## CONTENTS

1 AlphaSeqOpt  
   1.1 Introduction ........................................... 5  
   1.2 Program history ......................................... 5  
   1.3 List of contributors to development and testing .......... 5  
   1.4 Funding .................................................. 5  
   1.5 Availability ............................................. 5  
   1.6 Conditions of use ....................................... 6  
   1.7 Citation .................................................. 6  
   1.8 Disclaimer ............................................... 6  
   1.9 Advertisement .......................................... 6  

2 Description of the method .................................. 7  
   2.1 GENERAL DESCRIPTION .................................... 7  
   2.2 METHOD 1 .................................................. 7  
   2.3 ALGORITHM 1 ............................................... 7  
   2.4 ALGORITHM 2 ............................................... 8  
   2.5 METHOD 2 .................................................. 9  
   2.6 ALGORITHM 1 ............................................... 9  
   2.7 ALGORITHM 2 ............................................. 10  

3 USING ALPHASEQOPT .......................................... 11  
   3.1 INPUT FILES ............................................... 11  
   3.2 OUTPUT FILES ALPHASEQOPT METHOD 1 .................... 21  
   3.3 OUTPUT FILES ALPHASEQOPT METHOD 2 .................... 26  

4 EXAMPLES ..................................................... 29
· SPECIFICATIONS FILE
  · AlphaSeqOptSpec.txt (mandatory)

· INPUT FILES ALPHASEQOPT METHOD 1
  · INPUT FILES ALGORITHM 1
    · Phased genotype/haplotype/sequence data (compulsory if haplotype IDs files not used)
    · Haplotype ID data (compulsory if phased genotype/haplotype/sequence files not used)
  · INPUT FILES ALGORITHM 2
    · Matrix of sequencing scenarios file (compulsory)
    · Matrix of metrics file (compulsory)
    · Prior sequenced individuals file (optional)
    · Prior sequencing costs file (optional)
    · Top individuals file (compulsory)

· INPUT FILES BOTH ALGORITHMS
  · Pedigree.txt (compulsory)
  · Individuals not to sequence file (optional)

· INPUT FILES ALPHASEQOPT METHOD 2
  · Haplotype ID data (compulsory if phased genotype/haplotype/sequence files not used)

· SEED FILE

* OUTPUT FILES ALPHASEQOPT METHOD 1
  · OUTPUT FILES ALGORITHM 1
    · TopIndividualsToSequence.txt
    · TopIndividualsToSequenceTmatrix.txt
    · FreqOfHaplotypesIndividualsCarry.txt
    · NumberOfHaplotypesPerCore.txt
    · SummaryPedigree.txt
    · CoreIndex.txt
    · HaplotypesIndividualsCarryFullChromosomeX.txt
    · HaplotypesIndividualsCarrySummaryChromosomeX.txt
    · ImputedHaplotypesIndividualsCarryChromosomeX.txt
    · HaplotypesIndividualsCarryPerCoreChromX.txt
    · HaplotypesIndividualsCarryPerCore.txt

  · OUTPUT FILES ALGORITHM 2
    · OutputIteration.txt
    · OutputCoverageCurrent.txt
    · OutputCoverageFinal.txt
· OutputFamilyScenarios.txt
· OutputFamilyAccuracies.txt
· OutputFamilyPropHaplotypes.txt
· OutputFamilyRealisedAccuracies.txt

* OUTPUT FILES ALPHASEQOPT METHOD 2

· InitialInd.txt and FinalInd.txt
· InitialHapSeqFreq.txt and FinalHapSeqFreq.txt
· AlphaSeqOpt_log1_SelectInitialSet.txt
· AlphaSeqOpt_log2_RefineSet.txt
· PopHapFreq.txt

– EXAMPLES
1.1 Introduction

AlphaSeqOpt is a software package for determining the optimum distribution of sequencing resources within a population of interest. AlphaSeqOpt consists of two methods.

The first method determines which individuals should be prioritised for sequencing and the total investment to be made in sequencing this individual and its family members to enable phasing of its haplotypes. The aim of the method is to capture, sequence and phase as many of the population haplotypes as possible for maximum sequence imputation ability into the rest of the population. The second method determines which individuals should be prioritised for sequencing and the total investment to be made in sequencing these individuals so that the maximum number of haplotypes in the population are sequenced at a target coverage that is high enough to derive highly accurate consensus haplotypes used for whole-population imputation of sequence data.

1.2 Program history

The first algorithm in AlphaSeqOpt was designed and developed by Serap Gonen (SG) in 2015-16 from a prototype algorithm developed and designed by John Hickey (JH) in 2015. The second algorithm was designed and developed by Roger Ros-Freixedes (RRF) in 2015-16 from a prototype algorithm developed and designed by JH in 2015. The latest manual was updated by SG and RRF.

1.3 List of contributors to development and testing

Serap Gonen, Roger Ros-Freixedes, Gregor Gorjanc, John M Hickey.

1.4 Funding

The development of AlphaSeqOpt has been largely not funded by specific grants, instead the work has been undertaken as part of other projects which have been funded by Genus PLC, Aviagen, Illumina, Medical Research Council, and ISPG funding from the BBSRC to The Roslin Institute.

1.5 Availability

AlphaSeqOpt is available from: http://www.alphagenes.roslin.ed.ac.uk/
Material available includes the compiled programs for 64 bit Linux, Mac OSX, and Windows machines, together with a User Manual.

Please report bugs or suggestions on how the program / user interface / manual could be improved or made more user friendly to John.Hickey@roslin.ed.ac.uk.

1.6 Conditions of use

AlphaSeqOpt is available to the scientific community free of charge. Users are required, however, to credit its use in any publications. Commercial users should contact John Hickey (John.Hickey@roslin.ed.ac.uk).

1.7 Citation


1.8 Disclaimer

While every effort has been made to ensure that AlphaSeqOpt does what it claims to do, there is absolutely no guarantee that the results provided are correct. Use of AlphaSeqOpt is entirely at your own risk!

1.9 Advertisement

AlphaSeqOpt is part of the AlphaSuite collection of software programs that we have developed. The AlphaSuite collection can perform many of the common tasks in animal breeding, plant breeding, and human genetics including genomic prediction, breeding value estimation, variance component estimation, GWAS, imputation, phasing, optimal contributions, simulation, field trial designs, and various data recoding and handling tools.

The AlphaSuite is available at this link: http://www.alphagenes.roslin.ed.ac.uk/software-packages/

---

1 People often ask as to the origins on the AlphaSuite naming convention. The original program in the AlphaSuite was AlphaBayes, which was named after the “Bayesian Alphabet” that it implemented. The name was suggested by Brian Kinghorn. Subsequent programs evolved via mutation with a conserved primer!
2.1 GENERAL DESCRIPTION

AlphaSeqOpt is an exact method for optimal allocation of limited sequencing resources in livestock populations with existing phased genomic data to maximise the ability to phase and impute sequenced haplotypes into the whole population.

AlphaSeqOpt is a software package designed to direct breeders in the decision of individuals to prioritise for sequencing in a population of interest and the coverage at which to sequence individuals in order to maximise the use of the collected sequence data for sequence imputation into the rest of the (unsequenced) population.

AlphaSeqOpt implements two methods: (1) select a group of individuals that collectively represent the maximum possible portion of the haplotype diversity in the population and optimally allocate a fixed sequencing budget amongst the families of selected individuals to enable phasing of the haplotypes carried by the selected individuals at the sequence level. The aim is that the final haplotypes, when phased at the sequence level, represent the maximum possible portion of the haplotype diversity in the population that can be sequenced and phased at a fixed sequencing budget. (2) select a group of individuals that collectively carry the maximum possible proportion of population haplotypes at a frequency such that a fixed sequencing budget is allocated optimally to provide a minimum target coverage for those haplotypes. The aim is that the maximum possible number of haplotypes reaches an accumulated sequencing coverage that is high enough to derive accurate consensus haplotypes used for accurate phasing and imputation of whole-population sequence data.

A brief description of both methods is given below. Further details can be found in the associated manuscripts.

2.2 METHOD 1

We have split the description of method 1 into two parts: (1) algorithm to select the group of individuals to sequence, and (2) algorithm to optimally allocate a fixed sequencing budget amongst the families of selected individuals.

2.3 ALGORITHM 1

2.3.1 AIM

At each iteration, algorithm 1 determines the best individual whose genome is most representative of the most frequent haplotypes in the population. The idea of the algorithm is to enable sequence data to be collected on as many different haplotypes as possible, prioritising the most frequent haplotypes in the population. The assumption of the algorithm is that choosing individuals whose genomes carry many of the most frequent haplotypes in the population would enable more complete sequence imputation into other individuals that have genotype but may not necessarily have sequence data.
2.3.2 INPUT DATA

Existing phased, true or imputed genotype, haplotype or sequence data

2.3.3 STEPS

1. For each chromosome, determine a set of m haplotypes of length n SNPs
2. Determine the haplotypes of each individual and construct a population haplotype library. Calculate the frequency of each haplotype in the population
3. Select the individual that carries the most high frequency haplotypes as a candidate for sequencing (a ‘focal individual’)
4. Mask the haplotypes carried by this focal individual in the rest of the population, assuming that sequencing and phasing its haplotypes at the sequence level would enable the imputation of these haplotypes into individuals that share these haplotypes (i.e. these haplotypes would be accounted for)
5. Repeat steps 1 – 4 to generate a list of the k focal individuals for sequencing, each time ensuring that the most frequent haplotypes that are not already sequenced are prioritised. Since k is user-defined, this process can be repeated until all haplotypes in the population would be sequenced

2.4 ALGORITHM 2

2.4.1 AIM

The optimisation of the allocation of a fixed sequencing budget in algorithm 2 is addressed using an evolutionary algorithm that recursively samples and evaluates different combinations of sequencing scenarios within and across the focal families. We define a sequencing scenario as the selected sequencing coverages for each member of a focal family. Expected phasing accuracies for a sequencing scenario may be calculated using algorithms such as that implemented in AlphaFamSeq. Algorithm 2 can be run for any number of rounds until convergence is reached or until no further improvements are made.

2.4.2 INPUT DATA

1. The k focal individuals and the proportion of times their haplotypes appear in the rest of the population
2. The accuracies of phasing each member of a focal family given a chosen sequencing scenario.
3. Pedigree relationships in the whole population
4. The cost of preparing and sequencing a DNA library at any coverage
5. The total fixed sequencing budget

2.4.3 STEPS

1. Determine the network of relationships between the k focal families
2. For each member of a focal family, sample a random sequencing coverage from a user-defined list of possible sequencing coverages and determine the sequencing scenario and haplotype phasing accuracy
3. Calculate the overall cost of the selected set of sequencing scenarios across all focal families. The overall cost takes into account pre-existing DNA libraries and/or sequence data for some individuals
4. Compute a “goodness criterion” for this set of sequencing scenarios and associated cost. The criterion takes into account:

(a) The proportion of population haplotypes that would be phased at the sequence level (as defined by algorithm 1)

(b) The accuracy of phasing the haplotypes carried by the focal individual given sampled sequencing scenarios (AlphaFamSeq)

(c) The fixed sequencing budget. If the total cost is above the budget then this set of sequencing scenarios is penalised

5. Repeat steps 2-4 n times (n is the number of rounds for convergence) and select the optimum set of sequencing scenarios as the one that maximises the goodness criterion

AlphaSeqOpt is flexible in that the user may choose to run the two algorithms independently or directly one after the other.

More details about the methods used by AlphaSeqOpt are provided in section AlphaSeqOptSpec.txt below, which describes the parameters file.

2.5 METHOD 2

We have split the description of method 1 into two parts: (1) algorithm to select an initial set of individuals to sequence, and (2) algorithm to refine the sequencing set.

2.6 ALGORITHM 1

2.6.1 AIM

At each iteration, algorithm 1 selects the individual that carries more haplotypes with a sequencing coverage closer to the target sequencing coverage. The idea of the algorithm is to distribute sequencing resources evenly across haplotypes to reduce the number of haplotypes that are under- or over-sequenced.

2.6.2 INPUT DATA

1. Existing phased, true or imputed genotype, haplotype or sequence data

2. The cost of preparing and sequencing a DNA library at any coverage

3. The total fixed sequencing budget

2.6.3 STEPS

1. For each chromosome, determine a set of m haplotypes of length n SNPs

2. Determine the haplotypes of each individual and construct a population haplotype library. Calculate the frequency of each haplotype in the population and in the sequencing set.

3. Calculate a score for each haplotype, using a score function based on the frequency of the haplotype in the sequencing set relative to the target coverage

4. Calculate a individual score as the sum of haplotype scores
5. Add the animal with the maximum score to the sequencing set. If more than one individual has the maximum score choose one randomly. Repetition of animals is allowed and the number of times that an individual is added to the set indicates the sequencing coverage for that individual.

6. Repeat steps 3 – 5 to generate a list of individuals in the sequencing set, masking haplotypes that reach the target sequencing coverage. Stop when costs equal the sequencing resources available.

2.7 ALGORITHM 2

2.7.1 AIM

The optimisation of the allocation of a fixed sequencing budget in algorithm 2 is achieved through rounds of exchanges of the individuals in the sequencing set.

2.7.2 INPUT DATA

The input data for algorithm 2 is generated internally by algorithm 1. Algorithm 2 cannot be run independently from algorithm 1.

1. Existing phased, true or imputed genotype, haplotype or sequence data
2. The sequencing set produced in Algorithm 1
3. The cost of preparing and sequencing a DNA library at any coverage
4. The total fixed sequencing budget

2.7.3 STEPS

1. Choose a predetermined number of slots of the set and remove them from the set
2. Repeat steps 3 - 5 of Algorithm 1 until the emptied slots are assigned to new individuals. If the fixed budget is consumed before all the slots are filled, some slots can be left empty; and vice versa, if the fixed budget is not consumed when all the slots are filled, extra slots can be added to sequence extra individuals.
3. If the exchanges result in the same or a greater percentage of haplotypes sequenced at (or above) the target coverage, keep the new set. Otherwise, discard the new set in favour of the previous set.
4. Repeat steps 1 - 3 for a predefined number of exchange rounds

More details about the methods used by AlphaSeqOpt are provided in section AlphaSeqOptSpec.txt below, which describes the parameters file.
3.1 INPUT FILES

The main input file for AlphaSeqOpt is the AlphaSeqOptSpec.txt file. Depending on whether Algorithm 1 and 2 are being run independently or together, different input files may be required and not all options need be set in the AlphaSeqOptSpec.txt file.

3.1.1 SPECIFICATIONS FILE

AlphaSeqOptSpec.txt (mandatory)

An example of the AlphaSeqOptSpec.txt file is shown in Figure 1. Text to the left of the comma should not be changed. The program is controlled by changing the input right of the first comma. Each parameter is described in detail below:

```
#--------------------------
###ALPHASEQOPTSPEC####
#--------------------------
Method ,1
### ALPHASEQOPT1 PARAMETERS ###
OptimisationMethod ,Sequence
NumberOfChromosomes ,2
NumberOfSnps ,Constant,1000
IndividualPhasedThreshold ,0.9
SnphasedThreshold ,0.9
HaplotypeThresholdPhasing ,0.97
HaplotypeThresholdComparing ,0.97
HaplotypeThresholdExcluding ,0
HaplotypeMismatchAllowed ,3
HaplotypePrinting ,Summary
NumberOfIndividualsToSequence ,5000
CoreLength ,100
ByPassSharedHaploDeinition ,Yes
PriorNumberOfCores ,10,10
SequencedIndividualFile ,seq.txt
IncludeSequencedIndivContrib ,Yes
IndividualsNotToSequenceFile ,none
MatrixOfScenariosFile ,Scenarios.txt
MatrixOfMetricsFile ,Metrics.txt
TopIndividualsFile ,TopIndividualsToSequence.txt
NumberOfFamiliesToSequence ,50
PriorSeqCostsFile ,none
OverallTotalCost ,100000
```
**Number of Coverage**: 6
**Allowed Coverage**: 0, 1, 2, 5, 10, 20
**EvolAlgProbOrMaxCoding**: 1
**EvolAlgPopSize**: 100
**EvolAlgOptimisationMethod**: EvolutionaryAlgorithm
**EvolAlgOptimisationRounds**: 10
**EvolAlgOptimisationRoundsConv**: 10
**EvolAlgOptimisationRoundsPrint**: 1
**EvolAlgRecombinationType**: Regions
**EvolAlgRecombination**: 0.05, 2.00
**EvolAlgWeightFactor**: 0.05, 0.50
**EvolAlgRoundsSwapGAorDE**: 1
**CostOfLibraryPerIndividual**: 40
**CostOfLibraryOneXSequencing**: 80
**Metric**: Accuracy

### ALPHASEQOPT2 PARAMETERS ###
**HapIndexFile**: filename.txt
**NumberOfIndividuals**: 5000
**NumberOfCores**: 10
**Budget**: 100000
**CostPerLibrary**: 40
**CostPerX**: 80
**HaplotypeSequenceThreshold**: 10
**MaxIndCoverage**: 5
**NumberOfExchangeRounds**: 1000
**NumberOfChangesPerRound**: 25
**AlreadySequencedFile**: No
**ExcludedFile**: No
**FilterRareHaplotypes**: All
**PopulationFrequencyToFilter**: 2
**ContextCountFlankingHaps**: 3
**ContextCountCombinationFlanks**: 2
**SecondaryHapSeqThresh**: 2
**PopFreqForSecondThresh**: 5

**Figure 1**: Example of AlphaSeqOptSpec.txt

**PARAMETER DESCRIPTIONS**

**Method**
This parameter determines the method that the user wishes to run.

Method 1 runs if this parameter is set to 1.
Method 2 runs if this parameter is set to 2.

**PARAMETERS FOR ALPHASEQOPT METHOD 1**

**Optimisation Method**
This parameter determines the algorithm that the user wishes to run.

Algorithm 1 runs if this parameter is set as **Sequence**
Algorithm 2 runs if this parameter is set as **Coverage**
Algorithms 1 and 2 run concurrently if this parameter is set as **Sequenceandcoverage**
An additional option to select top individuals to sequence using pedigree relationships (as given in Goddard et al. 2011, JABG (T Matrix calculation)) has also been coded. To use this option, this parameter should be set to Pedigree

**NumberOfChromosomes** (algorithm 1)

Tells the program how many chromosomes are being analysed.

This option is only required if **OptimisationMethod** is set as Sequence or Sequenceandcoverage

If set then this option assumes that either phased genotypes or haplotype files are provided for each chromosome individually. See parameter **ByPassSharedHaploDeinition** for more description on the genotype or haplotype files required to run algorithm 1.

**NumberOfSnps** (algorithm 1)

This parameter includes a character string specifying if the number of SNP per chromosome as constant or variable, followed by one or more numbers indicating the number of SNPs for each chromosome in order, each of them being separated by a comma. The character string is Constant or Variable. If the number of SNPs per chromosome is constant, the term Constant must be followed by only one value for the number of SNPs. If the number of SNPs per chromosome is Variable, the term Variable must be followed by n values separated by commas, each one representing the number of SNPs for each chromosome in order. This parameter is required only if phased genotypes are provided as input.

**IndividualPhasedThreshold** (algorithm 1)

This parameter is a proportion that ranges from 0.0-1.0. This parameter is a threshold that determines whether an individual’s genotype information is included in the construction of the population haplotype library in algorithm 1. Only individuals with full genotype data exceeding this threshold are considered in the construction of the population haplotype library. This parameter is required only if phased genotypes are provided as input.

**SnpPhasedThreshold** (algorithm 1) This parameter is a proportion that ranges from 0.0-1.0. This parameter is a threshold that determines whether any given SNP is included in the construction of the population haplotype library in algorithm 1. Only SNPs with more than this threshold number of individuals genotyped are considered in the construction of the population haplotype library. If a SNP is below this threshold then population haplotypes are defined and constructed without including the genotype information of this SNP. This parameter is required only if phased genotypes are provided as input.

**HaplotypeThresholdPhasing** (algorithm 1)

This parameter is a proportion that ranges from 0.0-1.0. This parameter determines whether a haplotype present in the haplotype library should be considered as a haplotype for sequencing. If it is below this threshold then this haplotype is considered incomplete, thus there is not enough information to determine its frequency in the population. This parameter is required only if phased genotypes are provided as input.

**HaplotypeThresholdComparing** and **HaplotypeMismatchAllowed** (algorithm 1)

These two parameters jointly determine whether two individuals share a haplotype. **HaplotypeThresholdComparing** is a proportion that ranges from 0.0-1.0 and **HaplotypeMismatchAllowed** is an integer. These parameters should be set considering each other. For example, if a haplotype is defined as 100 SNP (see parameter **CoreLength** below for setting haplotype length), setting **HaplotypeThresholdComparing** to 0.97 and **HaplotypeMismatchAllowed** to 3 ensures agreement between these two parameters. This parameter is required only if phased genotypes are provided as input.

**HaplotypeThresholdExcluding** (algorithm 1)

This parameter determines the frequency at which a haplotype should be present in the population in order to include it for sequencing. If set to zero, all haplotypes are considered.

**HaplotypePrinting** (algorithm 1)

This parameter determines the output information that the user requires. **HaplotypePrinting** should be set to All to print the whole haplotype of individual or to Summary to print out haplotypes constructed from SNPs that exceed the **SnpPhasedThreshold** parameter. This parameter is required only if phased genotypes are provided as input.
**NumberOfIndividualsToSequence** (algorithm 1)

This parameter is the number of focal individuals to select for sequencing.

**CoreLength** (algorithm 1)

This parameter sets the number of SNPs to define a haplotype length on a chromosome.

**ByPassSharedHaplotypeDefinition** (algorithm 1)

If this parameter is set to No then algorithm 1 requires phased genotype files to first determine the haplotypes carried by each individual before building a population library, defining shared haplotypes and identifying the individuals to sequence. Phased genotype files should be placed in a subdirectory of the working directory. The subdirectory should be named “FinalHaplotypes” and should contain phased genotype files for each chromosome.

Files for each chromosome should be labelled as “ChromosomeX.txt”, where X is the chromosome number from 1 to *NumberOfChromosomes*. Missing values should be coded as ‘9’.

If this parameter is set to Yes then algorithm 1 expects a matrix of haplotype IDs within each core for each individual, which it will use to construct a population haplotype library, determine haplotype sharing between individuals and identify the individuals for sequencing. The matrix file must be supplied separately for each chromosome in a subdirectory of the working directory named “Cores”. Files for each chromosome should be labelled as “ChromosomeX.txt”, where X is the chromosome number from 1 to *NumberOfChromosomes*. These files should have one line per individual, with two columns per core specifying the paternal and maternal haplotypes (the algorithm is insensitive to whether the paternal or maternal haplotype is supplied first).

Missing values should be coded as ‘-99’.

**PriorNumberOfCores** (algorithm 1)

This parameter should be set if **ByPassSharedHaplotypeDefinition** is Yes. In this case, the user should supply a comma-separated vector for the number of cores on each chromosome.

**SequencedIndividualFile** (algorithm 1)

This parameter may be utilised to supply the program with the path to a file containing list of individuals that already have sequence data and that should not be considered for sequencing. For these individuals, the haplotypes that they carry will be masked in the rest of the population so that other high frequency haplotypes with no sequence data are prioritised for sequencing. If not used, this parameter should be set as “None”.

**IncludeSequencedIndivContrib** (algorithm 1)

If set to Yes, this parameter prints the proportion of population haplotypes carried by individuals that already have sequence data. If set to No then the haplotypes of these individuals are still accounted for but their proportions are set to zero.

**IndividualsNotToSequenceFile** (algorithm 1 & 2)

This parameter may be utilised to supply the program with the path to a file containing list of individuals that have no existing sequence data but should not be selected for sequencing (for example, because tissue/DNA samples are not available for these individuals). In this case, the algorithm will exclude these individuals from being sequenced by setting their haplotype sharing contributions to zero. Their haplotypes may be sequenced in other individuals. If not used, this parameter should be set as “None”.

**MatrixOfScenariosFile** (algorithm 2)

This parameter may be utilised to supply the program with the path to a file containing a matrix of different family sequencing scenarios. A sequencing scenario is defined as the selected sequencing coverages for a focal individual, its parents and its grandparents. The file should have seven columns of integers representing coverages with no header. The seven columns should be coverages for the family members in the following order: Column 1 - Focal individual; Column 2 - Sire; Column 3 - Dam; Column 4 - Paternal Grand sire; Column 5 - Paternal Granddam; Column 6 - Maternal Grand sire; Column 7 - Maternal Granddam.
The file should have all combinations of sequencing scenarios for the different coverages specified in the AllowedCoverage parameter (see below).

**MatrixOfMetricsFile** (algorithm 2)

This parameter may be utilised to supply the program with the path to a file containing a matrix of expected haplotype phasing accuracies for each individual in a focal family given the family sequencing scenario. Therefore, each line in this file must correspond directly to the scenarios in MatrixOfScenariosFile. The file should have seven columns of numbers between 0-1 representing phasing accuracies given a sequencing scenario with no header. The seven columns should be accuracies for each family member in the following order: Column 1 - Focal individual; Column 2 - Sire; Column 3 - Dam; Column 4 - Paternal Grandsire; Column 5 - Paternal Granddam; Column 6 - Maternal Grandsire; Column 7 - Maternal Granddam.

**TopIndividualsFile** (algorithm 2)

This parameter specifies the location of the file specifying the information for the k focal individuals and their families to sequence. If algorithm 1 was run on the dataset, this will be the TopIndividualsToSequence.txt file which is the output from algorithm 1. Otherwise, users must specify a file with the same seven columns as the TopIndividualsToSequence.txt file (see Output Files section below for further detail (TODO LINK)).

**NumberOfFamiliesToSequence** (algorithm 2)

This parameter sets the total number of focal families across which the fixed sequencing budget will be distributed. Only these individuals will have an assigned coverage.

**PriorSeqCostsFile** (algorithm 2)

This parameter specifies the path to a file containing information on existing coverage and prior sequencing costs incurred for an individual in a focal family. This file has 4 columns as follows:

- Column 1: Individual ID
- Column 2: Prior library cost incurred for this individual (exclude currency signs)
- Column 3: Existing average sequencing coverage for this individual
- Column 4: Prior sequencing cost incurred for this individual (per 1x coverage).

If this parameter is not to be used, it should be set as ‘none’.

**OverallTotalCost** (algorithm 2)

This parameter specifies the maximum fixed sequencing budget for the project.

**NumberOfCoverage** (algorithm 2)

This parameter is a single integer specifying the total number of coverages at which an individual can be sequenced.

**AllowedCoverage** (algorithm 2)

This parameter specifies a vector of coverages at which a given individual can be sequenced. For example, this may be 1,2,5,10,20. In this case NumberOfCoverage should be 5 (see above).

**EvolAlgProbOrMaxCoding** (algorithm 2)

Internally, algorithm 2 sets up a matrix of probabilities for sequencing a given individual at each specified coverage. Each row in the matrix corresponds to each member of a focal family and each column corresponds to a sequencing coverage. For example, with coverages of 1,2,5,10,20, individual number 1 may have probabilities of 0.1, 0.2, 0.4, 0.3, 0.0.

If set to 0, this parameter internally sets Max coding to be True. In this case, the algorithm will always choose to sequence an individual with the highest sampled probability (i.e. 5x in the example above).

If set to 1, this parameter internally sets Prob coding to be True. In this case, the algorithm will sample a coverage given the probabilities. Therefore, the individual may be chosen to be sequenced at 1x as there is 10% chance for that case.
**EvolAlgPopSize** (algorithm 2) This parameter is the number of solutions to work with at each round. We recommend values between 50-100; smaller values will limit the search space of the algorithm and larger values may increase run time.

**EvolAlgOptimisationMethod** (algorithm 2)

This parameter can be set as *EvolutionaryAlgorithm* or *RandomSearch*.

If set as *RandomSearch*, the probabilities of being sequenced at a given coverage are sampled from a discrete uniform distribution.

If set as *EvolutionaryAlgorithm*, the probabilities of being sequenced at a given coverage are sampled from a Gumbel (type-I) distribution.

**EvolAlgOptimisationRounds** (algorithm 2)

This parameter specifies the number of optimisation rounds to run. Higher values may give the algorithm a better chance to evolve and converge but may take longer to run.

**EvolAlgOptimisationRoundsConv** (algorithm 2)

This parameter should be set equal to or less than **EvolAlgOptimisationRounds**, and provides a way of overruling the **EvolAlgOptimisationRounds** parameter. For example, if **EvolAlgOptimisationRounds** is set to 10000 and this parameter is set to 1000, if at any point during the optimisation process 1000 consecutive rounds of no improvement in the algorithm is seen, then the program will print the best solution and terminate.

**EvolAlgOptimisationRoundsPrint** (algorithm 2)

This parameter determines the rounds at which the algorithm prints the progress to screen. For example if set to 10, the solution of each 10th round will be printed to screen. Setting this value to 1 enables the user to track the progress of the algorithm in each round.

**EvolAlgRecombinationType** (algorithm 2)

This parameter can be set to *Regions* or *NonRegions*. This parameter determines how “recombination” between two solutions occurs, where a “recombination” is defined as the swapping of probabilities between solutions.

If this parameter is *Regions*, the positions and number of recombinations (or swaps) between solutions is sampled from a Poisson distribution.

If this parameter is *NonRegions*, a single recombination (swap) position is sampled from a discrete uniform distribution.

**EvolAlgRecombination** (algorithm 2)

This parameter should be a comma separated vector of two values. It provides the minimum and maximum values to define the properties of the uniform distribution to sample the probability of a recombination (or swap) occurring between two solutions in each optimisation round. This parameter is used to set up the matrix of probabilities if **EvolAlgRecombinationType** is set to *Regions*.

**EvolAlgWeightFactor** (algorithm 2)

This parameter should be a comma separated vector of two values. It provides the minimum and maximum values to define the properties of the uniform distribution to sample the probability of a recombination (or swap) occurring between two solutions in each optimisation round. This parameter is used to set up the matrix of probabilities if **EvolAlgRecombinationType** is set to *NonRegions*.

**EvolAlgRoundsSwapGAorDE** (algorithm 2) This parameter is the rounds at which switching between the **EvolAlgOptimisationMethod** i.e. *EvolutionaryAlgorithm* or *RandomSearch* occurs. For example, if set to 10, then a switch between the methods will occur at every 10th round.

**CostOfLibraryPerIndividual** (algorithm 2)

This parameter sets the cost of preparing a DNA library for one individual.
**CostOfLibraryOneXSequencing** (algorithm 2)
This parameter sets the cost of sequencing a DNA library at 1x coverage.

**Metric** (algorithm 2)
This parameter should be set to *Accuracy* if the file provided in the *MatrixOfMetricsFile* parameter are phasing accuracies (e.g. from AlphaFamSeq). This metric will be used to determine how informative the sequencing scenarios will collectively be for haplotype imputation into the rest of the population.

**PARAMETERS FOR ALPHASEQOPT METHOD 2**

**HapIndexFile**
Name of the file with the haplotypes that each individual carries.

The first column is animal ID, followed by two columns per core with the IDs of the two haplotypes that the animal carries at that core.

AlphaSeqOpt method 1 can be used to generate the file “HaplotypesIndividualsCarryPerCore.txt”, which can be used as the input for method 2. To do so, set *OptimisationMethod* to *Sequence* or *Sequenceandcoverage* and set *ByPassHaploDefinition* to *No*. The input file can also be produced externally.

**NumberOfIndividuals**
Tells the program how many individuals are being analysed.

**NumberOfCores**
Tells the program how many cores per individual are being analysed.

**Budget**
This parameter specifies the maximum fixed sequencing budget for the project.

**CostPerLibrary**
This parameter sets the cost of preparing a DNA library for one individual.

**CostPerX**
This parameter sets the cost of sequencing a DNA library at 1x coverage.

**HaplotypeSequenceThreshold**
This parameter specifies the target sequencing coverage of the haplotypes.

**MaxIndCoverage**
This parameter specifies the maximum coverage allowed for each individual. This is a cumulative coverage that includes any pre-existent sequence data if available (see ‘AlreadySequencedFile’ below)

**NumberOfExchangeRounds** (algorithm 2) Number of rounds of exchanges during the refinement step.

The greatest increases in percentage of haplotypes sequenced at the target coverage are often achieved within the first thousands of rounds. There is less benefit in having very long chains of exchanges.

Set to 0 if you want to obtain an initial set without refinement.

Recommended values: 10000-100000

**NumberOfChangesPerRound** (algorithm 2)
Number of slots of the set that are emptied in each iteration. This parameter can be set as in integer or, alternatively, as “WholeSet”.

3.1. INPUT FILES
If this value is too low there is a lower probability of retrieving a better solution using the algorithm. If this value is too high the new solution in each iteration may be too divergent from the initial set, leading to lack of improvement. An intermediate value can produce improvements of the percentage of haplotypes sequenced at the target coverage for large sets with more unique individuals.

However, exchanging all the slots in the set in each iteration can provide good solutions for small sets with few unique individuals. Specify “WholeSet” instead of an integer if that is your preferred option. This is equivalent to producing a new initial set in each iteration and keeping the best one.

Recommended values: 10-100.

**AlreadySequencedFile**

File with previous sequencing data. Column 1 is the ID of the individual, column 2 is the coverage already existent for that individual; no headers.

Specify “No” if there is no previous sequencing data.

**ExcludedFile**

File with a list of animals that cannot be sequenced (e.g., because they have no sample available). Just 1 column with the ID of the individuals; no headers.

Specify “No” if there are no individuals to exclude.

**FilterRareHaplotypes**

This parameter can be set to No, All or Context. This parameter specifies the approach to handle haplotypes with low population frequency.

If this parameter is No, no filtering is done and all rare haplotypes are targeted by the algorithms.

If this parameter is All, all haplotypes below a certain population frequency are filtered out and excluded from being targeted by the algorithms.

If this parameter is Context, rare haplotypes are filtered based on flanking context and only those that could have derived from a recombination event between common haplotypes are targeted by the algorithms. For this option to work, **ContextCountFlankingHaps** and **FilterRareHaplotypes** should be defined as detailed below.

**PopulationFrequencyToFilter**

When **FilterRareHaplotypes** is set to All or Context, this parameter specifies which population frequency (expressed as count) should be used as a threshold. Any haplotype with population count equal or lower than this value will be filtered using the specified approach.

Use 1 for filtering singletons, 2 for filtering singletons and doubletons, or greater if desired (according to the population size).

**ContextCountFlankingHaps**

When **FilterRareHaplotypes** is set to Context only, this parameter specifies the minimum population count required for the flanking haplotypes. Rare haplotypes flanked by haplotypes with a population count lower than this value will be filtered out. Rare haplotypes flanked by common haplotypes with a population count equal or greater than this value will remain as targets.

This value should be set according to **PopulationFrequencyToFilter** and the population size.

**ContextCountCombinationFlanks**

When **FilterRareHaplotypes** is set to Context only, this parameter specifies the maximum population count allowed for the particular combination of haplotypes flanking the rare haplotype at each side. Rare haplotypes flanked by a common combination of flanking haplotypes with a population count greater than this value will be filtered out. Rare haplotypes flanked by a rare combination of flanking haplotypes with a population count equal or lower than this value will remain as targets.
A recommended value is the same as in PopulationFrequencyToFilter, but it can be set differently according to the population size.

**SecondaryHapSeqThresh**

This parameter sets a secondary target sequencing coverage for the haplotypes with population frequency below a certain threshold specified in PopFreqForSecondThresh.

Specify an integer to be used as secondary target coverage or type “No” if you do not want to use a secondary target coverage.

**PopFreqForSecondThresh**

When SecondaryHapSeqThresh is set to an integer, this parameter sets the maximum population frequency of the haplotypes targeted at the secondary coverage. Haplotypes with population frequency equal or lower than this value will be targeted at the secondary coverage.

### 3.1.2 INPUT FILES ALPHASEQOPT METHOD 1

### 3.1.3 INPUT FILES ALGORITHM 1

Existing phased, true or imputed genotype, haplotype or sequence data.

Phased genotype, haplotype or sequence data for each individual in the population to consider for sequencing should be supplied as follows.

**Phased genotype/haplotype/sequence data (compulsory if haplotype IDs files not used)**

Phased genotype/haplotype/sequence files should be placed in a subdirectory of the working directory. The subdirectory should be named “FinalHaplotypes” and should contain phased genotype files for each chromosome. Files for each chromosome should be labelled as “ChromosomeX.txt”, where X is the chromosome number from 1 to NumberOfChromosomes. Genotypes/sequences for each base pair position should be coded in the 0/1 format. Missing values should be coded as ‘9’. Each individual should have two lines in the file, one specifying the phase of its first haplotype and the other specifying the phase of the second haplotype. The algorithm is insensitive to whether the paternal or maternal haplotype is supplied first. The file should be space separated only, and should contain no tabs.

**Haplotype ID data (compulsory if phased genotype/haplotype/sequence files not used)**

If haplotypes of each individual have been pre-phased and coded as integers by haplotype ID rather than 0/1 format then the algorithm can make use of this information.

Files with haplotype IDs for each individual to consider for sequencing must be supplied separately for each chromosome in a subdirectory of the working directory named “Cores”. Files for each chromosome should be labelled as “ChromosomeX.txt”, where X is the chromosome number from 1 to NumberOfChromosomes. These files should have one line per individual, with two columns per core specifying the paternal and maternal haplotypes (the algorithm is insensitive to whether the paternal or maternal haplotype is supplied first). Missing values should be coded as ‘-99’. The file should be space separated only, and should contain no tabs.

### 3.1.4 INPUT FILES ALGORITHM 2

**Matrix of sequencing scenarios file (compulsory)**

The path to this file must be specified using the MatrixOfScenariosFile parameter in the AlphaSeqOptSpec.txt file. This file contains a matrix of different family sequencing scenarios (i.e. the selected sequencing coverages for a focal
individual, its parents and its grandparents). The file should be space separated and have seven columns of integers representing coverages with no header. The seven columns should be coverages for the family members in the following order: Column 1 - Focal individual; Column 2 - Sire; Column 3 - Dam; Column 4 - Paternal Grand sire; Column 5 - Paternal Grand dam; Column 6 - Maternal Grand sire; Column 7 - Maternal Grand dam. Each row corresponds to a single family sequencing scenario.

Matrix of metrics file (compulsory)

The path to this file must be specified using the MatrixOfMetricsFile parameter in the AlphaSeqOptSpec.txt file. This file contains a matrix of expected haplotype phasing accuracies for each individual in a focal family given a family sequencing scenario. Therefore, each line in this file must correspond directly to the scenarios in the MatrixOfScenariosFile. The file should be space separated and should have seven columns of numbers between 0-1 representing phasing accuracies given a sequencing scenario with no header. The seven columns should be accuracies for each family member in the following order: Column 1 - Focal individual; Column 2 - Sire; Column 3 - Dam; Column 4 - Paternal Grand sire; Column 5 - Paternal Grand dam; Column 6 - Maternal Grand sire; Column 7 - Maternal Grand dam.

Prior sequenced individuals file (optional)

The path to this file must be specified using the SequencedIndividualFile parameter in the AlphaSeqOptSpec.txt file. This file is a list of list of individuals that already have sequence data and that should not be considered for sequencing. Each individual ID should be supplied on a single line and should exclude whitespace.

Prior sequencing costs file (optional)

The path to this file must be specified using the PriorSeqCostsFile parameter in the AlphaSeqOptSpec.txt file. This file contains information on existing coverage and prior sequencing costs incurred for an individual in a focal family. This file is space separated and has 4 columns:

Column 1: Individual ID
Column 2: Prior library cost incurred for this individual (exclude currency signs)
Column 3: Existing average sequencing coverage for this individual
Column 4: Prior sequencing cost incurred for this individual (per 1x coverage)

Information for each individual ID should be supplied on a single line.

Top individuals file (compulsory)

The path to this file must be specified using the TopIndividualsFile parameter in the AlphaSeqOptSpec.txt file. If running the two algorithms consecutively then this file is not required, since the information of the top individuals to sequence as determined by algorithm 1 will be passed internally directly to algorithm 2. If only algorithm 2 is being used then this file must be supplied. This file can be the direct output of algorithm 1 i.e. the TopIndividualsToSequence.txt file. Otherwise the supplied file must have the same format as the TopIndividualsToSequence.txt output file (see output files description below).

3.1.5 INPUT FILES BOTH ALGORITHMS

Pedigree.txt (compulsory)

The Pedigree.txt file contains 3 columns separated by spaces as follows:
Column 1: Individual ID; Column 2: Sire ID; Column 3: Dam ID

In algorithm 1, this file is only required to read in the IDs of individuals in the population. As such, sire and dam ID columns can be set to “0” (i.e. missing) regardless of whether they are known or not, but may be supplied if the user wishes.

In algorithm 2, this file is necessary to determine the relationships between focal families when assigning the optimum sequencing coverage for an individual. Therefore, sire and dam IDs should be provided if known. If not known then sire and dam IDs should be set to missing (i.e. 0).

If both algorithms are run consecutively (i.e. the OptimisationMethod parameter in the AlphaSeqOptSpec.txt file (see above) is set as Sequenceandcoverage) then the full pedigree file should be supplied. All individuals in the genotype/haplotype/sequence files should be present in this file.

Individuals not to sequence file (optional)

The path to this file must be specified using the IndividualsNotToSequenceFile parameter in the AlphaSeqOptSpec.txt file. This file is a list of list of individuals that already have sequence data and that should not be considered for sequencing. Each individual ID should be supplied on a single line and should exclude whitespace.

3.1.6 INPUT FILES ALPHASEQOPT METHOD 2

Haplotype ID data (compulsory if phased genotype/haplotype/sequence files not used)

If haplotypes of each individual have been pre-phased and coded as integers by haplotype ID rather than 0/1 format then the algorithm can make use of this information.

Files with haplotype IDs for each individual to consider for sequencing must be supplied separately for each chromosome in a subdirectory of the working directory named “Cores”. Files for each chromosome should be labelled as “ChromosomeX.txt”, where X is the chromosome number from 1 to NumberOfChromosomes. These files should have one line per individual, with two columns per core specifying the paternal and maternal haplotypes (the algorithm is insensitive to whether the paternal or maternal haplotype is supplied first). Missing values should be coded as ‘-99’. The file should be space separated only, and should contain no tabs.

3.1.7 SEED FILE

If desired, a seed can be specified in a Seed.txt file in the same location as the AlphaSeqOptSpec.txt file. If provided, the seed must be a negative integer. If not provided, the system will generate one.

3.2 OUTPUT FILES ALPHASEQOPT METHOD 1

The output of AlphaSeqOpt depends on which algorithm is being used.

In addition to output files, AlphaSeqOpt prints statements to screen to show the progress of each algorithm and any error statements. Error statements will terminate the program.

3.2.1 OUTPUT FILES ALGORITHM 1

Algorithm 1 has the following output files:

- TopIndividualsToSequence.txt or TopIndividualsToSequenceTMatrix.txt
For each chromosome (X is a placeholder for chromosome number):

- HaplotypesIndividualsCarryFullChromosomeX.txt or HaplotypesIndividualsCarrySummaryChromosomeX.txt
- ImputedHaplotypesIndividualsCarryChromosomeX.txt
- HaplotypesIndividualsCarryPerCoreChromX.txt

A more detailed description of each file is provided below.

**TopIndividualsToSequence.txt**

This file is printed if OptimisationMethod is set to Sequence or Sequenceandcoverage. This file is space separated and has 7 columns and a header line. Each line in the file corresponds to information for each individual in the population. The columns are as follows:

Column 1: ID of individual

Column 2: Conditional count of the number of haplotypes this individual shares with the rest of the population (conditional on haplotypes being accounted for in previously sequenced individuals above it in the list)

Column 3: Total count of the number of haplotypes this individual shares with the rest of the population

Column 4: Total possible number of haplotypes this individual could share with the rest of the population

Column 5: Conditional percentage of haplotypes this individual shares with the rest of the population (i.e. column 2 / column 4)

Column 6: Potential percentage of haplotypes this individual shares with the rest of the population (i.e. column 3 / column 4)

Column 7: Total number of possible haplotypes in the population if all individuals carried unique haplotypes

The last 2 lines of this file are a summary of the file contents giving the sum of column 5 (OverallPopFootprint, will be 100% if all individuals are considered for sequencing) and the sum of column 2 (TotPossCountShared).

**TopIndividualsToSequenceTmatrix.txt**

This file is printed if OptimisationMethod is set to Tmatrix. This file is space separated and has 2 columns and a header line. Each line in the file corresponds to information for each individual in the population. The columns are as follows:

Column 1: Individual ID

Column 2: Pedigree-inferred genetic contribution of an individual

**FreqOfHaplotypesIndividualsCarry.txt**

This is a space separated file with a header and contains 4 columns and one line for each individual considered for sequencing. Columns contain the following information:

Column 1: ID of individual

Column 2: Total number of haplotypes carried by this individual
Column 3: Total number of haplotypes in the population
Column 4: Proportion of haplotypes carried by this individual

**NumberOfHaplotypesPerCore.txt**

This is a space separated file with a header and contains 3 columns. Columns contain the following information:

Column 1: Chromosome number
Column 2: Core number on chromosome
Column 3: Number of haplotypes in this core on this chromosome

**SummaryPedigree.txt**

This is a space separated file with a header and contains 7 columns. Each line corresponds to a single individual considered for sequencing. Columns contain the following information:

Column 1: Individual ID
Column 2: Sire ID
Column 3: Dam ID
Column 4: Paternal grandsire ID
Column 5: Paternal granddam ID
Column 6: Maternal grandsire ID
Column 7: Maternal granddam ID

**CoreIndex.txt**

This is a space separated file with a header and contains 6 columns. Columns contain the following information:

Column 1: Core number
Column 2: Chromosome number
Column 3: Core start position if no SNPs are excluded (due to not passing the phasing threshold set in parameter `SnpPhasedThreshold`)
Column 4: Core end position if no SNPs are excluded (due to not passing the phasing threshold set in parameter `SnpPhasedThreshold`)
Column 5: Core start position if SNPs that do not pass the phasing threshold set in parameter `SnpPhasedThreshold` are excluded
Column 6: Core end position if SNPs that do not pass the phasing threshold set in parameter `SnpPhasedThreshold` are excluded

**HaplotypesIndividualsCarryFullChromosomeX.txt**

This file is printed if the `HaplotypePrinting` parameter is set to `All`.
This file is space separated and will contain two lines of the full phase for each individual, one for each haplotype.
The first column of each line is the ID of the individual.
HaplotypesIndividualsCarrySummaryChromosomeX.txt

This file is printed if the HaplotypePrinting parameter is set to Summary.

This file is space separated and will contain two lines of the summary phase for each individual, one for each haplotype. The summary phase is constructed only of the SNPs which exceed the threshold set by the SnpPhasedThreshold parameter.

The first column of each line is the ID of the individual.

ImputedHaplotypesIndividualsCarryChromosomeX.txt

During the process of defining shared haplotypes between individuals, if an individual is not fully phased but exceeds the IndividualPhasedThreshold parameter then it’s haplotype may be fully inferred/imputed based on haplotype sharing. This imputed haplotype is given in this file. This file is space separated and will contain two lines of the imputed phase for each individual, one for each haplotype. The first column of each line is the ID of the individual.

HaplotypesIndividualsCarryPerCoreChromX.txt

This file is space separated and will contain one line for each individual. Each line will contain the numeric ID of the two haplotypes carried by each individual in each core. Missing values are given as ‘-99’. The first column of each line is the ID of the individual.

HaplotypesIndividualsCarryPerCore.txt

This file contains has the same information as “HaplotypesIndividualsCarryPerCoreChromX.txt” but for all the chromosomes. This file can be used as an input for AlphaSeqOpt method 2.

3.2.2 OUTPUT FILES ALGORITHM 2

Algorithm 2 has the following output files:

- OutputFamilyScenarios.txt
- OutputFamilyAccuracies.txt
- OutputFamilyPropHaplotypes.txt
- OutputFamilyRealisedAccuracies.txt
- OutputCoverageFinal.txt at program termination or OutputCoverageCurrent.txt for each optimisation round
- OutputIteration.txt

A more detailed description of each file is provided below.

OutputIteration.txt

This file contains the same information as that printed to screen, and contains information on the progress of the algorithm. This is a space separated file with a header and contains 3 columns. Columns contain the following information:

Column 1: Round number
Column 2: Criterion (a measure of how good the sampled sequencing scenarios are in this round)
Column 3: Cost of the sequencing scenarios in this round

**OutputCoverageCurrent.txt**

This file is printed every round and contains the set of sampled coverage for each individual in the selected focal families. This is a space separated file with a header and contains 2 columns. Columns contain the following information:

Column 1: Individual ID
Column 2: Sequencing coverage

**OutputCoverageFinal.txt**

This file is printed in the final round of the algorithm and contains the most optimum set of sampled coverage for each individual in the selected focal families. This is a space separated file with a header and contains 2 columns. Columns contain the following information:

Column 1: Individual ID
Column 2: Sequencing coverage

**OutputFamilyScenarios.txt**

This file contains the sequencing coverages sampled for each focal family. This is a space separated file with a header and contains 8 columns. Each line corresponds to a focal family. Columns contain the following information:

Column 1: Family number (from 1-N focal families)
Column 2: Focal individual sampled sequencing coverage
Column 3: Sire sampled sequencing coverage
Column 4: Dam sampled sequencing coverage
Column 5: Paternal grandsire sampled sequencing coverage
Column 6: Paternal granddam sampled sequencing coverage
Column 7: Maternal grandsire sampled sequencing coverage
Column 8: Maternal granddam sampled sequencing coverage

**OutputFamilyAccuracies.txt**

This file contains the accuracies of phasing each individual in a focal family given a sampled family sequencing scenario. This is a space separated file with a header and contains 8 columns. Each line corresponds to a sequencing scenario for a given focal family. Columns contain the following information:

Column 1: Family number (from 1-N focal families)
Column 2: Focal individual phasing accuracy
Column 3: Sire phasing accuracy
Column 4: Dam phasing accuracy
Column 5: Paternal grandsire phasing accuracy
Column 6: Paternal granddam phasing accuracy
Column 7: Maternal grandsire phasing accuracy
Column 8: Maternal granddam phasing accuracy

OutputFamilyPropHaplotypes.txt

This file contains the proportion of the population haplotypes carried by each individual in a focal family as given in the TopIndividualsToSequence.txt file. This is a space separated file with a header and contains 9 columns. Each line corresponds to a sequencing scenario for a given focal family.

Columns contain the following information:
Column 1: Family number (from 1-N focal families)
Column 2: Focal individual proportion of the population haplotypes
Column 3: Sire proportion of the population haplotypes
Column 4: Dam proportion of the population haplotypes
Column 5: Paternal grandsire proportion of the population haplotypes
Column 6: Paternal granddam proportion of the population haplotypes
Column 7: Maternal grandsire proportion of the population haplotypes
Column 8: Maternal granddam proportion of the population haplotypes
Column 9: Sum of columns 2:8

OutputFamilyRealisedAccuracies.txt

This file contains the realised phasing accuracy of each individual in a focal family given in the sampled sequencing scenarios and phasing accuracies and the proportion population haplotypes carried by each individual. This is a space separated file with a header and contains 9 columns. Each line corresponds to a focal family. Columns contain the following information:
Column 1: Family number (from 1-N focal families)
Column 2: Focal individual realised phasing accuracy
Column 3: Sire realised phasing accuracy
Column 4: Dam realised phasing accuracy
Column 5: Paternal grandsire realised phasing accuracy
Column 6: Paternal granddam realised phasing accuracy
Column 7: Maternal grandsire realised phasing accuracy
Column 8: Maternal granddam realised phasing accuracy
Column 9: Sum of columns 2:8

3.3 OUTPUT FILES ALPHASEQOPT METHOD 2

The method 2 of AlphaSeqOpt generates the following output files:
- InitialInd.txt and FinalInd.txt
- InitialHapSeqFreq.txt and FinalHapSeqFreq.txt
3.3.1 InitialInd.txt and FinalInd.txt

These files are the main output of the method 2 of AlphaSeqOpt. These files contain the list of individuals for sequencing selected in the initial set and in the refined set, respectively. Column 1 is the individual ID and column 2 is its target sequencing coverage.

3.3.2 InitialHapSeqFreq.txt and FinalHapSeqFreq.txt

These files contain the frequency of the target haplotypes in the initial set and in the refined set, respectively. This frequency is twice the expected coverage that results for that haplotype if the corresponding set is sequenced. Column 1 is core ID, column 2 is haplotype ID and column 3 is the frequency of that haplotype.

3.3.3 AlphaSeqOpt_log1_SelectInitialSet.txt

This file contains information of each iteration of step 1 of the method (selection of the initial set). This file has 7 columns and a header line. The columns are as follows:

Column 1: Iter. Iteration number.

Column 2: IndAdded. Individual ID of the individual added in that iteration.

Column 3: nIndMaxScore. Number of individuals that had the maximum individual score. If greater than 1, the individual added was chosen randomly between these number of individuals.

Column 4: MaxIndScore. Maximum individual score calculated in that iteration.

Column 5: HapsAtThresh%. Percentage of target haplotypes with an expected sequencing coverage equal to or greater than the target haplotype coverage. This is the criterion parameter for optimisation.

Column 6: CumulatedX. Total cumulated coverage generated across individuals in the set.

Column 7: CumulatedInd. Total number of unique individuals that are sequenced in the set.

3.3.4 AlphaSeqOpt_log2_RefineSet.txt

This file contains information of each iteration of step 2 of the method (refinement of the set). This file has 5 columns and a header line. The columns are as follows:

Column 1: Iter. Iteration number.

Column 2: SetKeptFlag. This is a flag variable to indicate if the set produced in that iteration improved on the previous set and therefore was kept (1) or if it produced worse results than the previous set and was therefore discarded (0).

Column 3: HapsAtThresh%. Percentage of target haplotypes with an expected sequencing coverage equal to or greater than the target haplotype coverage. The value displayed is for the best solution kept. This is the criterion parameter for optimisation.
Column 4: CumulatedX. Total cumulated coverage generated across individuals in the set. The value displayed is for the best solution kept.

Column 5: CumulatedInd. Total number of unique individuals that are sequenced in the set. The value displayed is for the best solution kept.

### 3.3.5 PopHapFreq.txt

This file contains the frequency of all the haplotypes in the population and a flag to indicate if the haplotype is targeted for sequencing or if it is filtered out. Column 1 is core ID, column 2 is haplotype ID, column 3 is the population frequency of that haplotype, and column 4 indicates if that haplotype is targeted for sequencing (1) or not (0).
Examples using simulated datasets (simulations done using AlphaSim) of how the algorithms can be run, the expected input and output after running the algorithms and full descriptions can be found on the AlphaSuite website (http://www.alphagenes.roslin.ed.ac.uk/).