Short course on

Methods and Tool for Genomic Predictions and GWAS in Breeding Programs

Mehdi Sargolzaei
Select Sires Inc.
University of Guelph

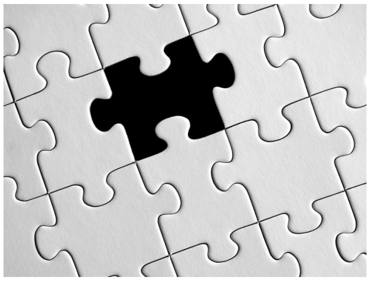
Daniela Lourenco
University of Georgia

20 -24 February 2023, University of New England

Genotype imputation

Missing Data

```
Var1    Var2
----    -----
17.2    1.8
12.5    5.6
?       3.2
21.9    ?
18.1    2.9
13.4    1.6
10.3    6.2
12.4    3.3
11.3    4.0
19.4    ?
```



Missing Data – Issues

- Creates bias in the results
- Can reduce statistical power
- Causes computational complexity and inefficiency
- Modeling incomplete data is more difficult

Data Imputation

Filling in missing data points with the expected values

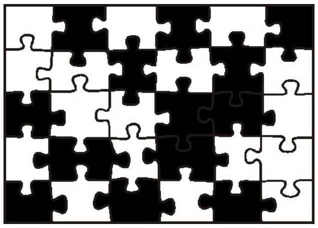
A simple sterategy: Use of average

2.3, **?**, 3.1, 4.7, 1.8, 2.4, 3.3

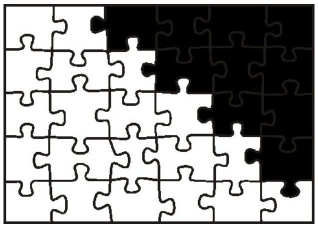
Avg = 2.9

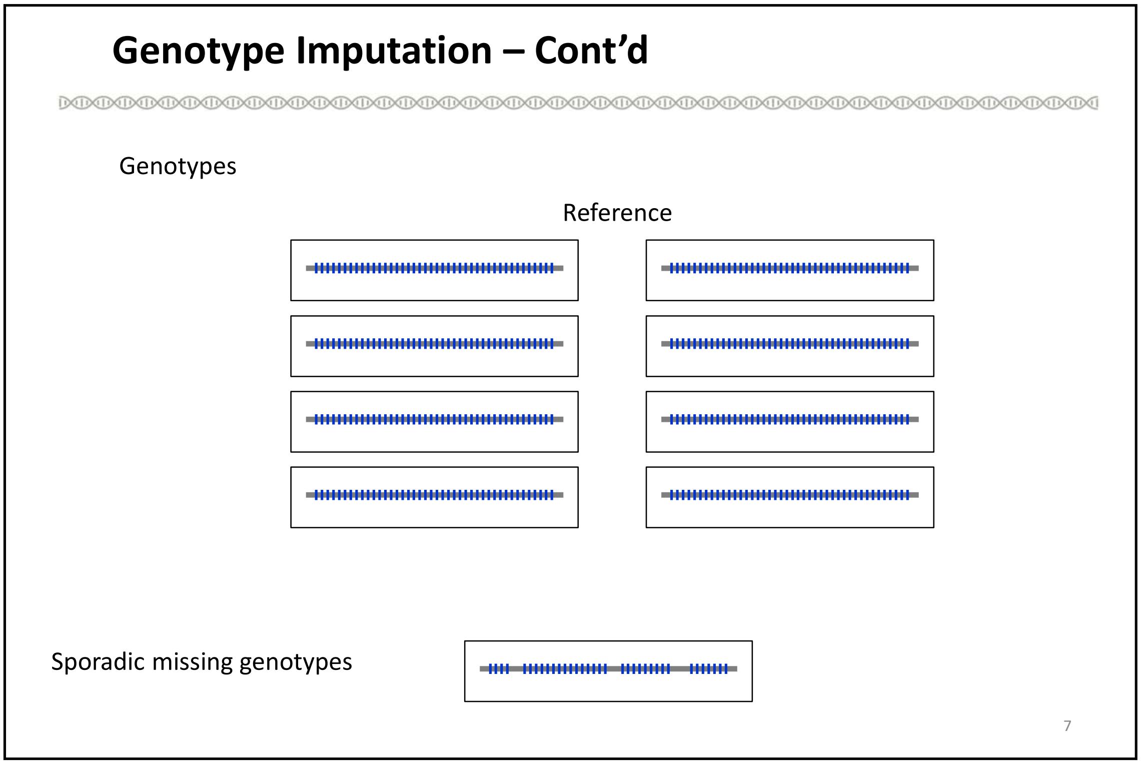
Missing Data Type

- Random



- Structured (not random)





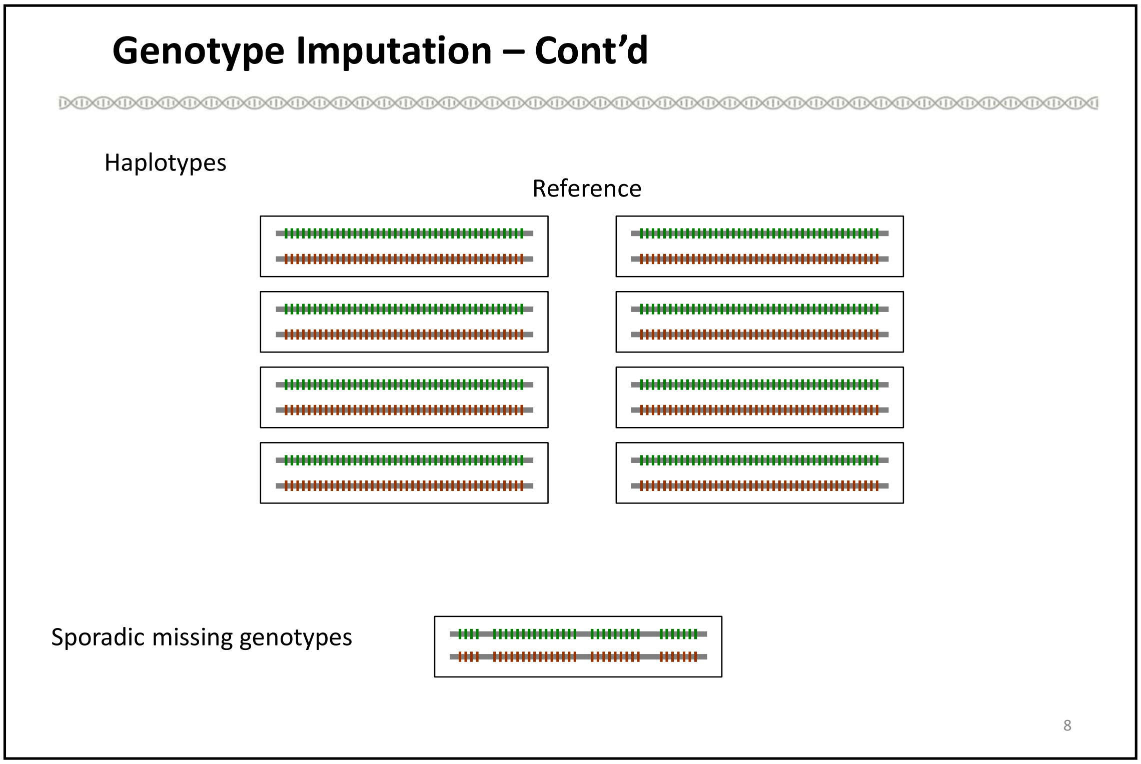
Genotype Imputation – Cont’d

Genotypes

Reference

Sporadic missing genotypes

7



Genotype Imputation – Cont’d

Haplotypes

Reference

Sporadic missing genotypes

8

How Important?

- Increase SNP density and probability of fine mapping
- Increase the power of genome-wide association studies (larger number of samples)
- Increase accuracy of genomic selection
- Decrease computational complexity of dealing with missing data
- Enable us to combine genotype data from different sources with different densities
- Enable us to move from chip genotype to sequence data
- Reduces cost of genotyping in commercial implementation of genomic selection

Quick Look at Genomic Selection

Substantial increase in accuracy of estimated breeding values for young animals with no record and sometime shorter generation interval, …

Main requirements for GS

- Dense marker panel (~50k has been standard so far)
- Large reference population (genotyped + Phenotyped)

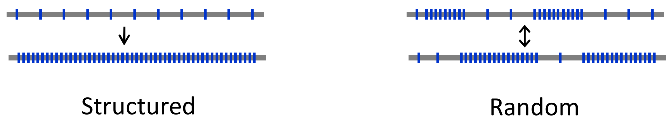
Quick Look at Genomic Selection – Cont’d

Challenge

- Large scale HD genotyping is still expensive
- Animals are usually genotypes with different panels as technology advances (e.g. 3k, 8k, 12k, 19k, 25k, 30k, 50k, 56k, 80k, 100k, 150k, 777k, sequence)

Solution

- Use a smaller, cheaper panel and “impute” to a larger panel; Combine different panels by “imputation”



Quick Look at Genomic Selection – Cont’d

Can we implement cost effective genomic selection without imputation?

- Genomic Selection was officially launched in 2008 in USA
- The initial and main use of GS was to select bulls
- With the adoption of GS and advancement in genotyping technology, the cost of genotyping started to decline
- Currently after 14 years, only 10-20% of active cows in US are genotyped!!!

Quick Look at Genomic Selection – Cont’d

	One animal	20,000 animals
Genotyping (800k)	~$100	$2,000,000
Genotyping (50k/80k)	~$50	$1,000,000
Genotyping (6k/8k/12k/19k)	~$30-40	$ 600,000

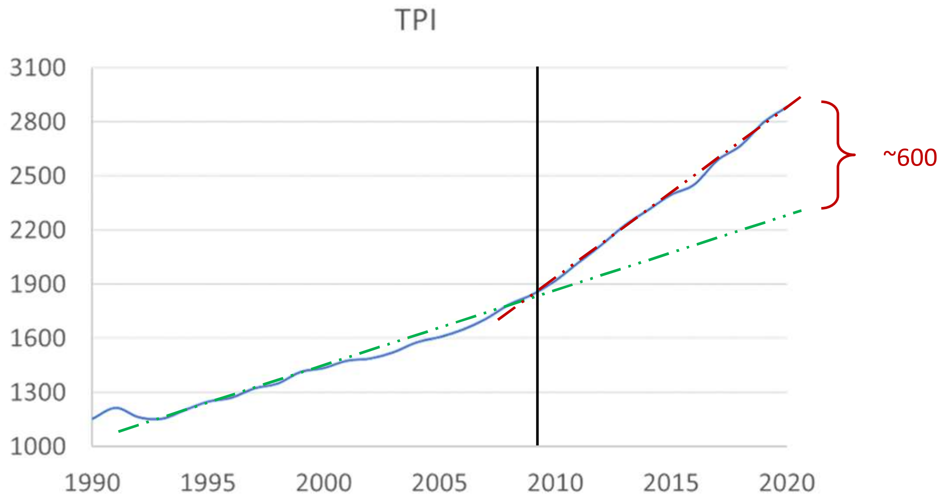
Prices are estimates

Substantial saving with low-density panel

13

13

Impact of Genomic Selection in Dairy Cattle

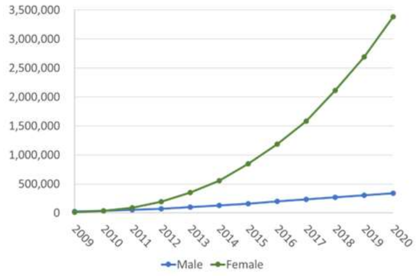


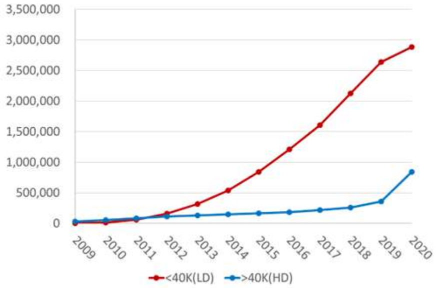
Genetic progress in US dairy cattle.

14

14

Impact of Genomic Selection in Dairy Cattle





- Significant drop in genotyping cost
- Better herd management with genomic info on cows

CDCB, June 2021

Genotype Imputation

Process of predicting and filling in the missing SNP genotypes using information from family or a reference population

Two key factors for imputation:

- Linkage disequilibrium (Association between SNPs)
- Relationship between individuals

Genotype Imputation

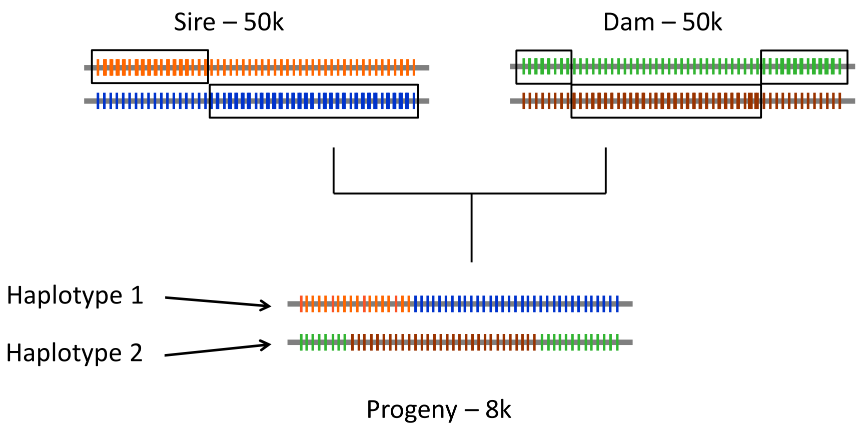
Groups:

- Reference group (e.g. genotyped with HD panel)
- Target group (e.g. genotyped with LD panel)

Steps:

- Estimating haplotype phase
- Finding shared haplotype blocks between reference and target groups
- Filling in information from the reference group

Genotype Imputation



Sire – 50k

Dam – 50k

Haplotype 1

Haplotype 2

Progeny – 8k

Genotype Imputation

Methods

- Random imputation based on allele frequency
- Prediction based on genotype similarity
- Prediction based on haplotype similarity

Most of prediction methods are based on a Hidden Markov Model (HMM)

Genotype Imputation

Haplotype similarity-based methods

Family-based imputation

- Mainly uses linkage information

Population-based imputation

- Mainly uses linkage disequilibrium information

Combined family & population imputation

- Both linkage and linkage disequilibrium information is used

Linkage or Linkage Disequilibrium?

- Linkage analysis information may be considered as long-range LD within family
- LD is a term used for population

Linkage or Linkage Disequilibrium?

LD is non-random association of alleles at two loci in population

LA is actually LD within family

Decay of LD over generations is a function of distance between the markers

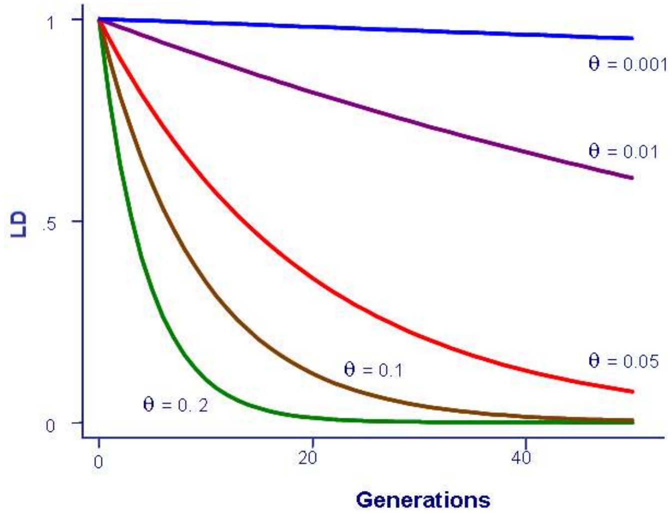
LD in population is created by:

- Drift (small effective population size)
- Selection

Main causes in livestock populations

- Mutation
- Migration
- Crossing

Closer Look at LD



Closer Look at LD

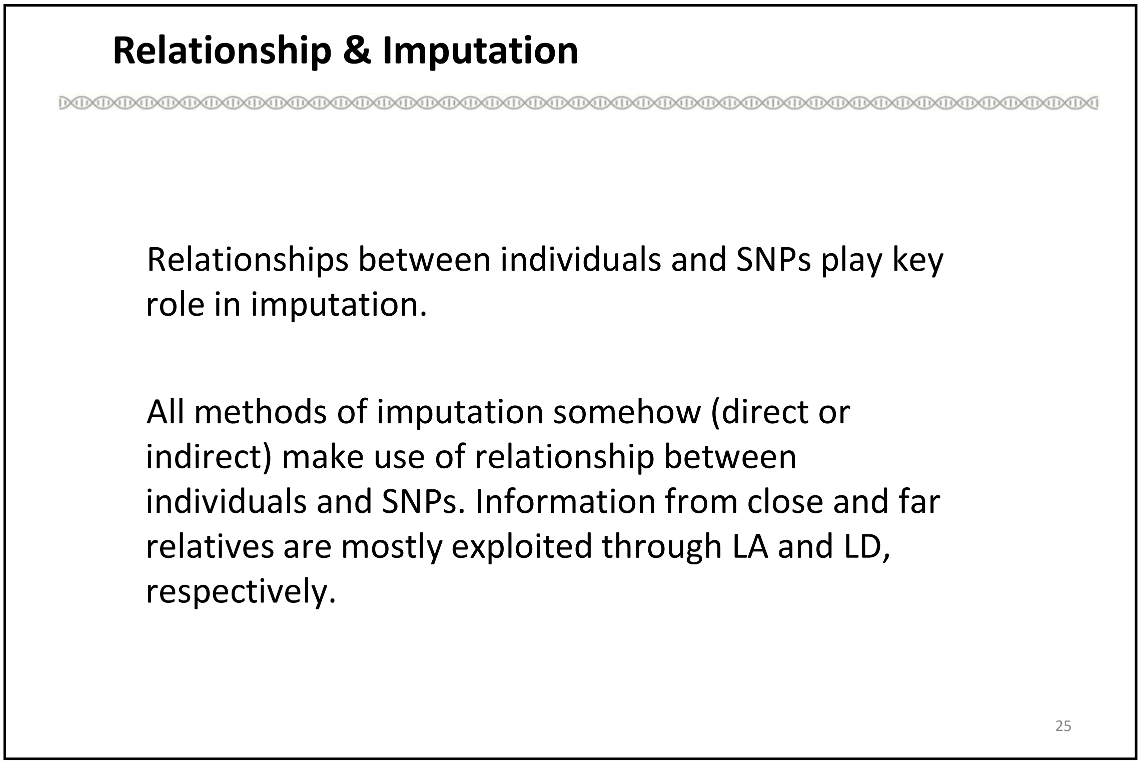
- Close relatives share long haplotypes
- Distant relatives share short haplotypes

↓

Length of shared haplotypes between two individuals stores information about the genetic relationship

↓

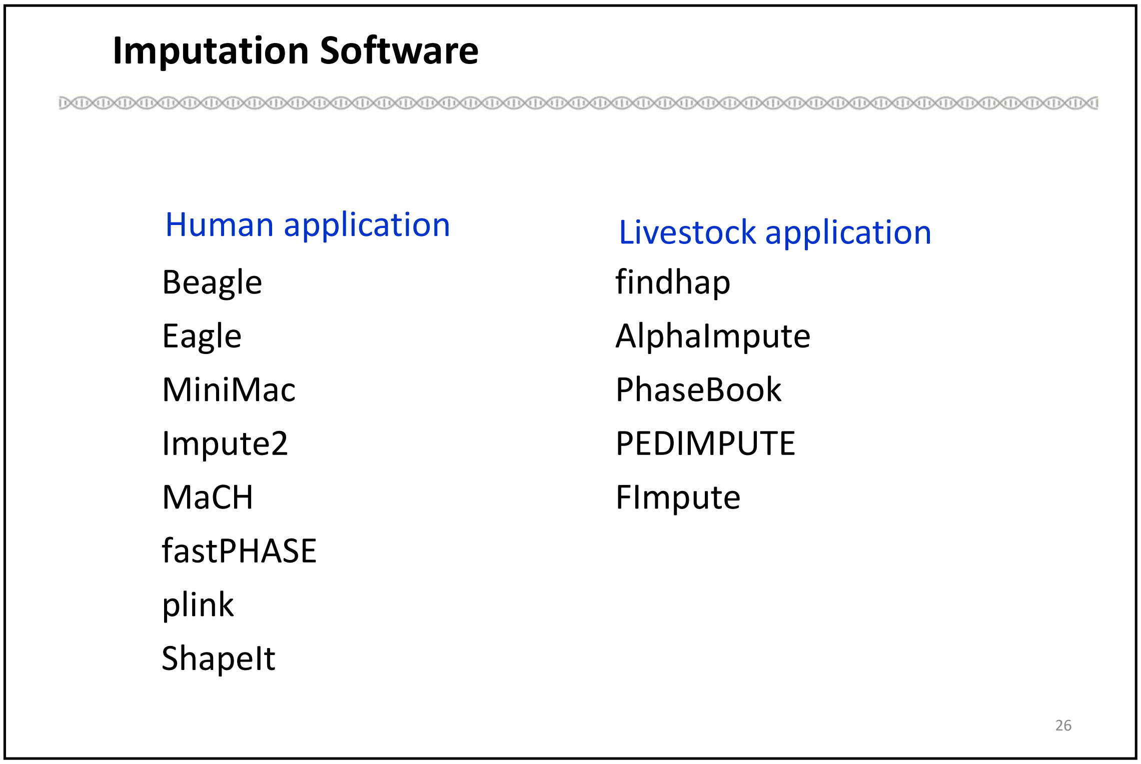
Age of relationship (Meuwissen et al. 2014)



Relationship & Imputation

Relationships between individuals and SNPs play key role in imputation.

All methods of imputation somehow (direct or indirect) make use of relationship between individuals and SNPs. Information from close and far relatives are mostly exploited through LA and LD, respectively.



Imputation Software

Human application

Beagle
Eagle
MiniMac
Impute2
MaCH
fastPHASE
plink
ShapeIt

Livestock application

findhap
AlphaImpute
PhaseBook
PEDIMPUTE
FImpute

Performance Issue!

Most of imputation methods use the hidden Markov model to calculate a posterior distribution

- Assume individuals are not related
- Computationally demanding
- Usually cannot handle very large data set

Imputation Method Implemented in FImpute

- Assumption: All individuals are related
- Real data usually shows a wide range of relationships between individuals from parent-progeny to individuals that are separated by many generations.
- Close relatives share longer haplotypes that have lower frequency and distant relatives share shorter haplotypes which usually have higher frequency
- Imputation and phasing are more accurate when using information from close relatives (i.e. long haplotypes with usually low frequency)

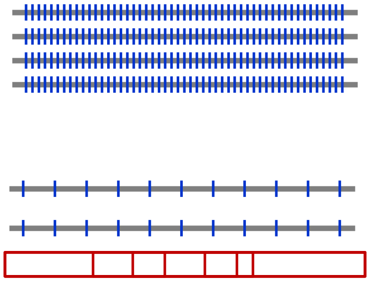
Imputation Method Implemented in FImpute

One effective phasing / imputation strategy is to exploit the genealogy or relationships between individuals by searching for shared haplotypes from the longest to the shortest.

29

Imputation Method Implemented in FImpute

Overlapping sliding windows approach:



Use family information for phasing if available

30

Imputation Method Implemented in FImpute

Validation Study

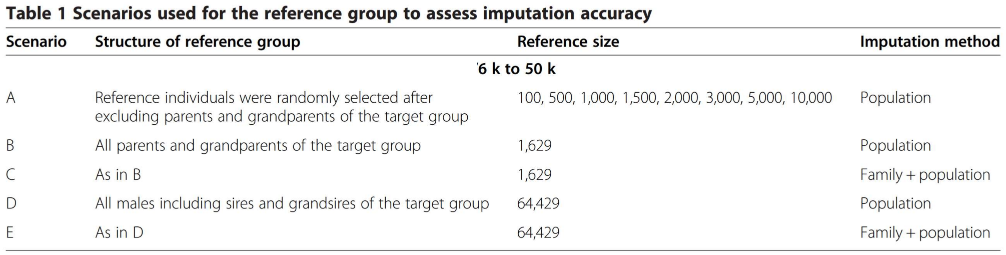
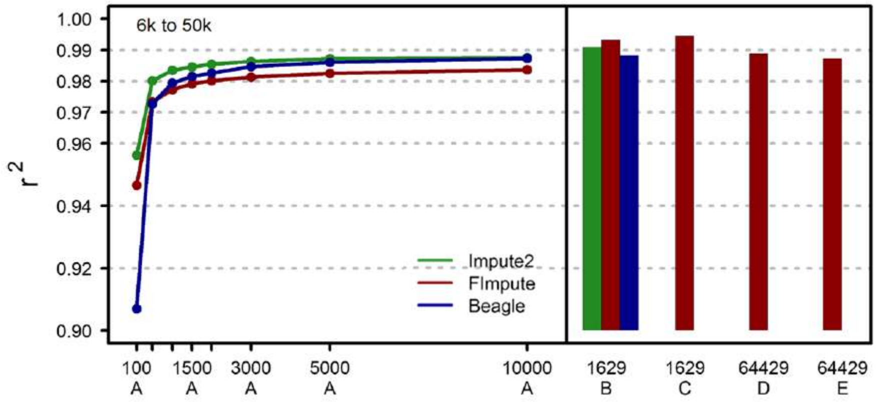


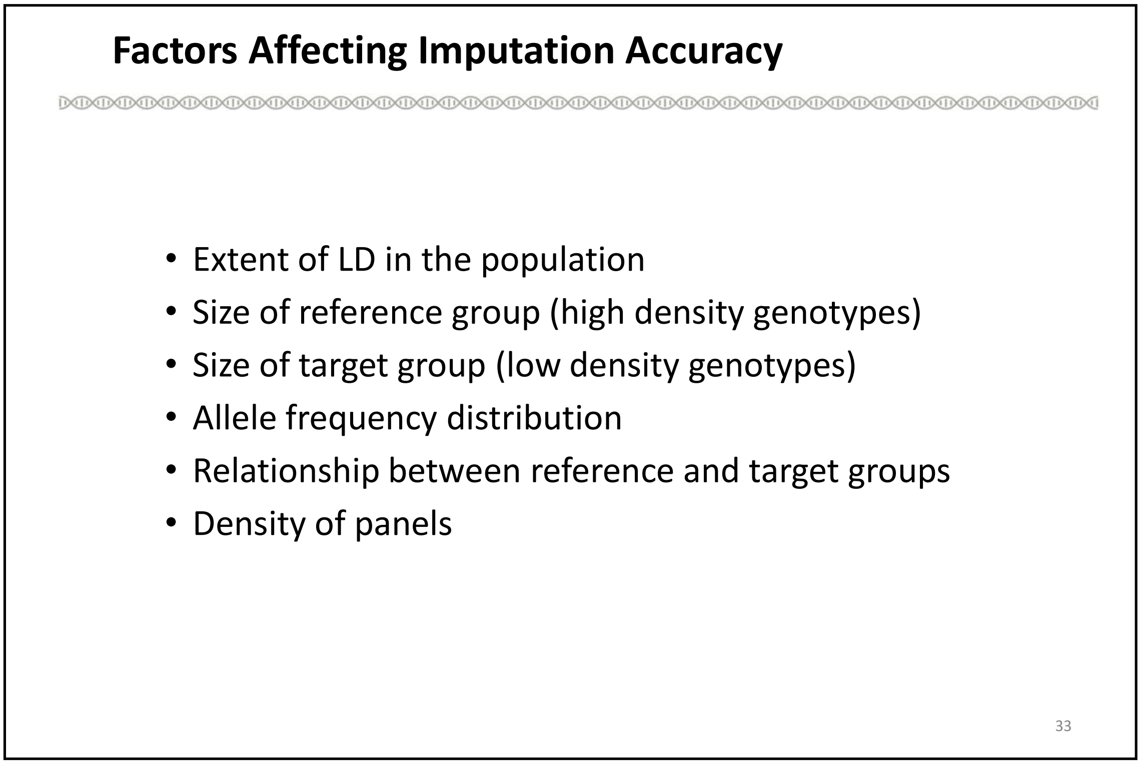
Table 1 Scenarios used for the reference group to assess imputation accuracy

Scenario	Structure of reference group	Reference size	Imputation method
		6 k to 50 k	
A	Reference individuals were randomly selected after excluding parents and grandparents of the target group	100, 500, 1,000, 1,500, 2,000, 3,000, 5,000, 10,000	Population
B	All parents and grandparents of the target group	1,629	Population
C	As in B	1,629	Family + population
D	All males including sires and grandsires of the target group	64,429	Population
E	As in D	64,429	Family + population

Imputation Method Implemented in FImpute

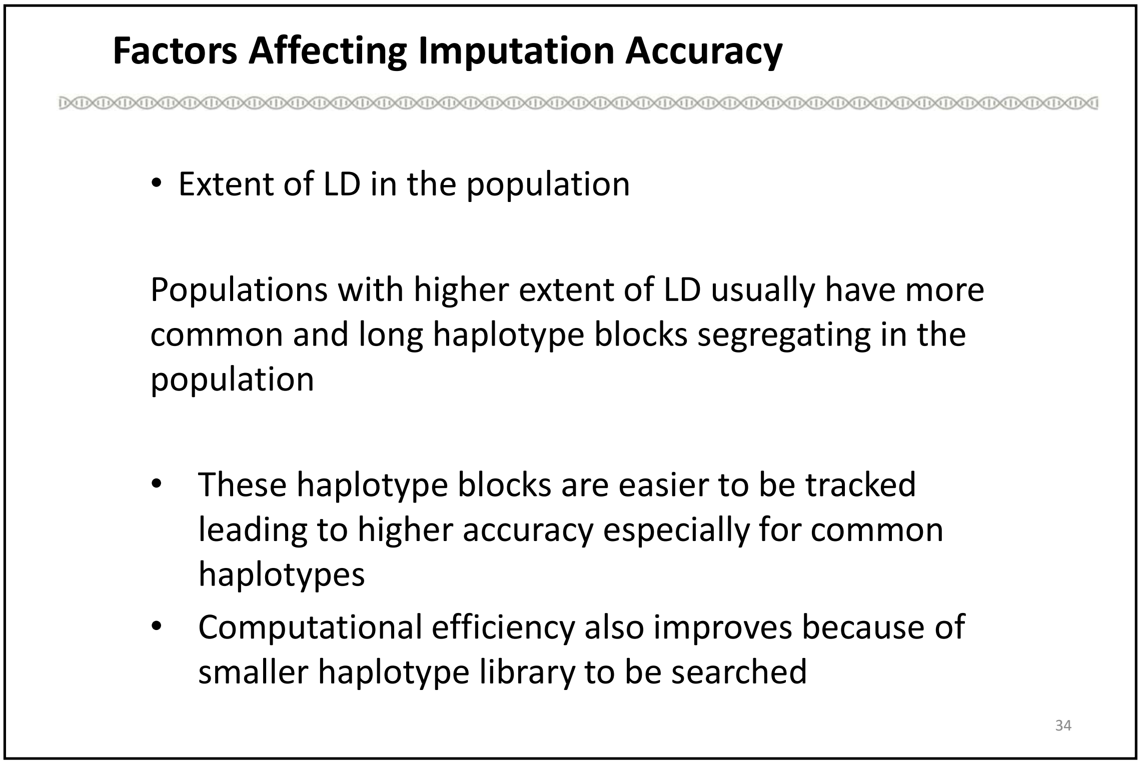


Reference size and imputation scenarios



Factors Affecting Imputation Accuracy

- Extent of LD in the population
- Size of reference group (high density genotypes)
- Size of target group (low density genotypes)
- Allele frequency distribution
- Relationship between reference and target groups
- Density of panels



Factors Affecting Imputation Accuracy

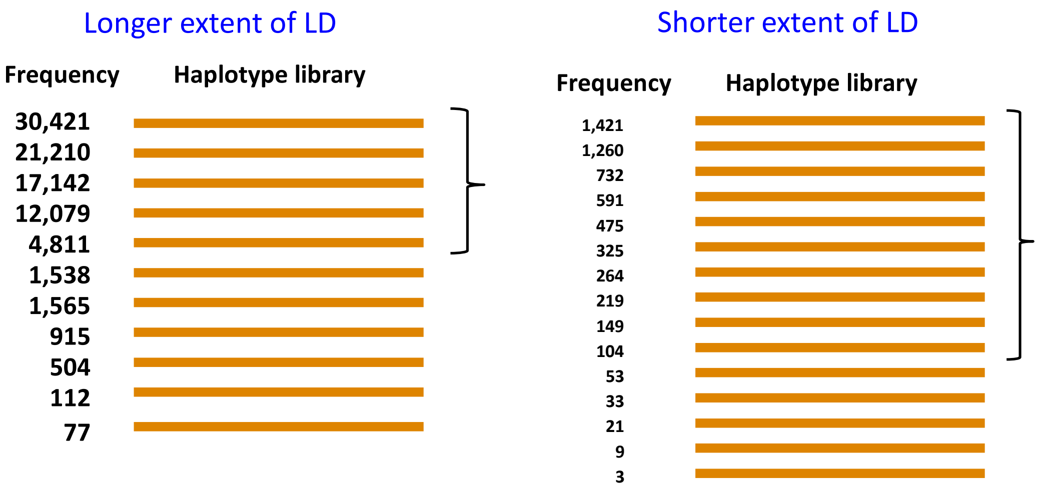
- Extent of LD in the population

Populations with higher extent of LD usually have more common and long haplotype blocks segregating in the population

- These haplotype blocks are easier to be tracked leading to higher accuracy especially for common haplotypes
- Computational efficiency also improves because of smaller haplotype library to be searched

Factors Affecting Imputation Accuracy

- Extent of LD in the population



Factors Affecting Imputation Accuracy

- Size of target group (low density genotypes)

- Accurate phasing of target group is important for high accuracy imputation
- Size of the target group is less important when size of reference group is large

  Low density genotypes from reference group is normally used to phase genotypes of target group

Factors Affecting Imputation Accuracy

- Allele frequency distribution

  - SNPs with very low allele frequency usually have lower accuracy
  - Non-segregating SNP in reference group are non-informative and can create add noise

There is ascertainment bias in most of microarray chips: More informative and segregating SNP are usually placed on chip?

Factors Affecting Imputation Accuracy

- Relationship between reference and target groups

Relatives share longer haplotypes which result in higher imputation accuracy when they are represented in both high and low density groups

A good example is a sire-progeny pair, when sire has HD genotype and progeny is on LD

Factors Affecting Imputation Accuracy

- Density of panels

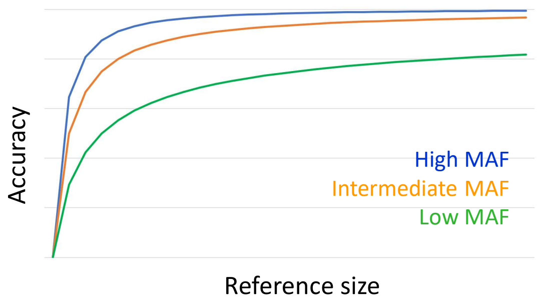
- The density of LD is more important for imputation accuracy
- The density of LD panel has implication on the method
  - Family imputation becomes more important when LD panel is very sparse (e.g., 3k panel in dairy cattle)
  - Family imputation can be less relevant when LD panel is dense enough (e.g., Roughly denser than 10k in dairy cattle)

Factors Affecting Imputation Accuracy

- Size of reference group (high density genotypes)

The optimal size of reference population mainly depend on allele frequency distribution

Larger reference size is needed when more SNPs have low MAF



Genotype Quality Check Before and After Imputation

- Excess of heterozygosity
- Minor allele frequency check
- Call rate for SNPs and animals
- Correlation of allele frequencies between reference animals and target animals
- Masking strategy for after-imputation check
- Mendelian inconsistency with parents both SNP and animals
- Local haplotype diversity test
- Checking for outliers (e.g high homozygosity rate)

Assessing Imputation Accuracy

Validation study:

- HD genotypes are known
- Only LD genotypes is available

In addition to assessing accuracy of imputation, validation study can be used to quantify the effect of factors influencing imputation accuracy like size of reference group.

Assessing Imputation Accuracy

Validation study using all HD genotypes:

- Separate the genotypes into reference and target groups:
  - Random
  - Young and old
- Mask genotype of target group (keep the LD genotypes and remove the rest)
- Perform imputation
- Compare imputed genotypes of target group with the original genotypes

Assessing Imputation Accuracy

Validation study using all HD genotypes - Issues:

- The calculated accuracy can have high error variance because accuracy may depends on how the data separated into reference and target groups

K-fold cross validation

- For small size samples, the size of target group may influence the validation power

Leave-one-out cross validation

Assessing Imputation Accuracy

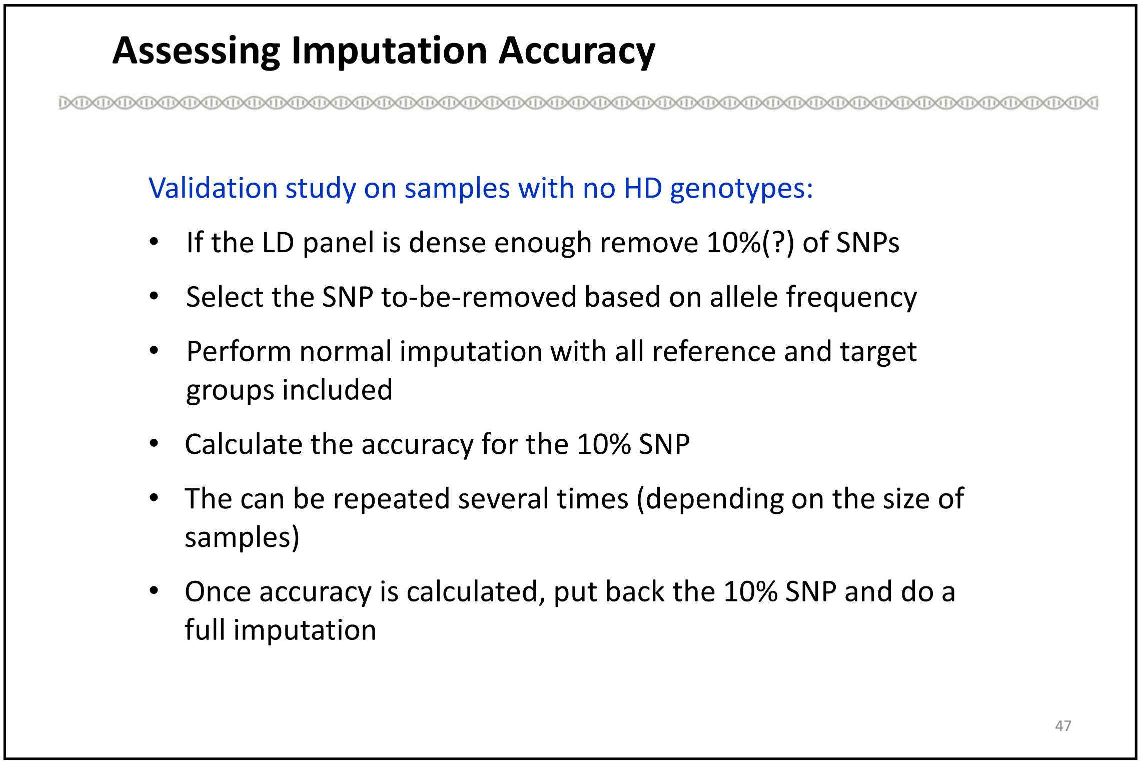
K-fold cross validation:

- Divide the data into K subgroups
- Run the validation K times
- Take the average of K accuracy measures

Assessing Imputation Accuracy

Leave-one-out cross validation:

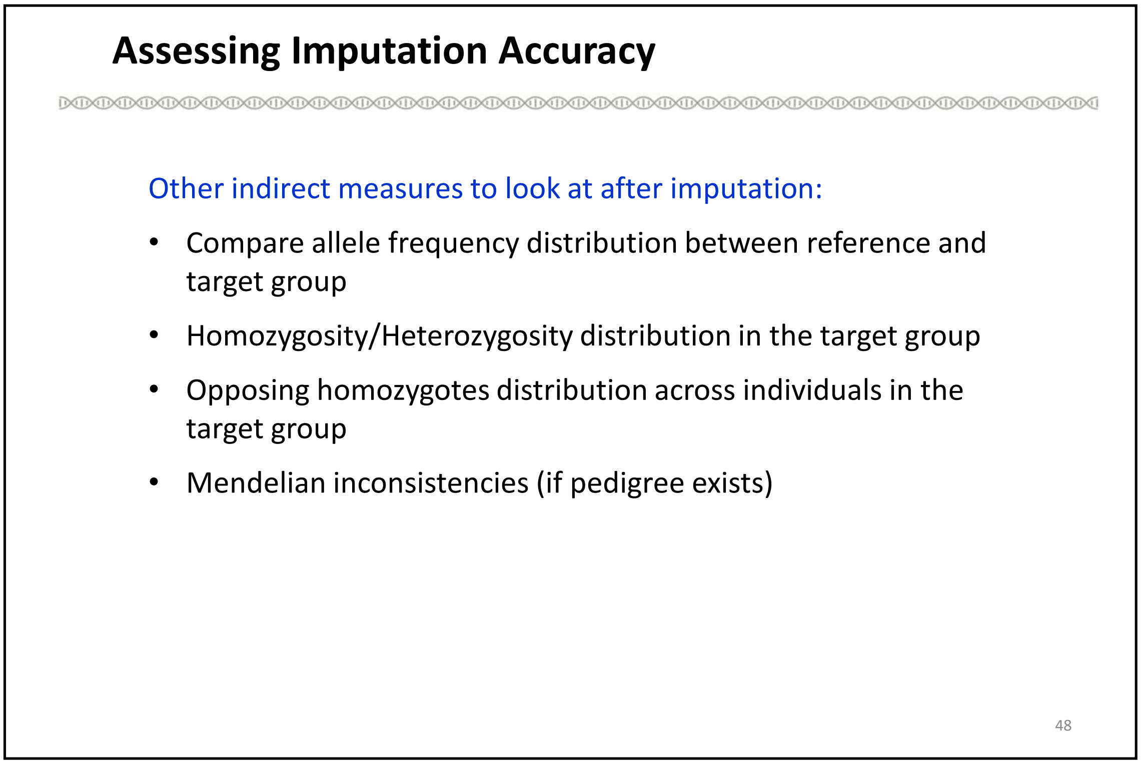
- Similar to K-fold cross validation (K equals to N)
- Select one sample as target at a time
- Run the validation N times
- Take the average of N accuracy measures



Assessing Imputation Accuracy

Validation study on samples with no HD genotypes:

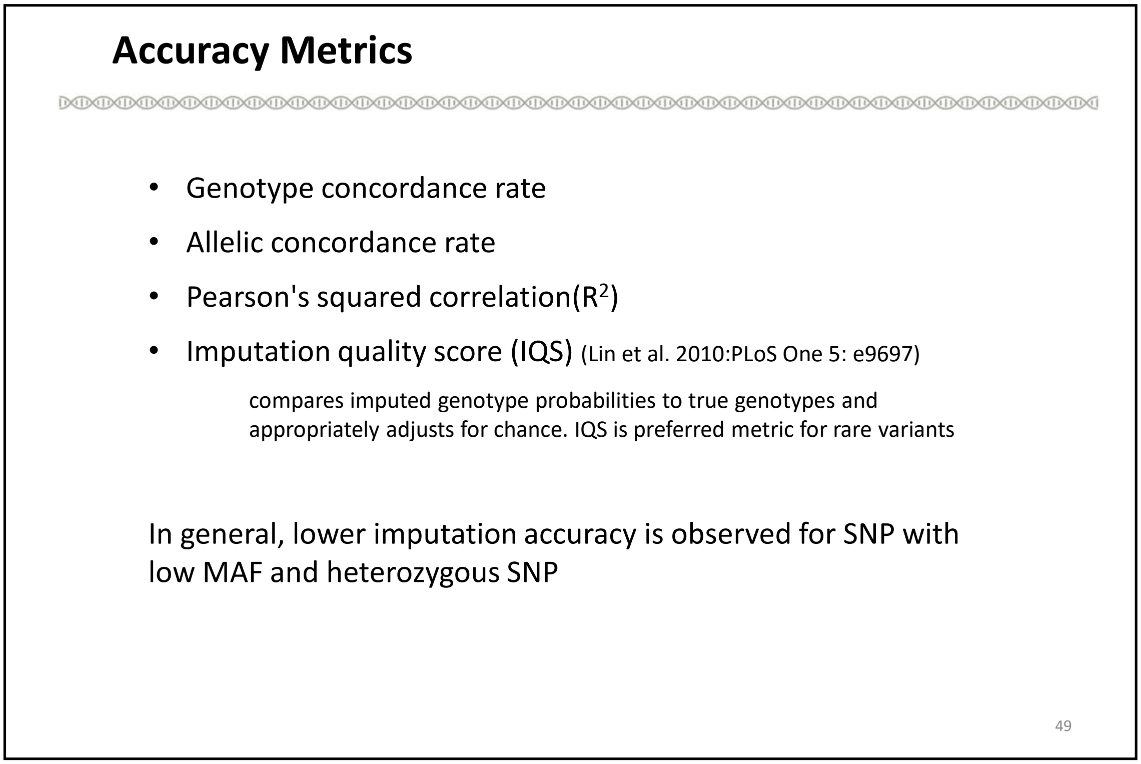
- If the LD panel is dense enough remove 10%(?) of SNPs
- Select the SNP to-be-removed based on allele frequency
- Perform normal imputation with all reference and target groups included
- Calculate the accuracy for the 10% SNP
- The can be repeated several times (depending on the size of samples)
- Once accuracy is calculated, put back the 10% SNP and do a full imputation



Assessing Imputation Accuracy

Other indirect measures to look at after imputation:

- Compare allele frequency distribution between reference and target group
- Homozygosity/Heterozygosity distribution in the target group
- Opposing homozygotes distribution across individuals in the target group
- Mendelian inconsistencies (if pedigree exists)

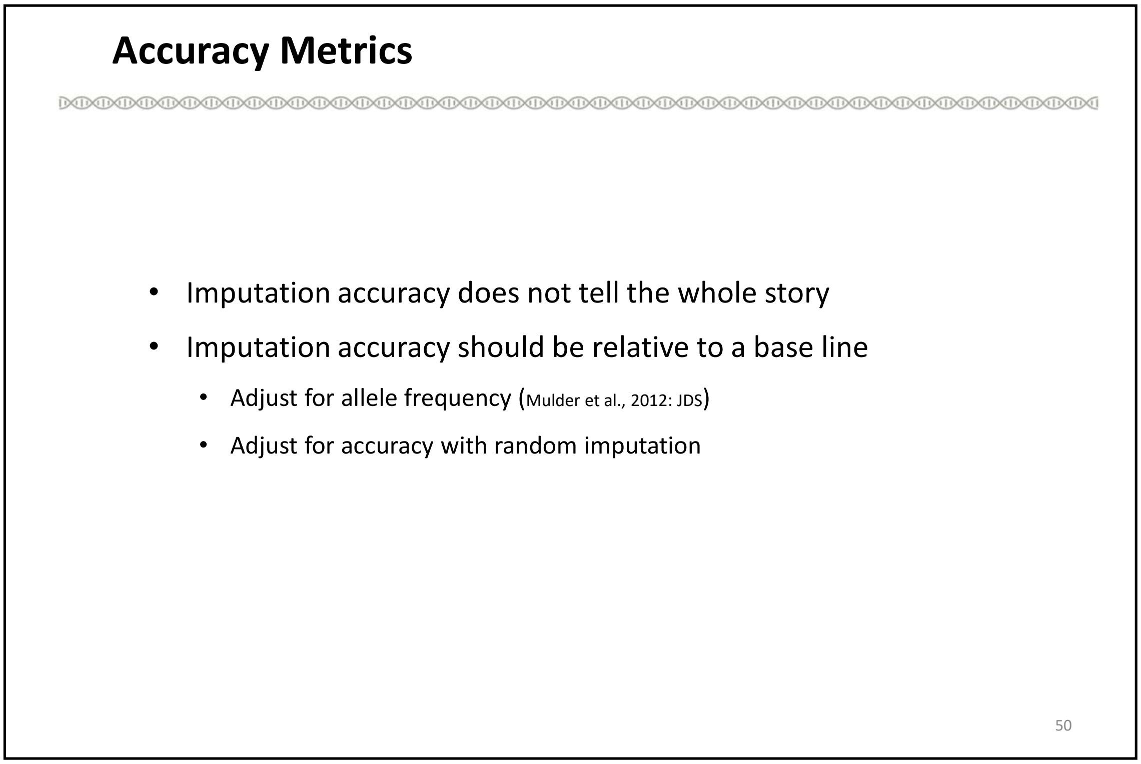


Accuracy Metrics

- Genotype concordance rate
- Allelic concordance rate
- Pearson's squared correlation(R^2)
- Imputation quality score (IQS) (Lin et al. 2010:PLoS One 5: e9697)

  compares imputed genotype probabilities to true genotypes and appropriately adjusts for chance. IQS is preferred metric for rare variants

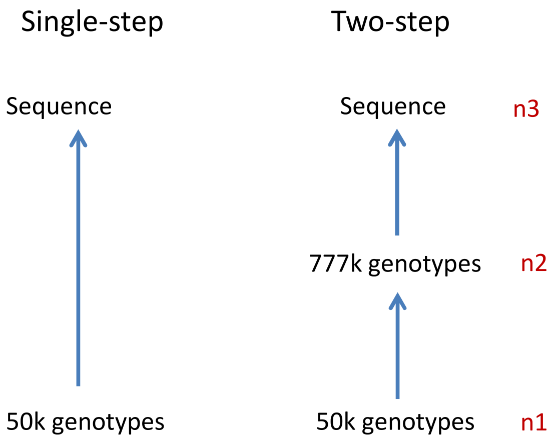
In general, lower imputation accuracy is observed for SNP with low MAF and heterozygous SNP



Accuracy Metrics

- Imputation accuracy does not tell the whole story
- Imputation accuracy should be relative to a base line
  - Adjust for allele frequency (Mulder et al., 2012: JDS)
  - Adjust for accuracy with random imputation

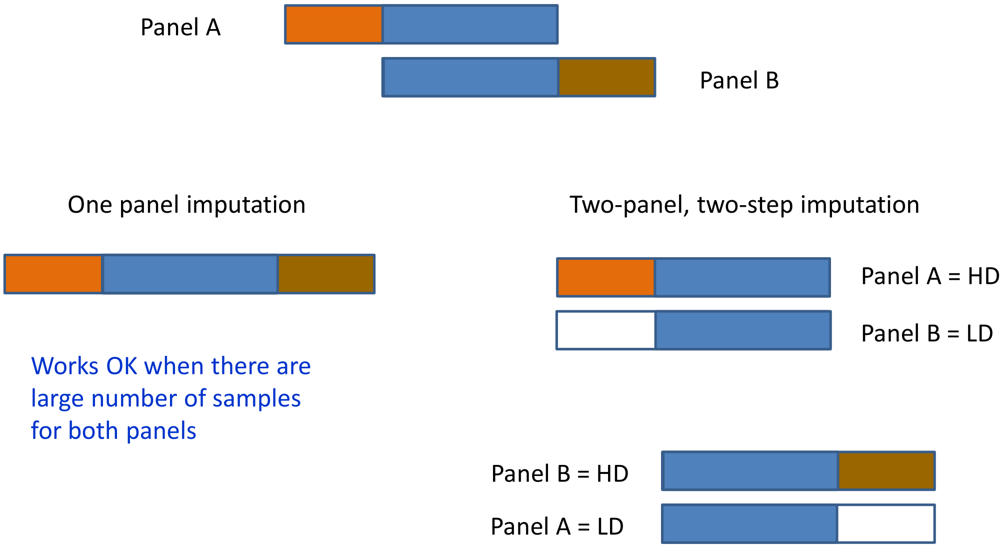
Two-Step Imputation



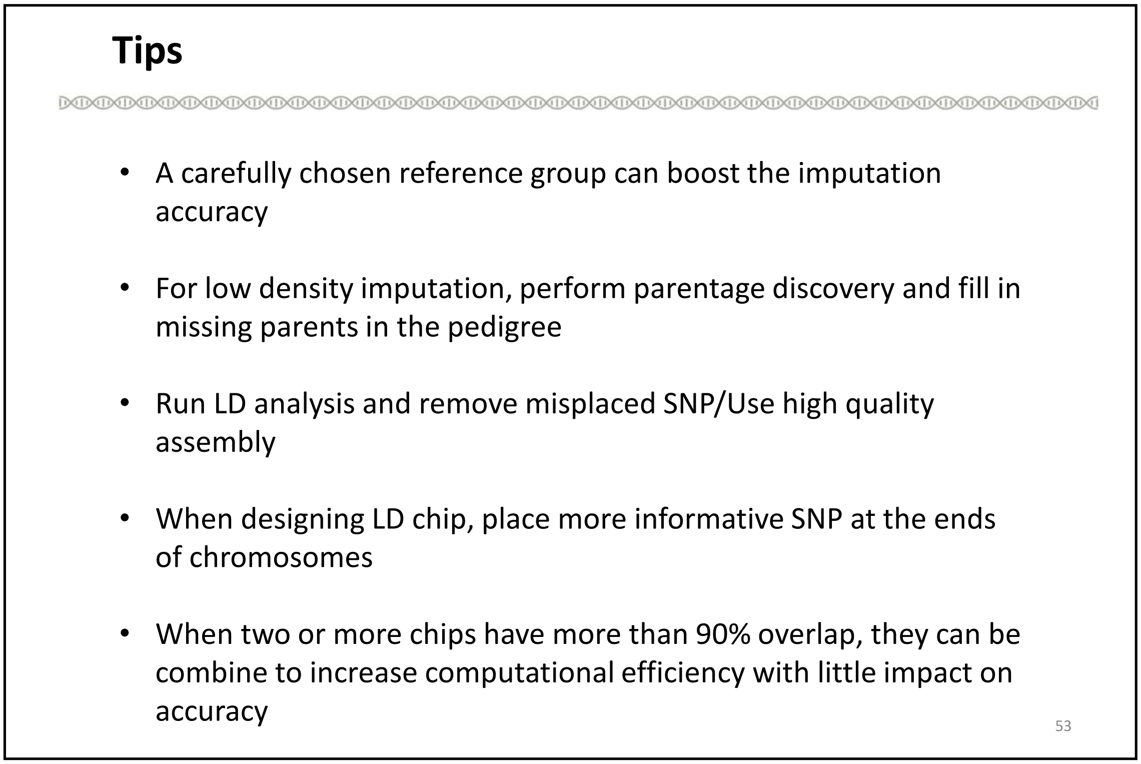
- Increase accuracy of variants with low MAF
- Increase accuracy of heterozygous variants
- Increase overall accuracy

n1 > n2 > n3   Other factors also important

Two-Step Imputation (2)

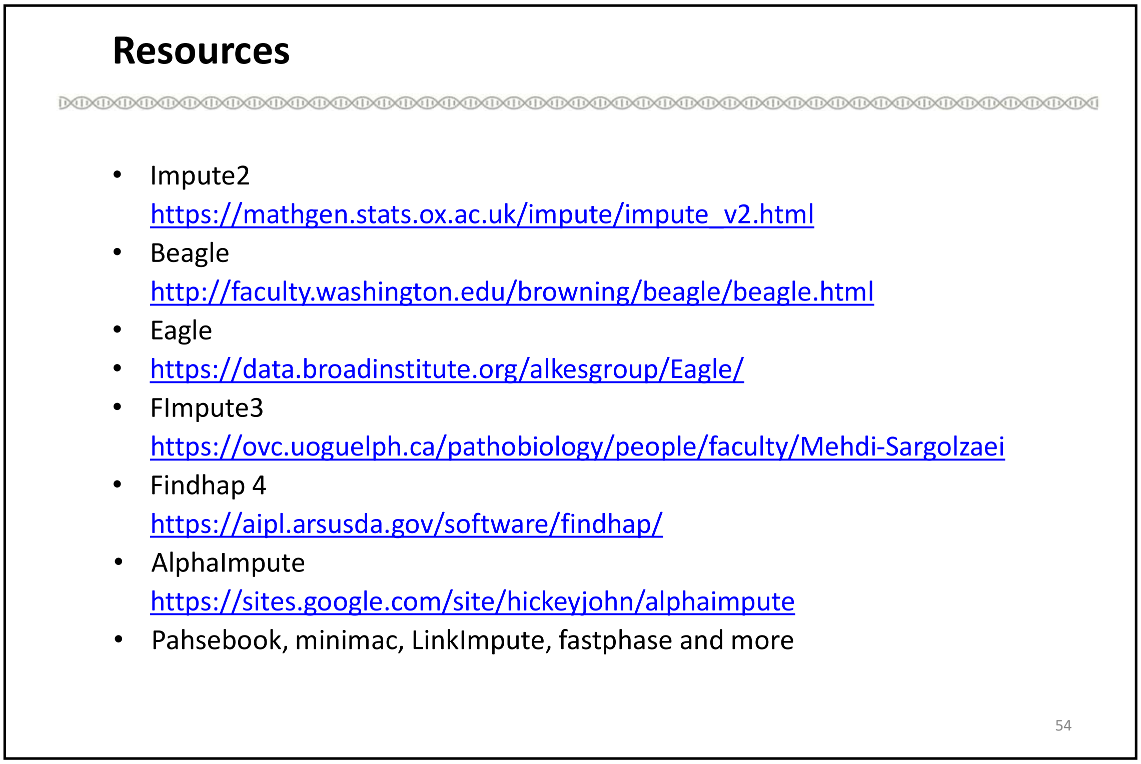


Works OK when there are large number of samples for both panels



Tips

- A carefully chosen reference group can boost the imputation accuracy
- For low density imputation, perform parentage discovery and fill in missing parents in the pedigree
- Run LD analysis and remove misplaced SNP/Use high quality assembly
- When designing LD chip, place more informative SNP at the ends of chromosomes
- When two or more chips have more than 90% overlap, they can be combine to increase computational efficiency with little impact on accuracy



Resources

- Impute2
  https://mathgen.stats.ox.ac.uk/impute/impute_v2.html
- Beagle
  http://faculty.washington.edu/browning/beagle/beagle.html
- Eagle
- https://data.broadinstitute.org/alkesgroup/Eagle/
- FImpute3
  https://ovc.uoguelph.ca/pathobiology/people/faculty/Mehdi-Sargolzaei
- Findhap 4
  https://aipl.arsusda.gov/software/findhap/
- AlphaImpute
  https://sites.google.com/site/hickeyjohn/alphaimpute
- Pahsebook, minimac, LinkImpute, fastphase and more

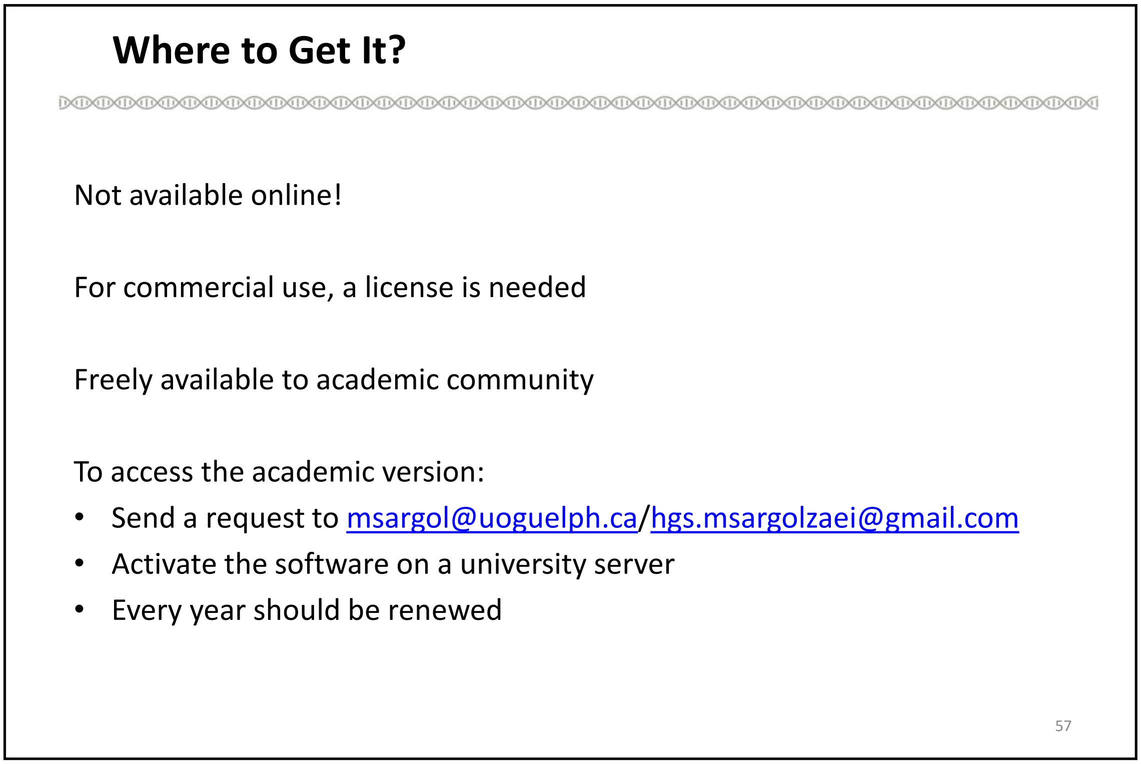
FImpute3

FImpute – Family Imputation

Designed for:

- livestock populations
- Large scale genotype imputation
- Family and population imputation
- Exploiting relationship between individuals

- Accurate: Especially for population structure like livestock
- Computationally efficient



Where to Get It?

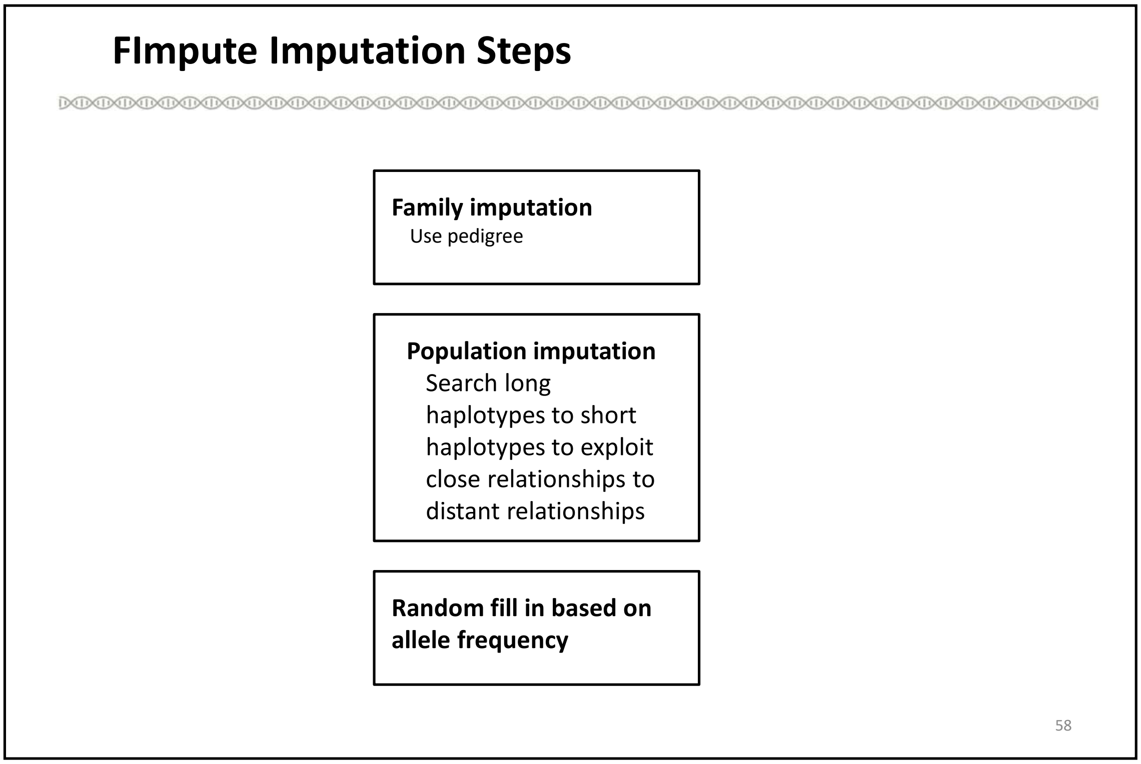
Not available online!

For commercial use, a license is needed

Freely available to academic community

To access the academic version:

- Send a request to msargol@uoguelph.ca/hgs.msargolzaei@gmail.com
- Activate the software on a university server
- Every year should be renewed



FImpute Imputation Steps

Family imputation
Use pedigree

Population imputation
Search long haplotypes to short haplotypes to exploit close relationships to distant relationships

Random fill in based on allele frequency

Data Requirement

Mandatory:

- Genotype file
- Map file

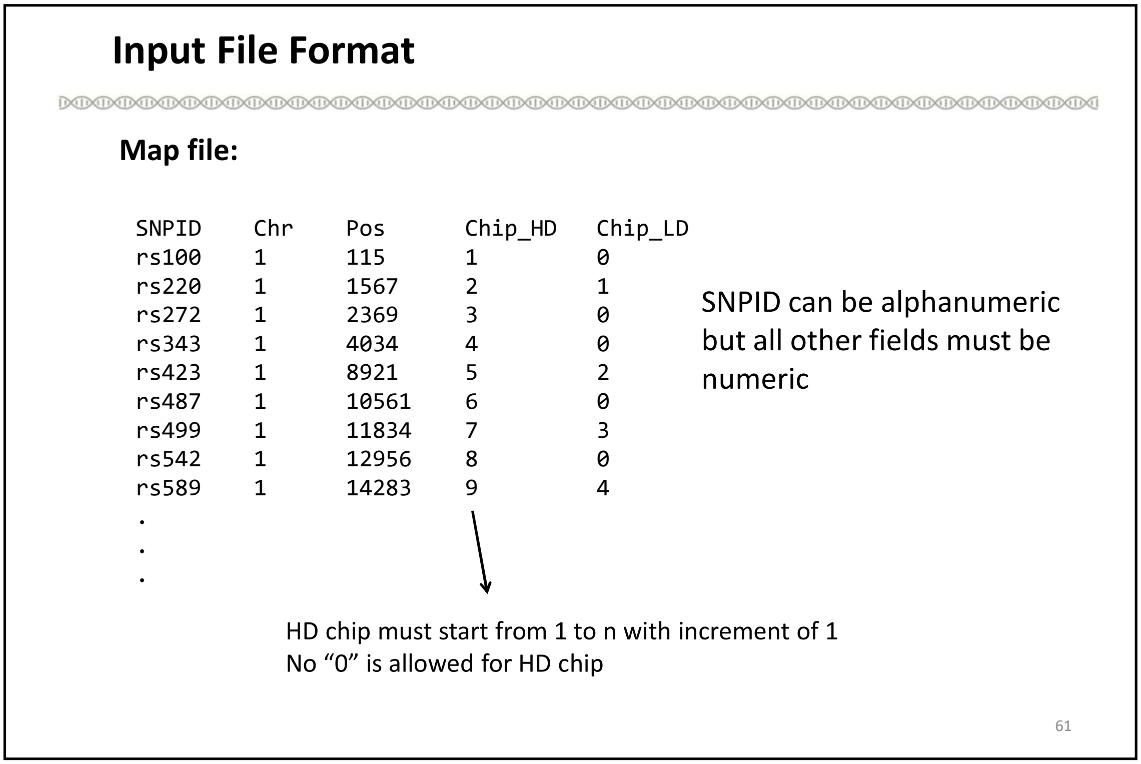
Optional:

- Pedigree file (Family imputation)

FImpute Control File Example

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
ped_file = "ped.txt";
output_folder = "output";
```

Used for family imputation and correction of genotyping errors



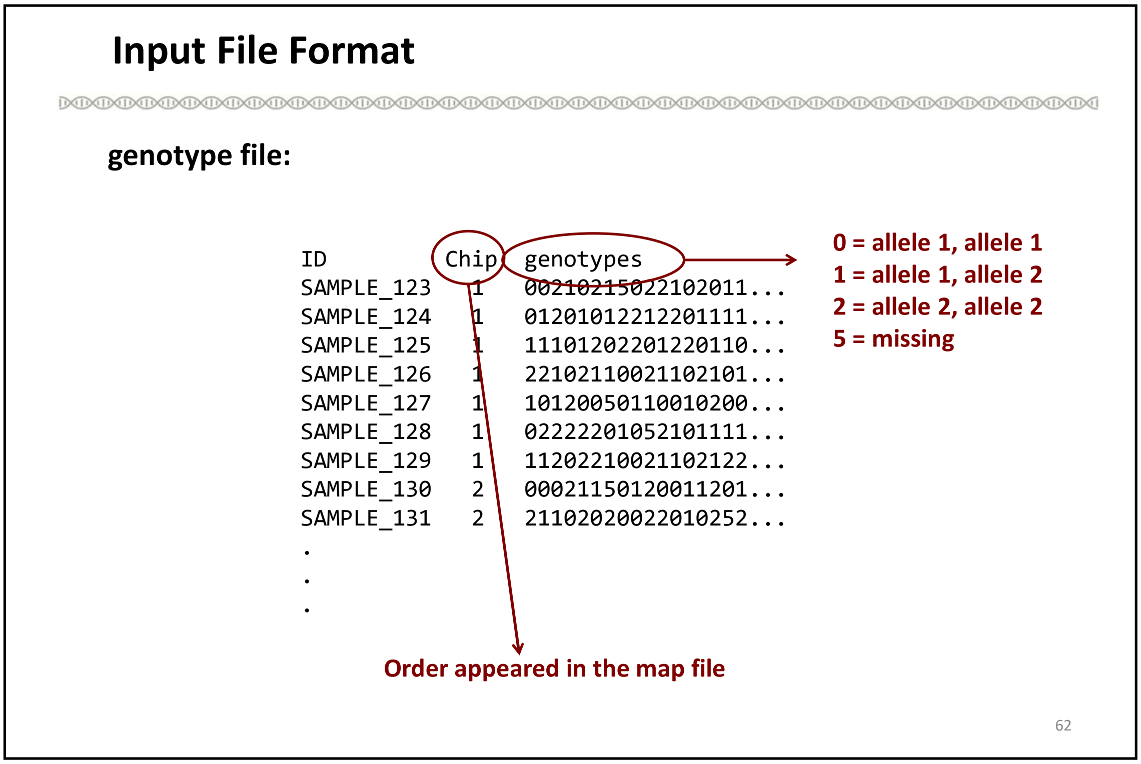
Input File Format

Map file:

```
SNPID    Chr    Pos      Chip_HD    Chip_LD
rs100    1      115      1          0
rs220    1      1567     2          1
rs272    1      2369     3          0
rs343    1      4034     4          0
rs423    1      8921     5          2
rs487    1      10561    6          0
rs499    1      11834    7          3
rs542    1      12956    8          0
rs589    1      14283    9          4
.
.
.
```

SNPID can be alphanumeric but all other fields must be numeric

HD chip must start from 1 to n with increment of 1
No “0” is allowed for HD chip



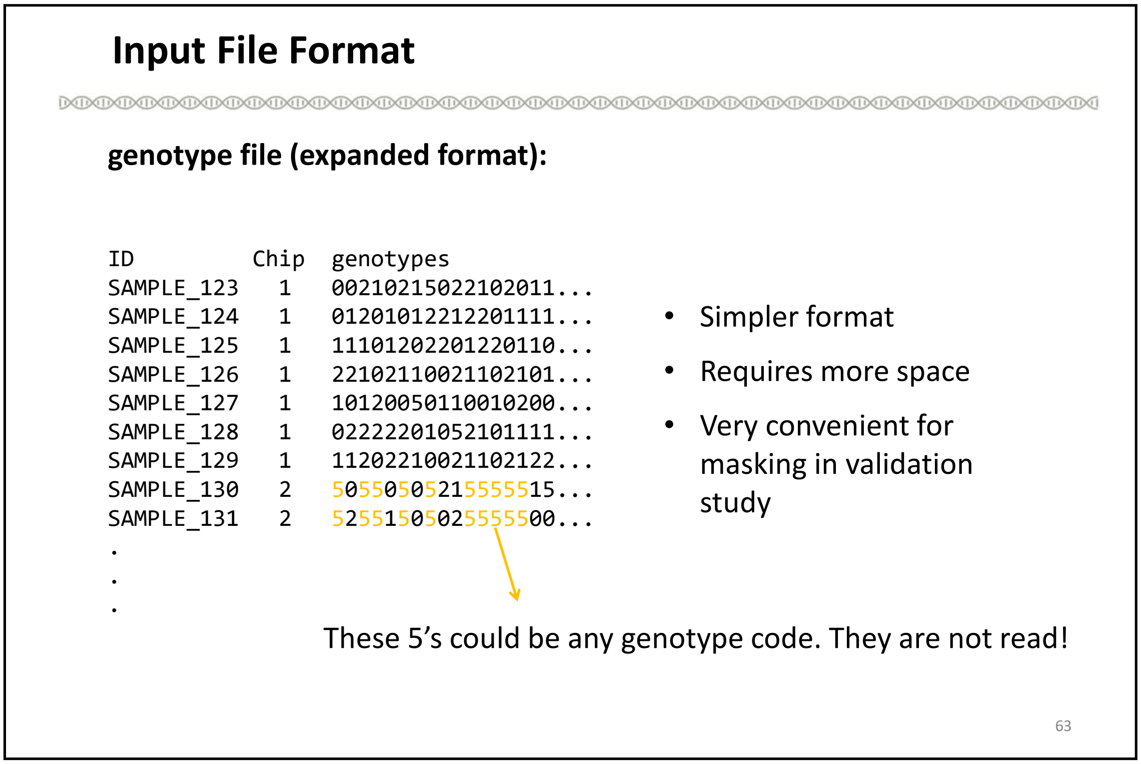
Input File Format

genotype file:

```
ID            Chip   genotypes
SAMPLE_123    1      00210215022102011...
SAMPLE_124    1      01201012212201111...
SAMPLE_125    1      11101202201220110...
SAMPLE_126    1      22102110021102101...
SAMPLE_127    1      10120050110010200...
SAMPLE_128    1      02222201052101111...
SAMPLE_129    1      11202210021102122...
SAMPLE_130    2      00021150120011201...
SAMPLE_131    2      21102020022010252...
.
.
.
```

0 = allele 1, allele 1
1 = allele 1, allele 2
2 = allele 2, allele 2
5 = missing

Order appeared in the map file



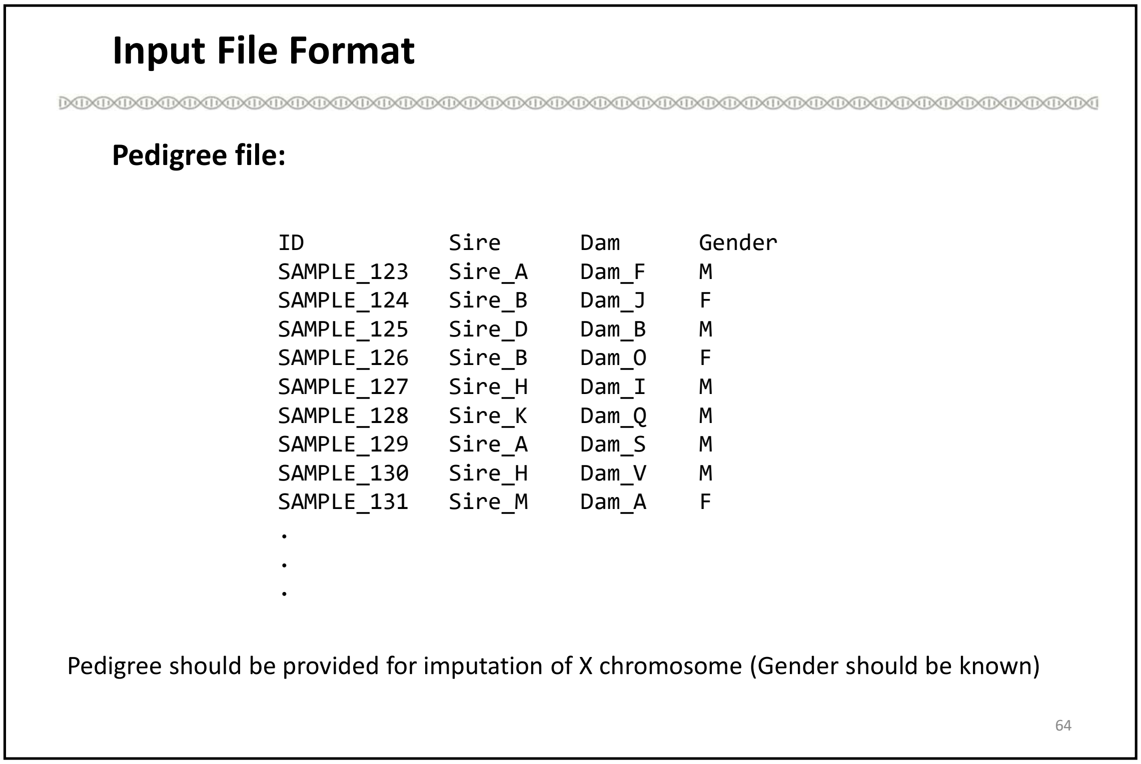
Input File Format

genotype file (expanded format):

```
ID          Chip  genotypes
SAMPLE_123    1   00210215022102011...
SAMPLE_124    1   01201012212201111...
SAMPLE_125    1   11101202201220110...
SAMPLE_126    1   22102110021102101...
SAMPLE_127    1   10120050110010200...
SAMPLE_128    1   02222201052101111...
SAMPLE_129    1   11202210021102122...
SAMPLE_130    2   50550505215555515...
SAMPLE_131    2   52551505025555500...
.
.
.
```

- Simpler format
- Requires more space
- Very convenient for masking in validation study

These 5's could be any genotype code. They are not read!



Input File Format

Pedigree file:

```
ID          Sire     Dam     Gender
SAMPLE_123  Sire_A   Dam_F   M
SAMPLE_124  Sire_B   Dam_J   F
SAMPLE_125  Sire_D   Dam_B   M
SAMPLE_126  Sire_B   Dam_O   F
SAMPLE_127  Sire_H   Dam_I   M
SAMPLE_128  Sire_K   Dam_Q   M
SAMPLE_129  Sire_A   Dam_S   M
SAMPLE_130  Sire_H   Dam_V   M
SAMPLE_131  Sire_M   Dam_A   F
.
.
.
```

Pedigree should be provided for imputation of X chromosome (Gender should be known)

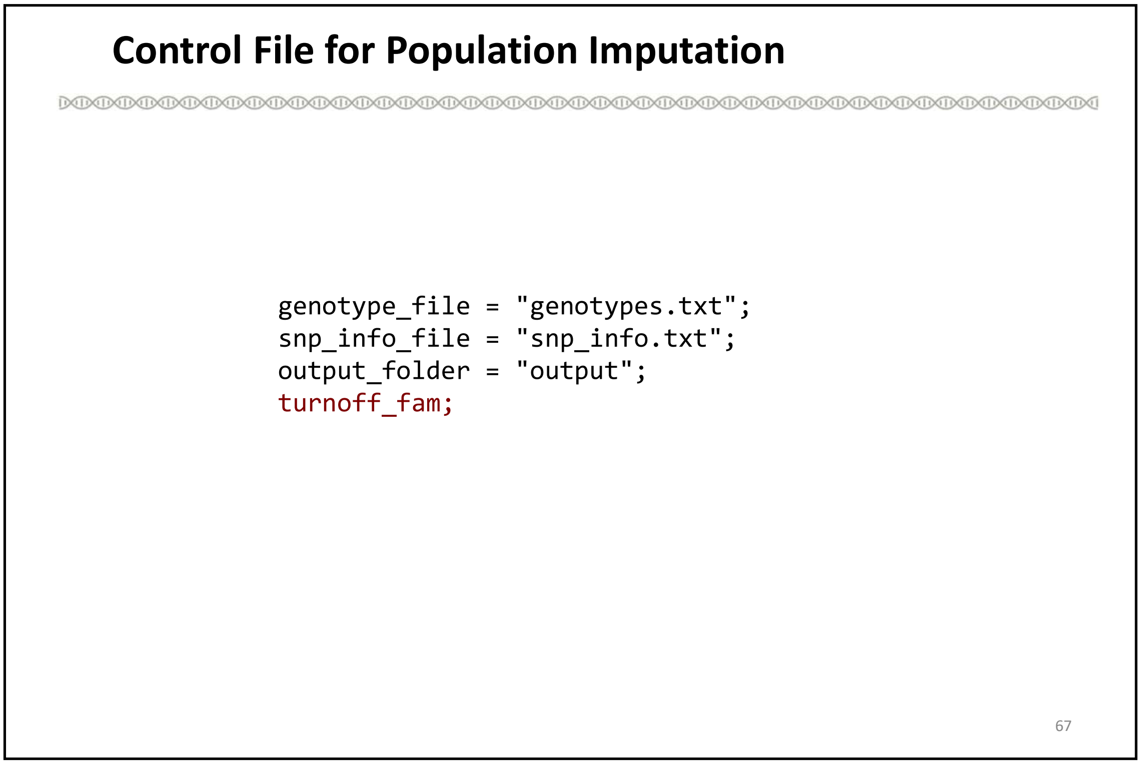
Input File Format

Pedigree file for imputation of X chromosome when pedigree is not available

```
ID            Sire       Dam       Gender
SAMPLE_123    0          0         M
SAMPLE_124    0          0         F
SAMPLE_125    0          0         M
SAMPLE_126    0          0         F
SAMPLE_127    0          0         M
SAMPLE_128    0          0         M
SAMPLE_129    0          0         M
SAMPLE_130    0          0         M
SAMPLE_131    0          0         F
.
.
.
```

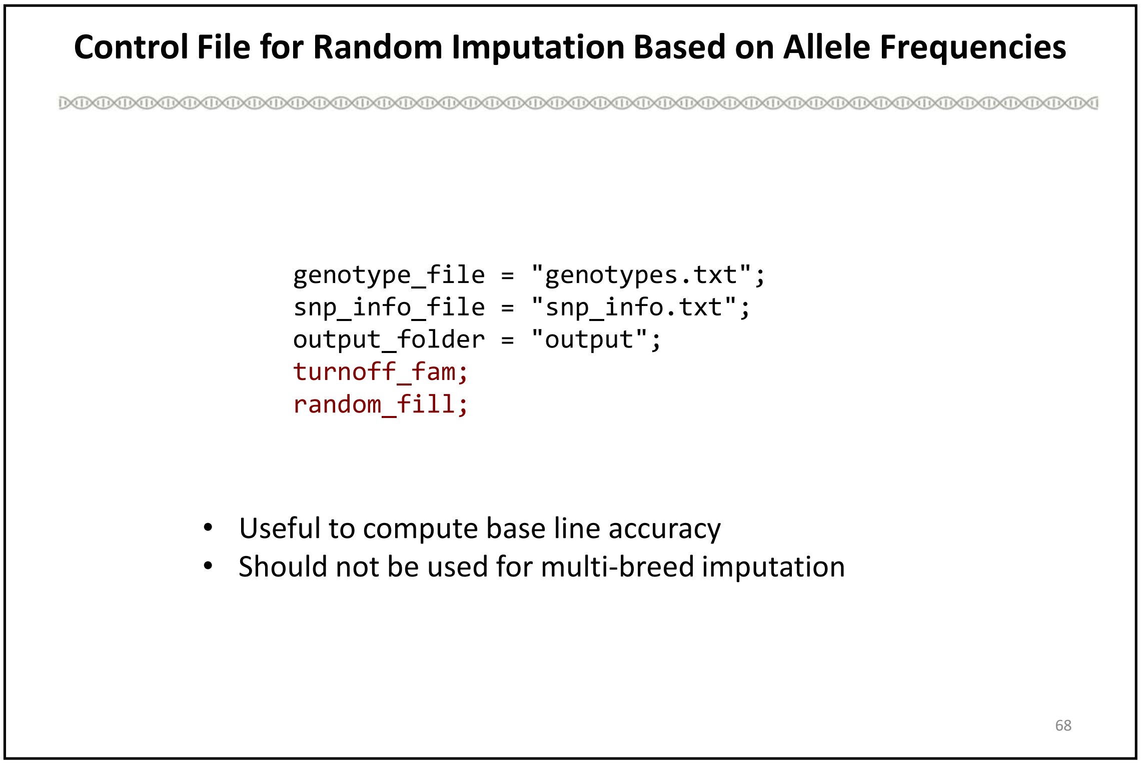
Control File for Family Imputation

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
ped_file = "ped.txt";
output_folder = "output";
turnoff_pop;
```



Control File for Population Imputation

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
output_folder = "output";
turnoff_fam;
```



Control File for Random Imputation Based on Allele Frequencies

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
output_folder = "output";
turnoff_fam;
random_fill;
```

- Useful to compute base line accuracy
- Should not be used for multi-breed imputation

Organizing Input Files

Big data and managing the input files

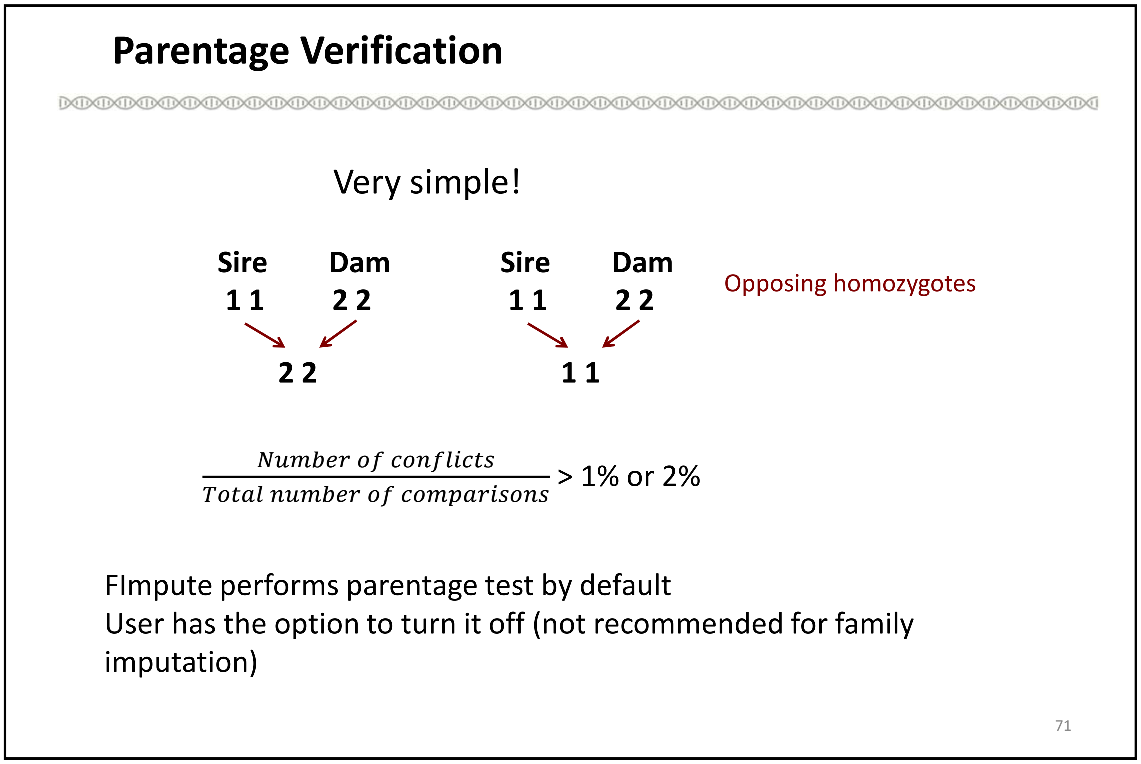
```
genotype_file = "gtype_HD.txt“ "gtype_LD.txt";
snp_info_file = "snp_info.txt";
ped_file = "ped_HD.txt“ "ped_LD.txt";
output_folder = "output";
```

Parentage Verification

- Mendelian inheritance agreement for autosomes & X
- Y-chromosome inheritance (father)
- Mitochondrial inheritance (mother)

FImpute performs parentage test based on opposing homozygotes by default

For dense marker map the accuracy is very high



Parentage Verification

Very simple!

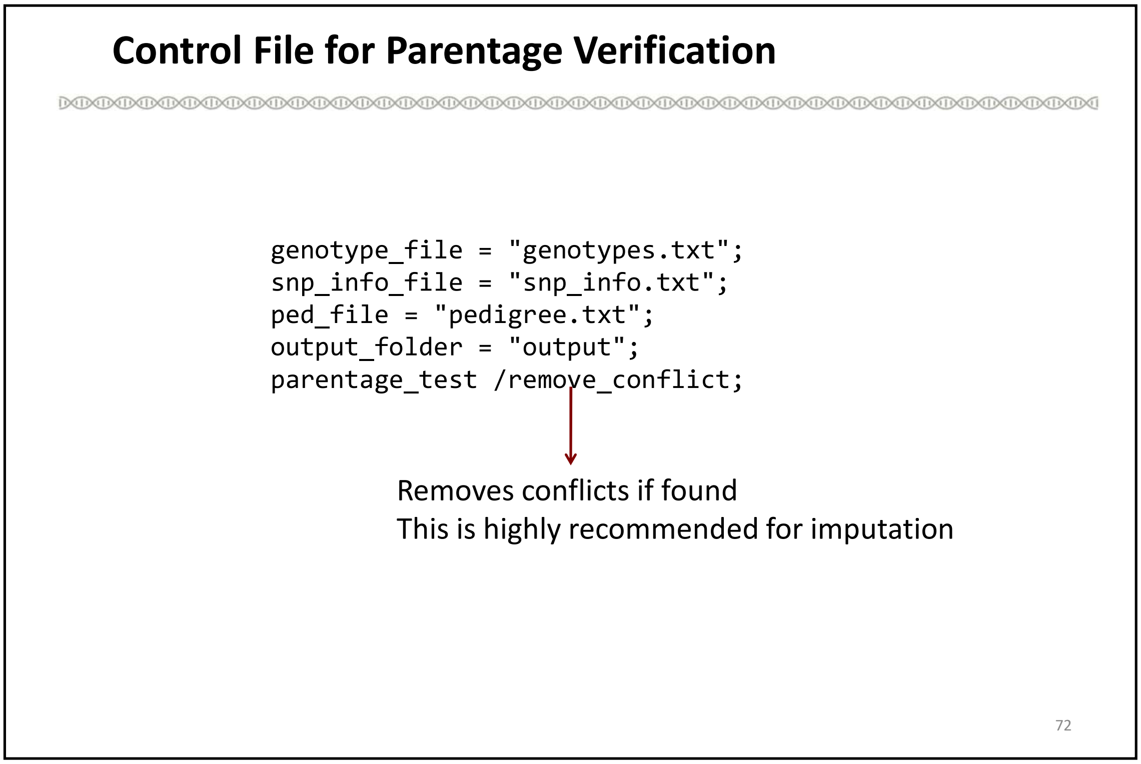
Sire	Dam		Sire	Dam
1 1	2 2		1 1	2 2

→ 2 2 → 1 1

Opposing homozygotes

$$\frac{Number\ of\ conflicts}{Total\ number\ of\ comparisons} > 1\%\ \text{or}\ 2\%$$

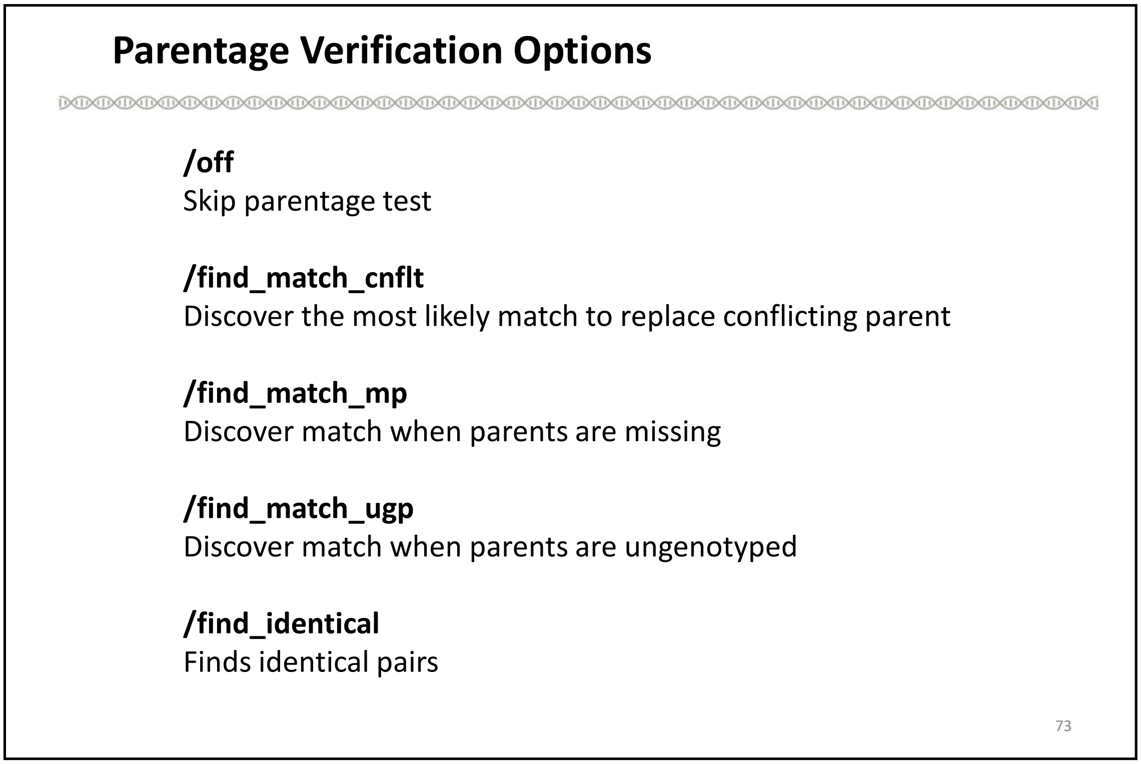
FImpute performs parentage test by default
User has the option to turn it off (not recommended for family imputation)



Control File for Parentage Verification

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
ped_file = "pedigree.txt";
output_folder = "output";
parentage_test /remove_conflict;
```

Removes conflicts if found
This is highly recommended for imputation



Parentage Verification Options

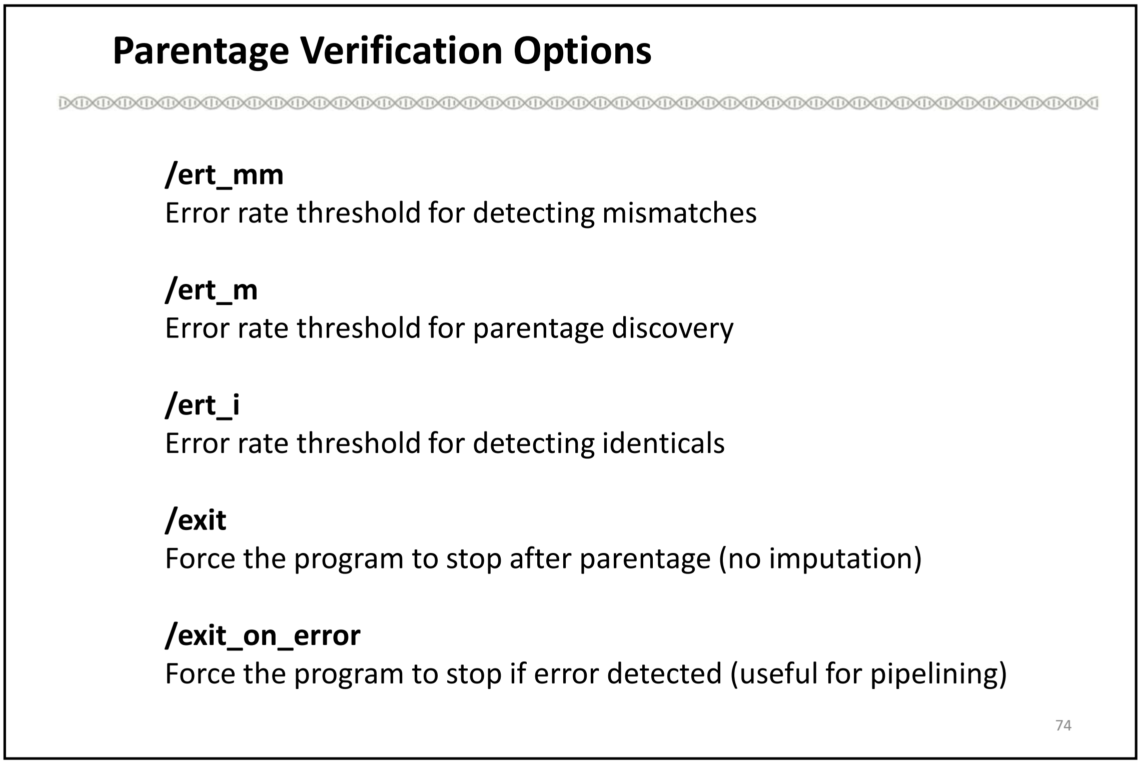
/off
Skip parentage test

/find_match_cnflt
Discover the most likely match to replace conflicting parent

/find_match_mp
Discover match when parents are missing

/find_match_ugp
Discover match when parents are ungenotyped

/find_identical
Finds identical pairs



Parentage Verification Options

/ert_mm
Error rate threshold for detecting mismatches

/ert_m
Error rate threshold for parentage discovery

/ert_i
Error rate threshold for detecting identicals

/exit
Force the program to stop after parentage (no imputation)

/exit_on_error
Force the program to stop if error detected (useful for pipelining)

Parentage Verification Options

/chip = n
Parentage will be based on SNP present on the specified chip

/chip = “file.txt”
Use user-defined list of SNP for parentage

/target = n
Parentage will be carried out for samples on the specified chip

/target = “file.txt”
Parentage will be carried out for user-defined list of samples

/chip and /target options are very useful for big data to speed up computation.

Imputation of Ungenotyped Parents

- Very useful when biological sample is not available (e.g. animals is not alive)
- If ungenotyped animals have enough number of genotype progeny (min 4) their genotypes can be reconstructed (min 8 progeny is recommended)
- Imputation accuracy is a function of the number of genotyped progeny

Imputation of Ungenotyped Parents

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
ped_file = "pedigree.txt";
output_folder = "output";
parentage_test /remove_conflict;
add_ungen /min_fsize=4 /save_sep;
```

/save_sep
Save in a separate file (genotypes_chip0.txt)

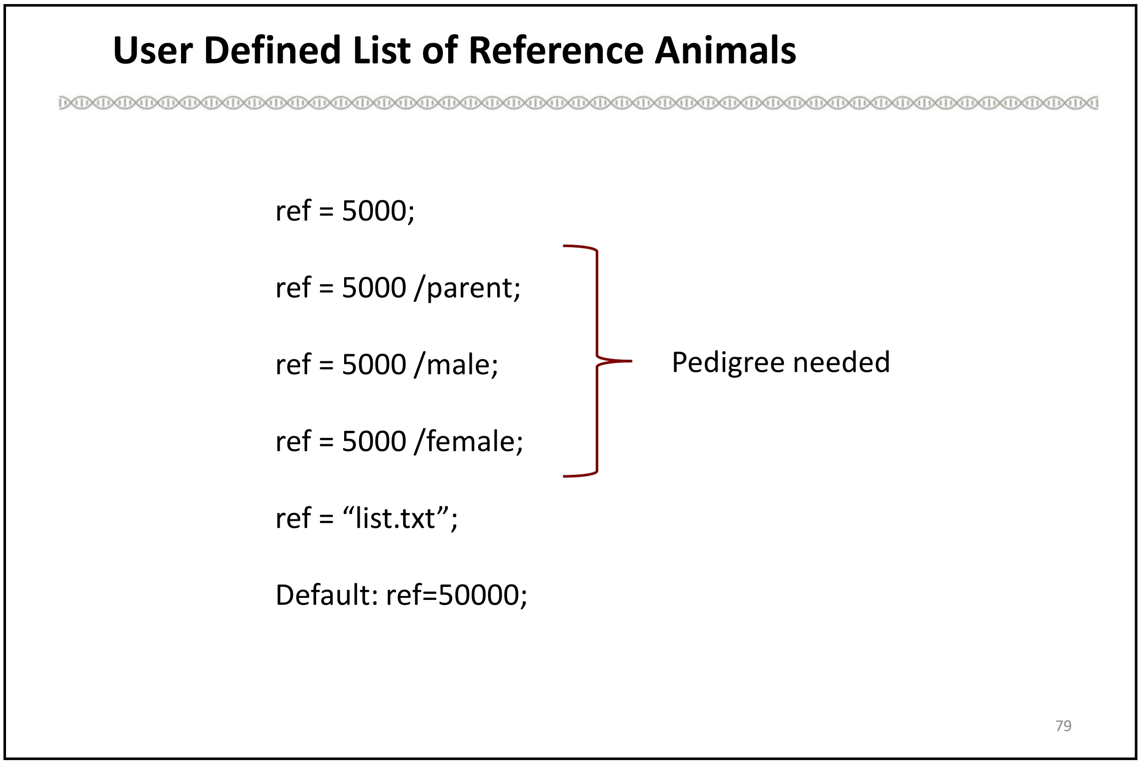
/output_min_fsize
/output_min_call_rate

User Defined List of Target Animals

- Working with large genotype file is challenging
- Avoid copying or duplicating the genotype file
- Avoid repeated editing of the genotype file

target = “list.txt”;

target = c1 c2 c3 …;

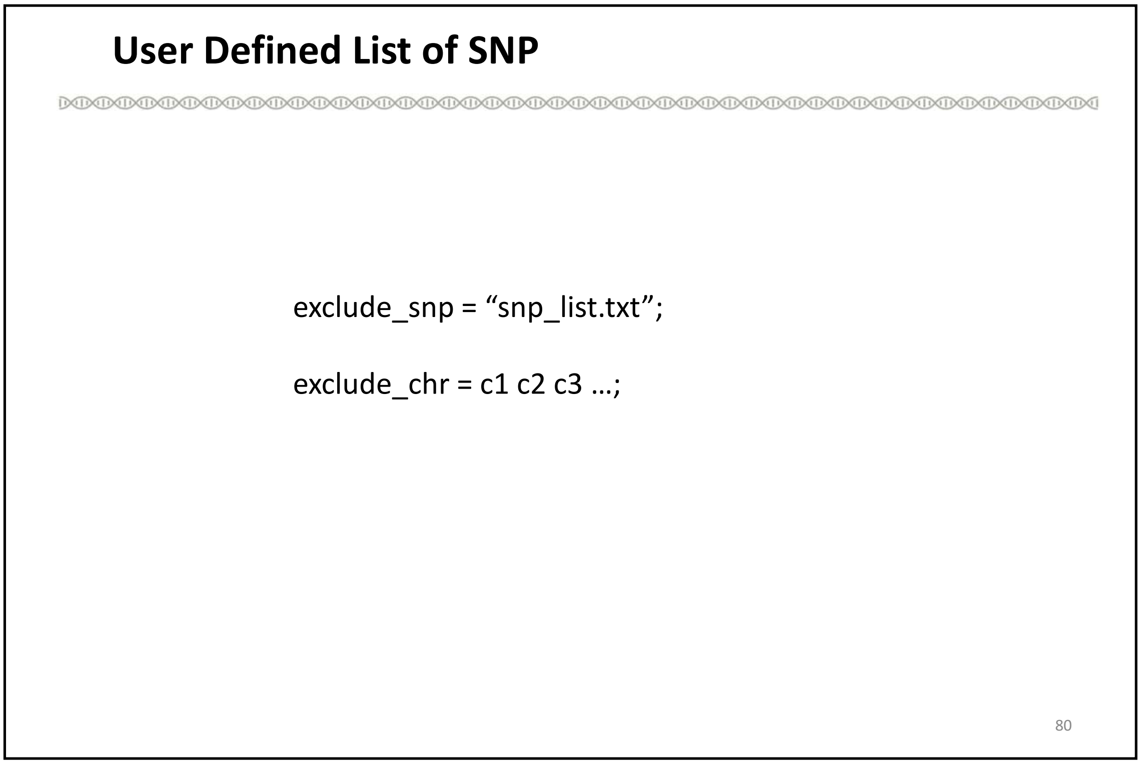


User Defined List of Reference Animals

```
ref = 5000;

ref = 5000 /parent;

ref = 5000 /male;

ref = 5000 /female;

ref = “list.txt”;

Default: ref=50000;
```

Pedigree needed

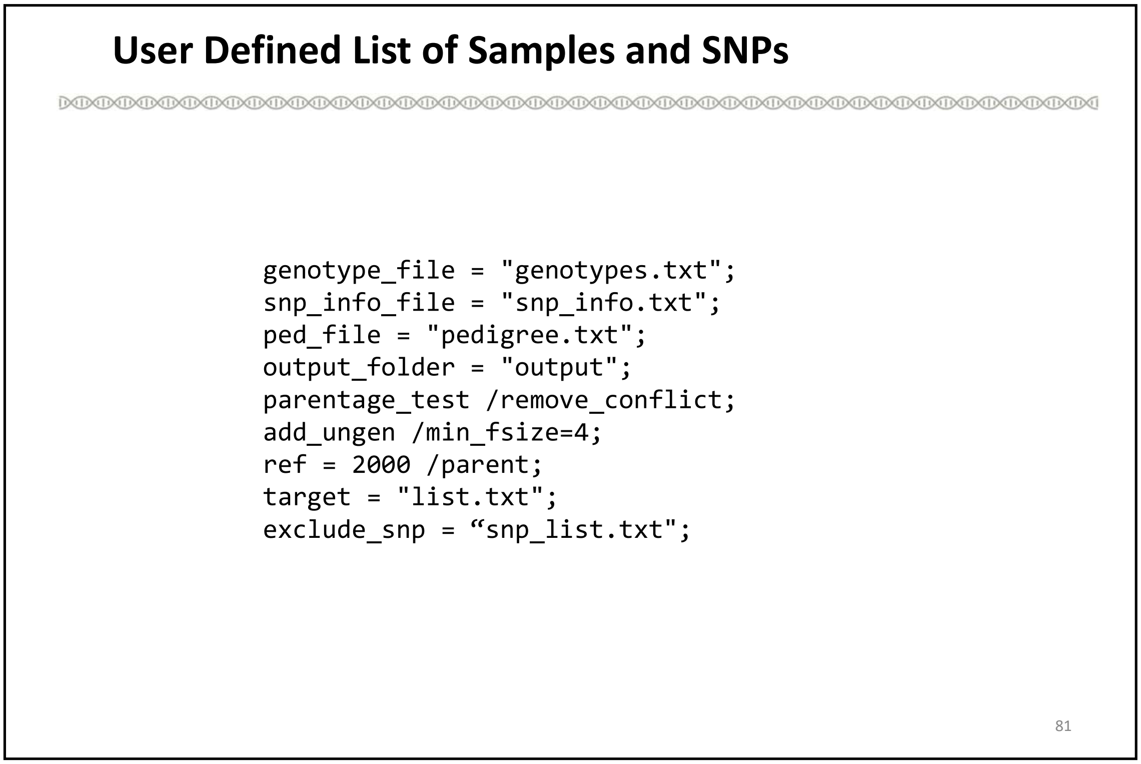
79



User Defined List of SNP

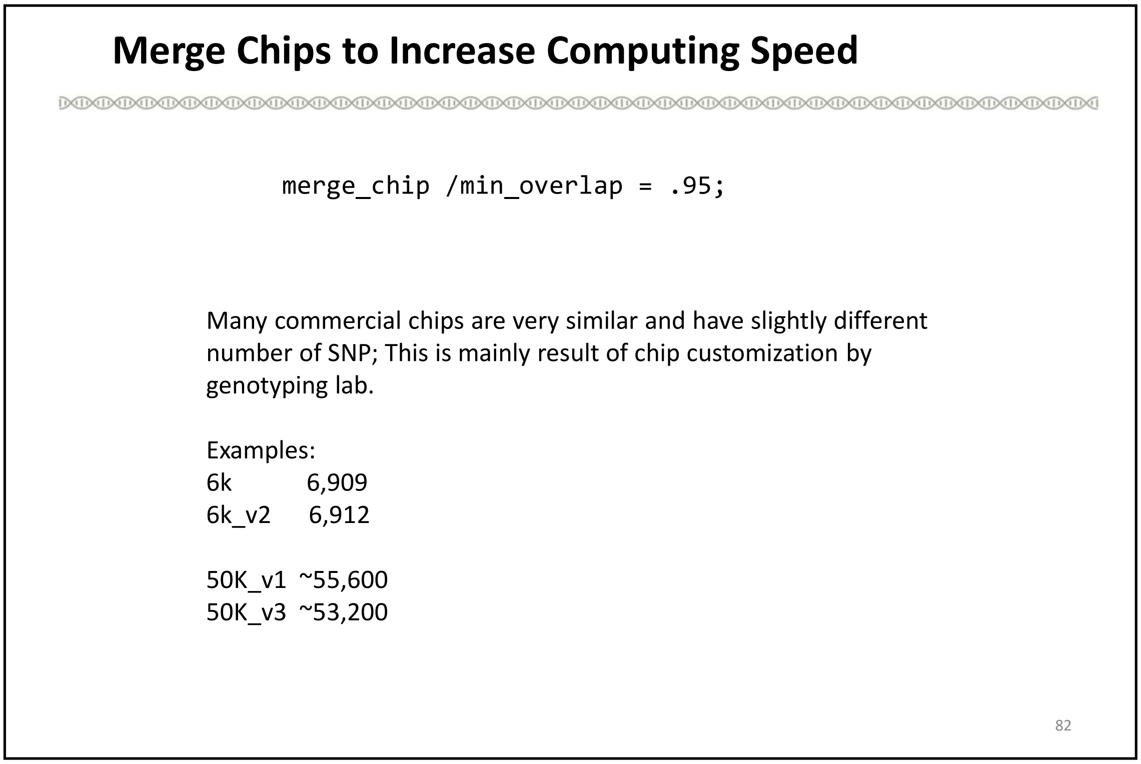
```
exclude_snp = “snp_list.txt”;

exclude_chr = c1 c2 c3 …;
```

80



User Defined List of Samples and SNPs

```
genotype_file = "genotypes.txt";
snp_info_file = "snp_info.txt";
ped_file = "pedigree.txt";
output_folder = "output";
parentage_test /remove_conflict;
add_ungen /min_fsize=4;
ref = 2000 /parent;
target = "list.txt";
exclude_snp = “snp_list.txt";
```



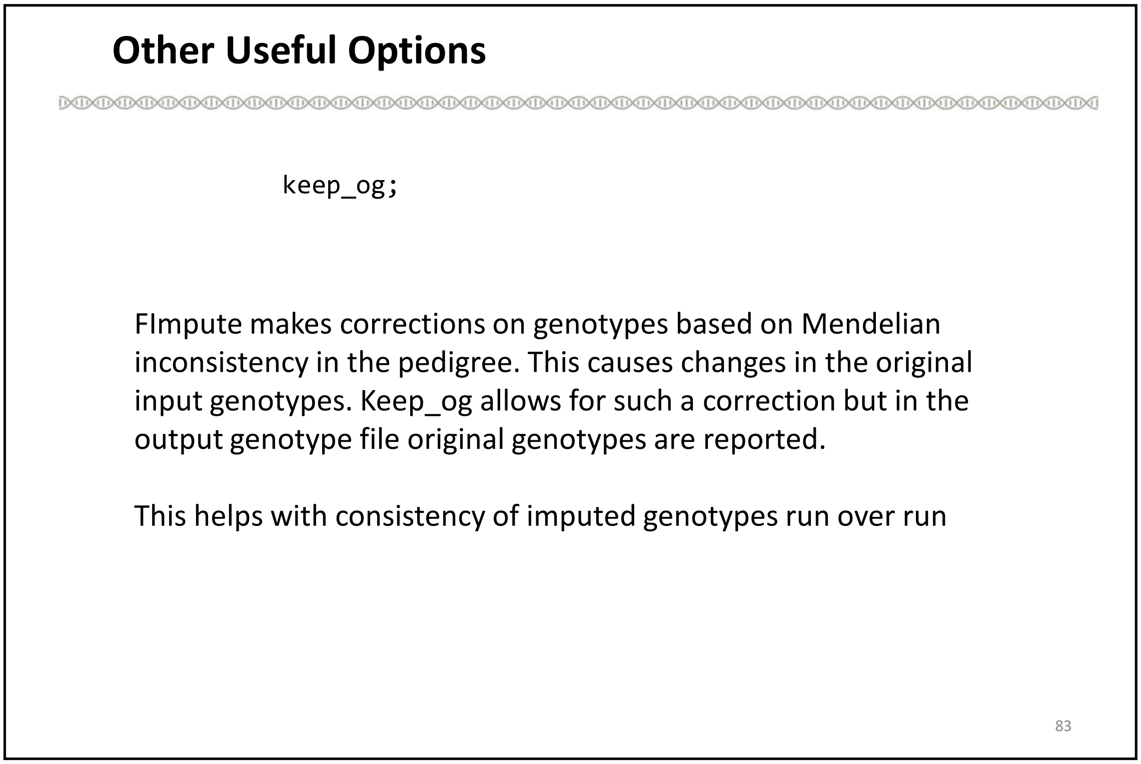
Merge Chips to Increase Computing Speed

```
merge_chip /min_overlap = .95;
```

Many commercial chips are very similar and have slightly different number of SNP; This is mainly result of chip customization by genotyping lab.

Examples:

6k	6,909
6k_v2	6,912
50K_v1	~55,600
50K_v3	~53,200



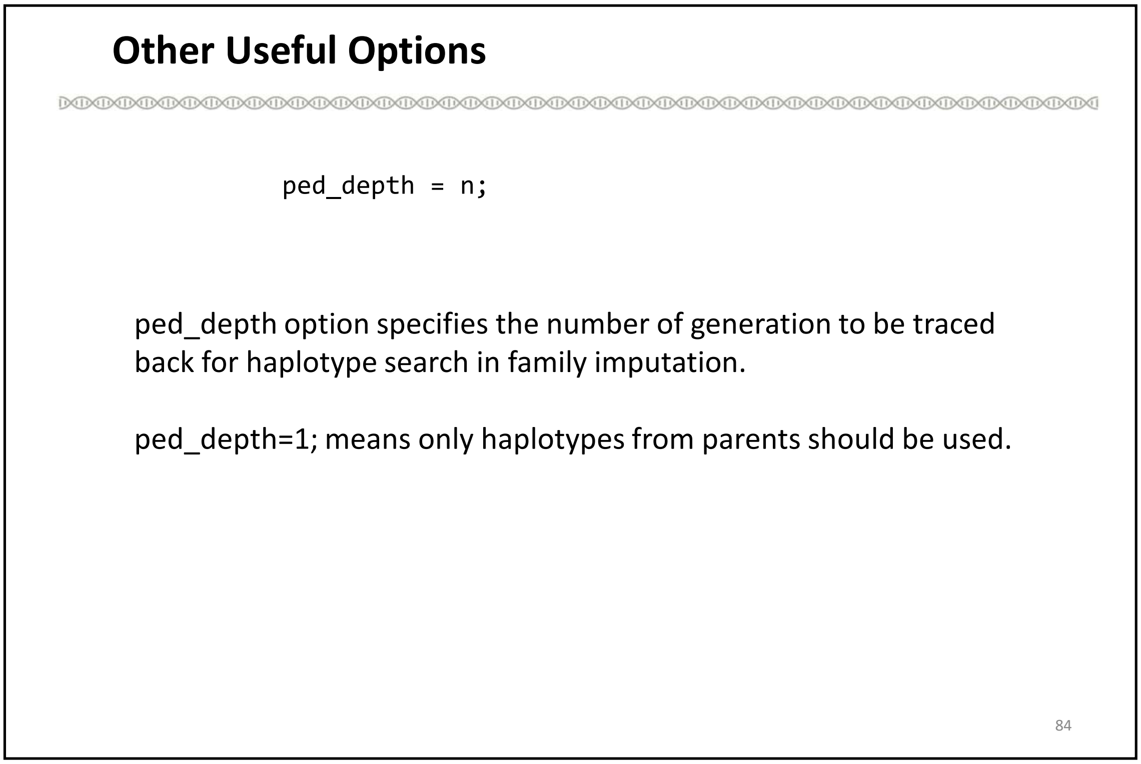
Other Useful Options

```
keep_og;
```

FImpute makes corrections on genotypes based on Mendelian inconsistency in the pedigree. This causes changes in the original input genotypes. Keep_og allows for such a correction but in the output genotype file original genotypes are reported.

This helps with consistency of imputed genotypes run over run

83



Other Useful Options

```
ped_depth = n;
```

ped_depth option specifies the number of generation to be traced back for haplotype search in family imputation.

ped_depth=1; means only haplotypes from parents should be used.

84

Other Useful Options

```
save_chr;
```

Genotypes for each chromosome will be saved separately.

Other Useful Options

```
save_genotype;
```

Haplotypes are the output by default. `save_genotype option converts 3 and 4 codes to 1.`

```
00230240022402043

      ↓

00210210022102011
```

Other Useful Options

```
save_2hap;
```

```
save_2hap option converts genotype/compressed
haplotype code to two haplotype rows per sample.

00230240022402043
          ↓

11211221122212121
11221211122112112
```

The output file is twice as large.

Other Useful Options

```
Ref = “list.txt”;
Ref = n /parent /male /female;
```

List of samples for reference group can be specified by this option with enough flexibility.

The default number of reference size is 50,000, with parents given priority.

Other Useful Options

```
ref_chip = chip;
```

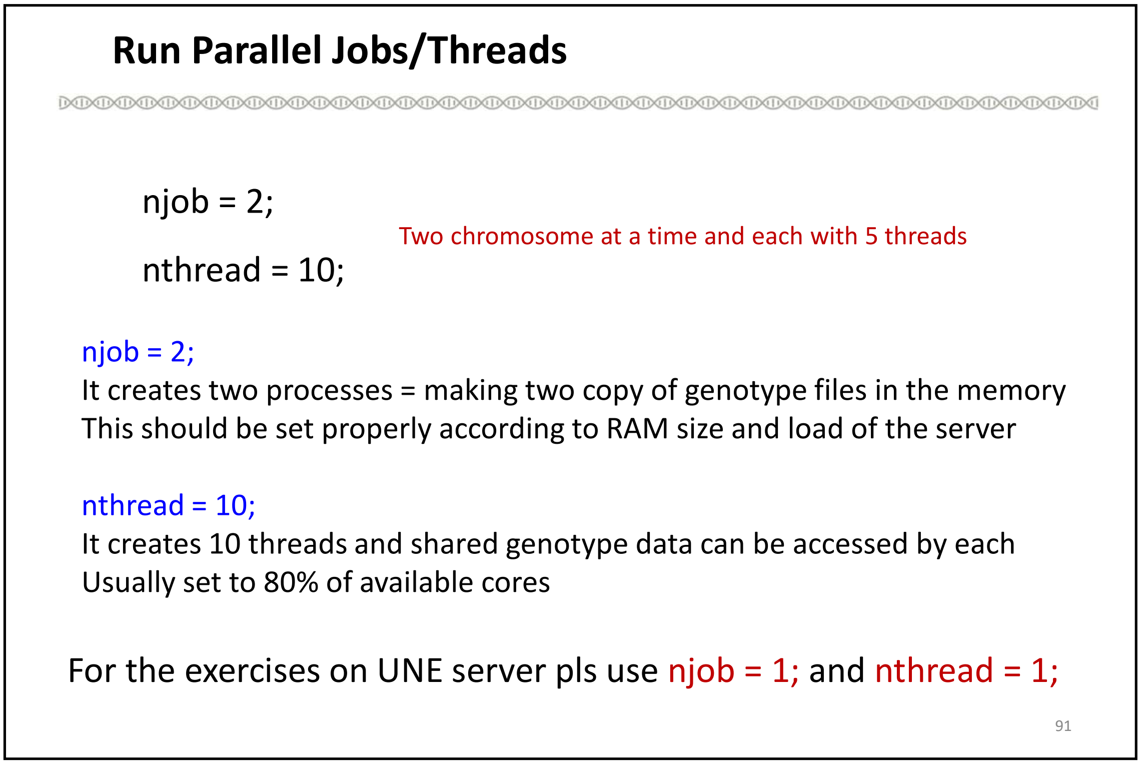
By default the denser panel is the reference panel. ref_chip option allows the user to select reference chip. Note that all SNP not on the reference panel will be removed.

Run Parallel Jobs/Threads

- For multi-core CPU
- Functional on Linux system
- Each chromosome is processed with one core with version 2.2 but version 3 implements a hybrid parallel processing allowing for parallel processing within each chromosome

njob = n1;

nthread = n2;



Run Parallel Jobs/Threads

njob = 2;

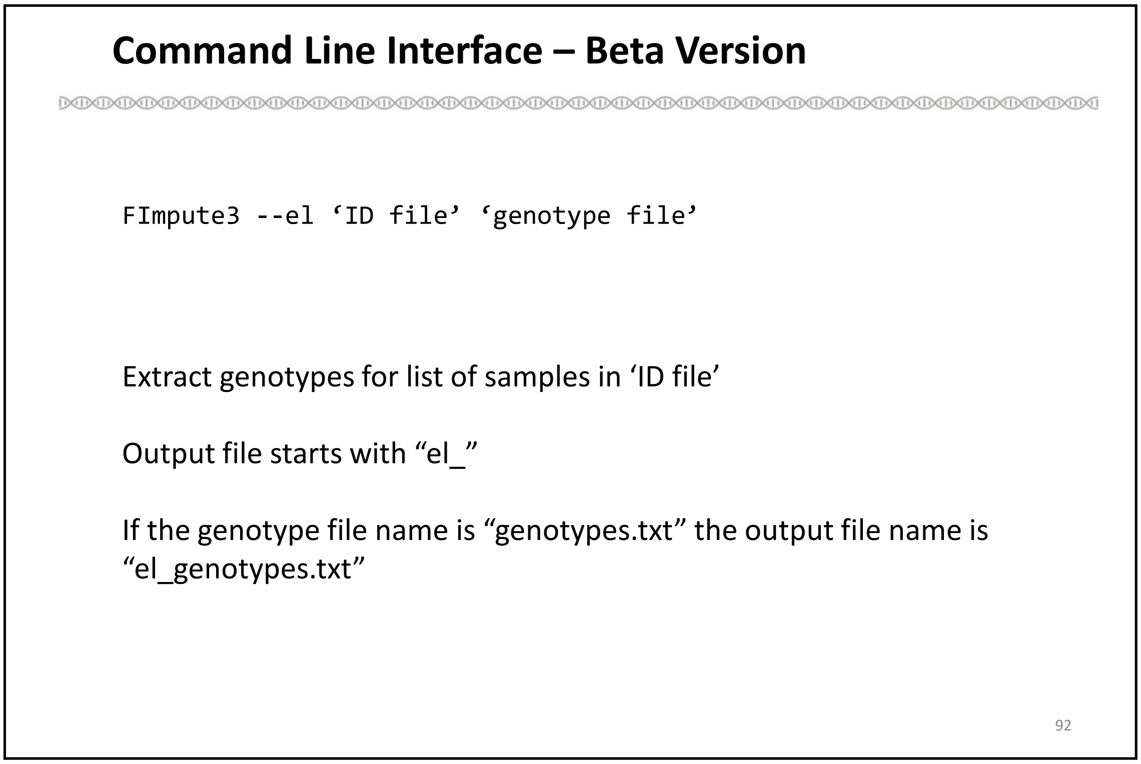
nthread = 10;

Two chromosome at a time and each with 5 threads

njob = 2;
It creates two processes = making two copy of genotype files in the memory
This should be set properly according to RAM size and load of the server

nthread = 10;
It creates 10 threads and shared genotype data can be accessed by each
Usually set to 80% of available cores

For the exercises on UNE server pls use njob = 1; and nthread = 1;



Command Line Interface – Beta Version

```
FImpute3 --el ‘ID file’ ‘genotype file’
```

Extract genotypes for list of samples in ‘ID file’

Output file starts with “el_”

If the genotype file name is “genotypes.txt” the output file name is “el_genotypes.txt”

Command Line Interface – Beta Version

```
FImpute3 --el ‘ID file’ ‘genotype file’ –col 1
```

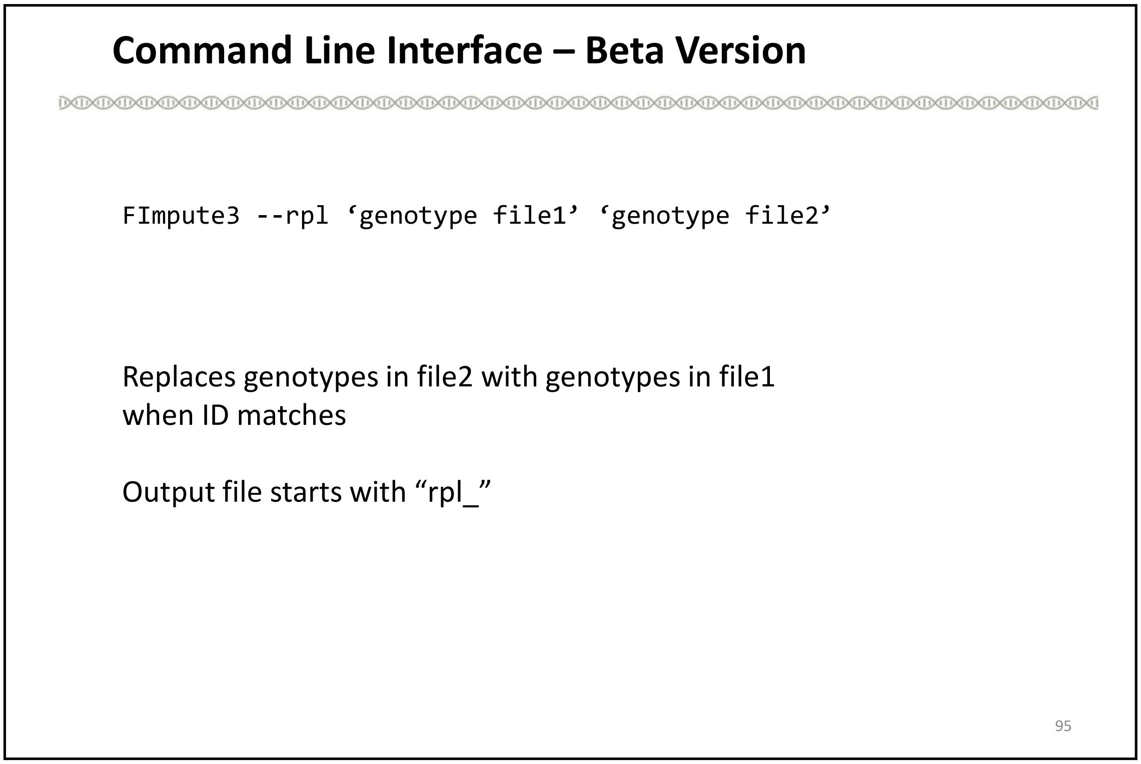
Extract genotypes for list of samples in ‘ID file’
-col option specifies the ID column

Command Line Interface – Beta Version

```
FImpute3 --rm ‘ID file’ ‘genotype file’ –col 1
```

Remove genotypes for list of samples in ‘ID file’

Output file starts with “rm_”

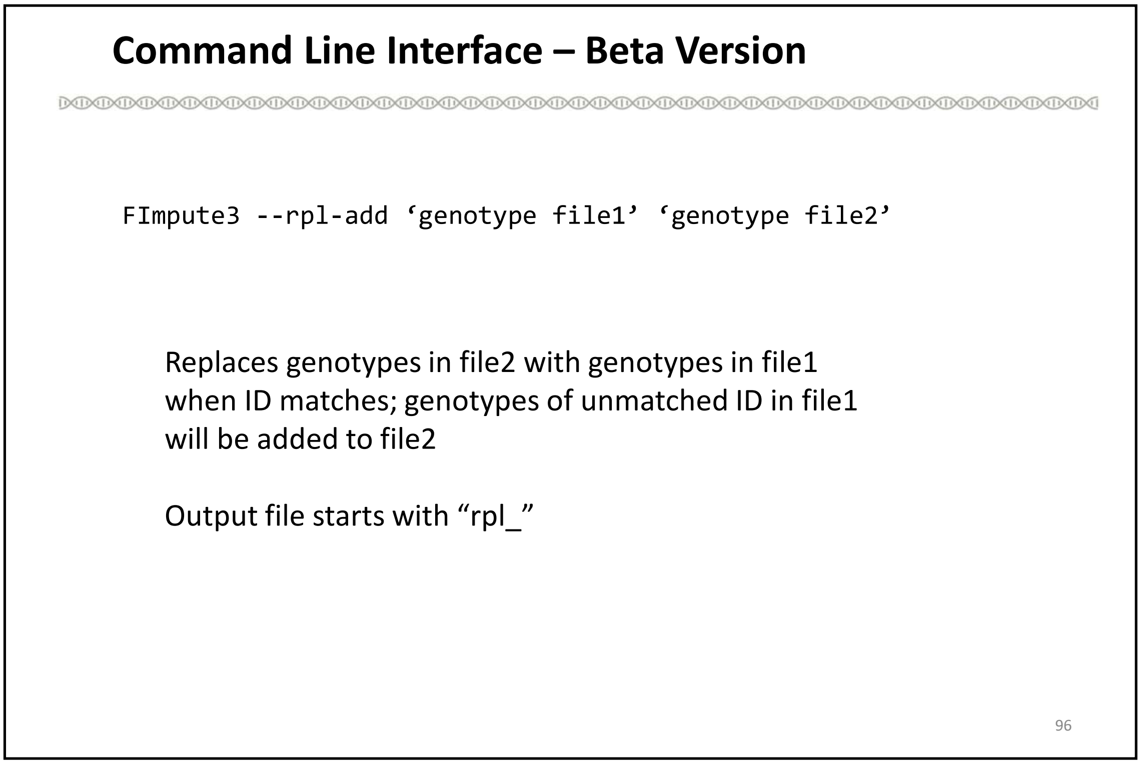


Command Line Interface – Beta Version

```
FImpute3 --rpl ‘genotype file1’ ‘genotype file2’
```

Replaces genotypes in file2 with genotypes in file1
when ID matches

Output file starts with “rpl_”



Command Line Interface – Beta Version

```
FImpute3 --rpl-add ‘genotype file1’ ‘genotype file2’
```

Replaces genotypes in file2 with genotypes in file1
when ID matches; genotypes of unmatched ID in file1
will be added to file2

Output file starts with “rpl_”

Command Line Interface – Beta Version

```
FImpute3 --ec ‘genotype file’ c1
```

Extracts column c1 from genotype file

Output file starts with “ec_”

```
FImpute3 --ec ‘genotype file’ c1 c2 c3 …
```

97

Command Line Interface – Beta Version

```
FImpute3 --fimpute2bed ‘output folder’
```

Converts FImpute format to PLINK bed format

plink.bed
plink.bim
plink.fam

98

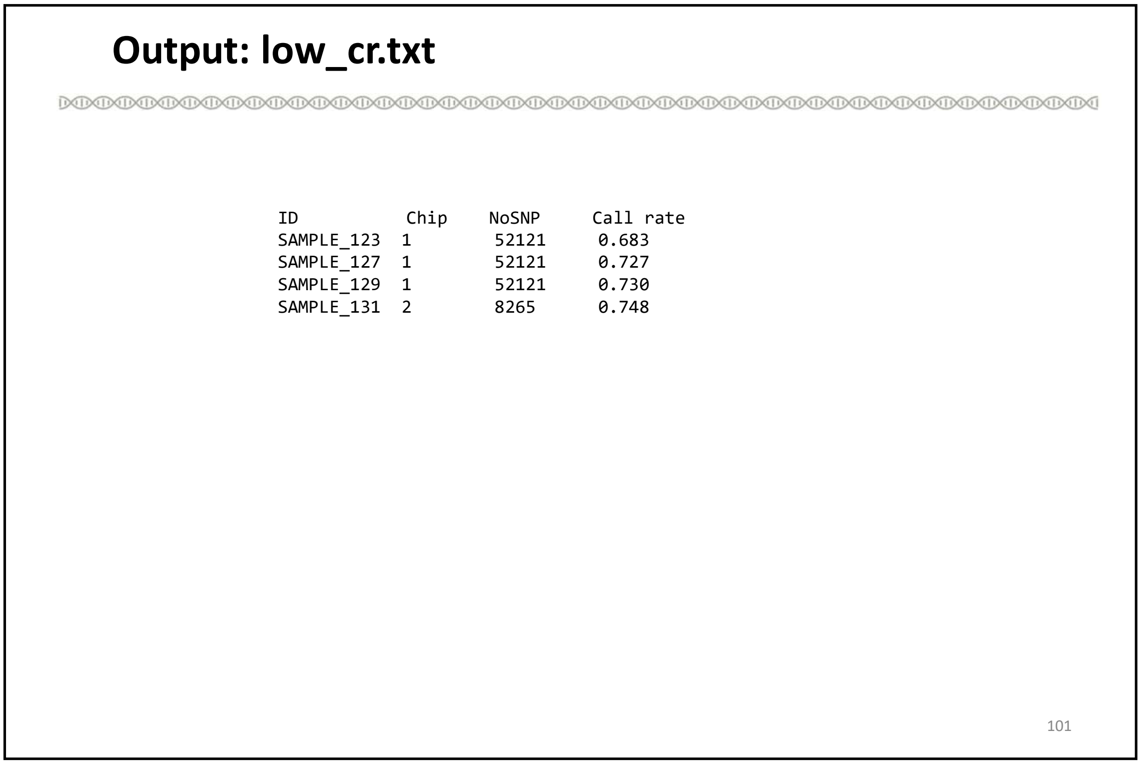
Output Files

```
genotypes_imp.txt
snp_info.txt
report.txt
excluded_snp_list.txt
ref_pop.txt
af_fill_rate.txt
afreq_diff_dist_imp.txt
afreq_diff_dist.txt
distribution.txt
low_cr.txt
org_vs_imp.txt
parentage_test.txt
stat_anim.txt
stat_snp.txt
stat_anim_imp.txt
stat_snp_imp.txt
```

Output: genotypes_imp.txt

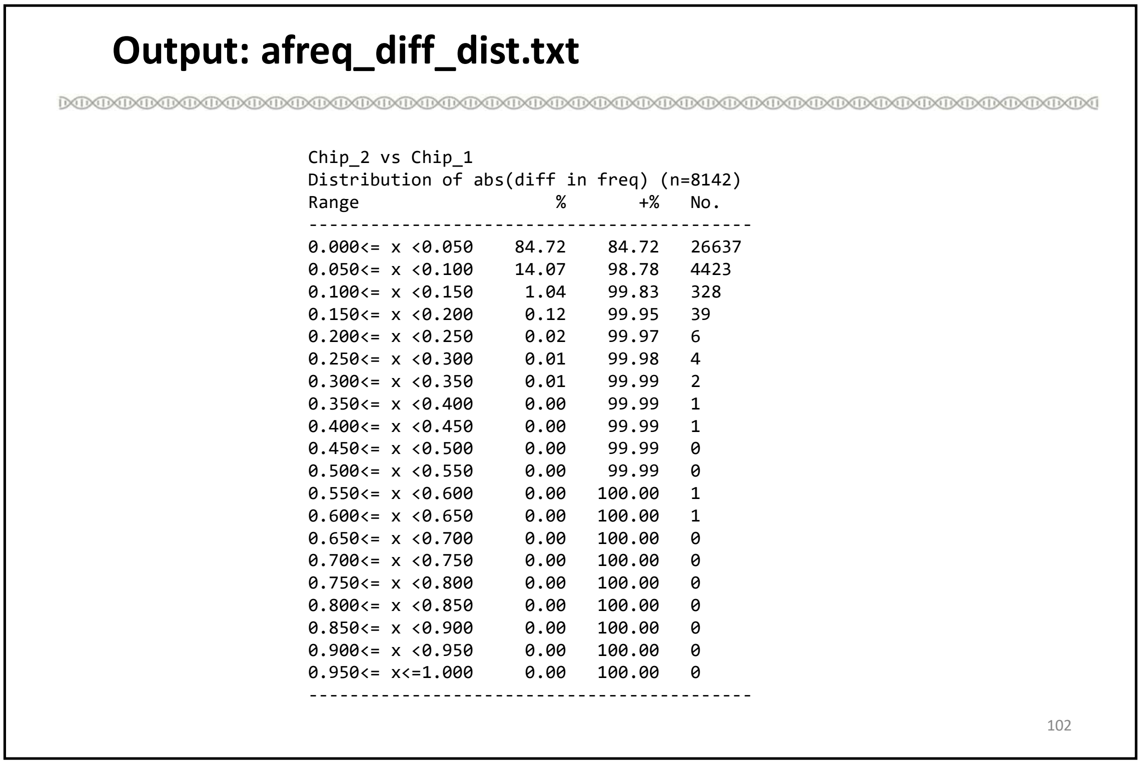
```
ID            Chip    genotypes
SAMPLE_123    1       00230240022402043
SAMPLE_124    2       03201032242203333
SAMPLE_125    1       44403202204220140
SAMPLE_126    1       22402340024402103
SAMPLE_127    2       40420000330030200
SAMPLE_128    1       02222204052304433
SAMPLE_129    1       33202230024402322
SAMPLE_130    1       00024450320044203
SAMPLE_131    2       24402020022030252
.
.
.
```

0 = allele 1, allele 1
1 = allele 1, allele 2 (unphased)
3 = allele 1, allele 2
4 = allele 2, allele 1
2 = allele 2, allele 2
5 = missing



Output: low_cr.txt

```
ID           Chip    NoSNP     Call rate
SAMPLE_123  1        52121     0.683
SAMPLE_127  1        52121     0.727
SAMPLE_129  1        52121     0.730
SAMPLE_131  2        8265      0.748
```



Output: afreq_diff_dist.txt

```
Chip_2 vs Chip_1
Distribution of abs(diff in freq) (n=8142)
Range                    %       +%   No.
-------------------------------------------
0.000<= x <0.050    84.72    84.72   26637
0.050<= x <0.100    14.07    98.78   4423
0.100<= x <0.150     1.04    99.83   328
0.150<= x <0.200     0.12    99.95   39
0.200<= x <0.250     0.02    99.97   6
0.250<= x <0.300     0.01    99.98   4
0.300<= x <0.350     0.01    99.99   2
0.350<= x <0.400     0.00    99.99   1
0.400<= x <0.450     0.00    99.99   1
0.450<= x <0.500     0.00    99.99   0
0.500<= x <0.550     0.00    99.99   0
0.550<= x <0.600     0.00   100.00   1
0.600<= x <0.650     0.00   100.00   1
0.650<= x <0.700     0.00   100.00   0
0.700<= x <0.750     0.00   100.00   0
0.750<= x <0.800     0.00   100.00   0
0.800<= x <0.850     0.00   100.00   0
0.850<= x <0.900     0.00   100.00   0
0.900<= x <0.950     0.00   100.00   0
0.950<= x<=1.000     0.00   100.00   0
-------------------------------------------
```

Output: parentage_test.txt

```
-------------------------- Parentage Test --------------------------
Error rate threshold for mismatch   : 0.02
Error rate threshold for match      : 0.01
A: individual call rate
B: Sire call rate
C: Dam call rate
D: No. Mendelian inconsistencies
E: No. loci compared

ID          Sire     Dam      Check   A      B      C      D       E       Possible match       D       E
SAMPLE_123  Sire_A   Dam_F    Sire    0.871  0.792  0.827  1574    52088   Sire_C               8       52101
SAMPLE_125  Sire_D   Dam_D    Dam     0.836  0.842  0.832  1218    52031   Dam_H                23      52044

--------------------------- Sex conflits ---------------------------
Number of SNP on sex chromosome = 636
Error rate threshold_sex = 0.05

ID                   No. heterozygous loci     Total loci compared
SAMPLE_125           45                        134
```

Output: stat_anim_imp.txt

```
* missing loci ignored.
ID                          Chip  Call0*   Call1*   Call2*   Call5    Homo*    Missing_allele
SAMPLE_123                   1  0.375268 0.324972 0.299760 0.000000 0.675028       0.000000
SAMPLE_124                   1  0.377942 0.317735 0.304323 0.000000 0.682265       0.000000
SAMPLE_125                   1  0.372424 0.331678 0.295898 0.000000 0.668322       0.000000
SAMPLE_126                   1  0.375289 0.330023 0.294688 0.000000 0.669977       0.000000
SAMPLE_127                   1  0.379597 0.322616 0.297787 0.000000 0.677384       0.000000
SAMPLE_128                   1  0.370493 0.331487 0.298020 0.000000 0.668513       0.000000
SAMPLE_129                   1  0.367013 0.340443 0.292545 0.000000 0.659557       0.000000
SAMPLE_130                   1  0.371851 0.326818 0.301331 0.000000 0.673182       0.000000
.
.
.
```