ignores the fact that variances of gene content are heterogeneous across populations, but this is likely to be negligible

(NOTE: it is very easy to extend this to quality control of h2 of gene content!!)

- Maximum likelihood (ML), where $\boldsymbol{P}$ is supposed to come from a normal distribution,
  - in the case of a single population this is $\boldsymbol{p} \sim \boldsymbol{N}\left(\boldsymbol{0}, \boldsymbol{I}\sigma_p^2\right)$. ML is done by an EM algorithm.
  - For several populations, this is $\boldsymbol{P} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{I} \otimes \boldsymbol{P}_0), \boldsymbol{P}_0 = \begin{pmatrix} \sigma^2_{\mu^1\mu^1} & \sigma_{\mu^1\mu^2} & \dots \\ \dots & \sigma^2_{\mu^2\mu^2} & \dots \\ \dots & \dots & \dots \end{pmatrix}$ and $\widehat{\boldsymbol{\Gamma}} = 2\widehat{\boldsymbol{P}}_0$. In this case, the covariance of $\boldsymbol{m}$ is more complex and involves partial relationship matrices (as in Garcia-Cortes and Toro 2006). Here we take a simpler approach in which we assume that for locus i $var(\boldsymbol{m}_i) = A_{22}2\bar{p}_i\bar{q}_i$ where $\bar{p}_i$ is the average allelic frequency across populations.
  -

In Garcia-Baccino simulations with a single population, GLS and ML are almost identical and very accurate.

Files with $\widehat{\boldsymbol{\Gamma}}$ are `estimatedGamma_EM` and `estimatedGamma_GLS`.

Estimate the parameter s

This parameter is such that $\boldsymbol{G}_{0.5} = \boldsymbol{ZZ'}/s$.
In practice its computation is trivial: is just half the number of markers including in the analysis (regardless of whether they are polymorphic or not; this is unimportant because fixed markers add constants to **G**).

Construct the matrix A inverse

The program constructs $\boldsymbol{A}^{(\boldsymbol{\Gamma})-1}$ into file `AiWithMetafounders`.

Construct the matrix H inverse

The program constructs

$$\boldsymbol{H}^{(\Gamma)-1} = \boldsymbol{A}^{(\Gamma)} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{G}_{0.5}^{-1} - \boldsymbol{A}_{22}^{(\Gamma)-1} \end{pmatrix}$$

in file `HiWithMetafounders` . This matrix is in the format (row, col, val), lower stored, so that it can be read by the blupf90 software. At the beginning the program asks some questions.

If $\boldsymbol{\Gamma}$ is not estimated, **a file with gamma coefficients** needs to be provided. This file contains the gamma coefficients, then the s parameter. For instance:

```
0.70 0.52
0.52 0.61
23876
```

Computation of inbreeding

This is achieved by sparse inversion (using FSPAK) of $\boldsymbol{H}^{(\Gamma)-1}$**,** followed by extraction of the diagonal elements. Thus, it may become prohibitive for large matrices. No doubt smarter strategies exist. In file `diagHWithMetafounders` there are 5 columns:
Id, kind (metafounder, genotyped or ungenotyped), (H_{ii}^{Γ} rescaled)-1 , 4th column F^{Γ} rescaled, 5th column F (normal, gamma=0)'.