User's guide for software
COLONY
Version 2.0.6.5 (July 30, 2018)

Jinliang Wang
Institute of Zoology
Zoological Society of London
Regent's Park, London NW1 4RY, UK

Email: Jinliang.wang@ioz.ac.uk
Software Webpage: http://www.zsl.org/science/research-projects/software/

Table of Contents

1. Introduction

The Colony program can be run in several computer platforms, including Windows, Mac, Linux, Unix. This documentation is prepared particularly for Windows users, but is also useful for users of other platforms.

1.1 Overview

Colony is a computer program implementing one likelihood method and two pairwise likelihood methods to assign/infer parentage, sibship and clonemates (duplicates) among individuals using their multi-locus genotypes. The methods are formally described in the following papers.

(1) Wang J. 2004. Sibship reconstruction from genetic data with typing errors. *Genetics* **166**: 1963-1979.
(2) Wang J & Santure AW. 2009. Parentage and sibship inference from multi-locus genotype data under polygamy. *Genetics* **181**: 1579-1594.
(3) Jones OR & Wang J. 2010. COLONY: a program for parentage and sibship inference from multilocus genotype data**.** *Molecular Ecology Resource*s **10**: 551-555.
(4) Wang J. 2012. Computationally efficient sibship and parentage assignment from multilocus marker data. *Genetics* **191**: 183-194.
(5) Wang J. 2013. An improvement on the maximum likelihood reconstruction of pedigrees from marker data. *Heredity* **111**: 165-174.
(6) Wang J. 2013. A simulation module in the computer program COLONY for sibship and parentage analysis. *Molecular Ecology Resource*s **13:** 734-739.
(7) Wang J & Scribner KT. 2014. Parentage and sibship inference from markers in polyploids. *Molecular Ecology Resource*s **14**: 541–553.
(8) Wang J. 2016. Individual identification from genetic marker data: developments and accuracy comparisons of methods. *Molecular Ecology Resource*s **16**: 163-175.

Colony can be used, among others, in estimating full- and half-sib relationships, inferring clones or duplicated individuals, assigning parentage, reconstructing parental genotypes, inferring mating systems (polygamous / monogamous, selfing rate) and reproductive skew, and re-estimating genotyping errors at each marker locus. It applies to both diploid and haplo-diploid species, monoecious and dioecious species. With slight modification of the data, it can also be applied to polyploid species (Wang & Scribner 2014). It can use codomiant and domiant marker data with or without genotyping errors. The Windows GUI version of Colony can also be used to simulate genotype data with a particular sibship & parentage structure, and the simulated genotype data can be used to check the accuracy of various pedigree reconstruction methods, and the marker information sufficiency.

In brief, the method assumes a sample of individuals subdivided into 3 sub-samples: offspring (OFS), candidate males (CMS) and candidate females (CFS). OFS is essential while CMS and CFS are both optional. Individuals in OFS are assigned (clustered) to K1 paternal and K2 maternal families (where K1 and K2 are unknown), and individuals in CMS and CFS, if available, are assigned or unassigned paternity and maternity to these K1 and K2 families. It is assumed that offspring individuals are either duplicates (or members of a clone), full sibs (sharing both parents), half sibs (sharing only one of the two parents), or unrelated (sharing no parents), while candidates are assumed unrelated among themselves and are either parents of or unrelated to the offspring. Markers are assumed to be in linkage equilibrium. Violation of these assumptions may lower the power of the analysis, but could be compensated by using more informative markers (Wang 2004). For example, the information about the

sex and age of the sampled individuals might be unavailable. In such a case, each individual is allowed to appear in all 3 sub-samples and the sibship and parentage are still inferred satisfactorily in some cases (Wang & Santure 2009). Similarly, the presence of background relationships (such as cousins and avunculate relationships which are assumed to be either absent or unrelated by the method) could reduce the accuracy. However, the accuracy is quickly improved by an increasing amount of marker information. The current model accounts for deviations from Hardy-Weinberg equilibrium. When desired, inbreeding (due to mating between close relatives or selfing, or due to population structure) can be accounted for and estimated together with the relationship structure.

The analysis results from the Colony program include mainly:

- Full and half sibship assignments among individuals in OFS;
- Paternity (if CMS available) and maternity (if CFS available) assignments;
- Duplicated individuals in OFS;
- Genotype inference at each locus of each offspring;
- Genotype inference at each locus of each parent, no matter it is assigned to a candidate in CFS, CMS or not;
- Possible genotype errors at each locus of each offspring;
- Possible genotype errors at each locus of a candidate assigned parentage;
- Inbreeding and self-fertilization (selfing rate for monoecious species) when the inbreeding model is used;
- Refined allele frequency estimates taking into account of the inferred relationships;
- Refined estimates of genotyping error rates at each locus;
- Effective population size from the estimated frequency of siblings.

The software package Colony includes the executable for Windows, user's guide, example datasets and example analysis results. The computational part of Colony program was written in Fortran 90/95, and the GUI front end for Windows was written in Visual Basic. The Windows GUI allows users to prepare input data and analysis parameters, to run the program, to view analysis results, and to monitor and plot intermediate results during the run. Colony packages for Linux and Mac platforms are also available on the same website as the package for Windows.

1.2 Features of the methods and software

The current version of Colony has the following features:

- Allowing for polygamy for both males and females. In other words, offspring are allowed to be maternal half sibs, paternal half sibs, full sibs, clonal mates (or duplicates), and unrelated, and all these relationships are inferred jointly;
- Inferring clonal mates (or duplicates) against sibling relationships, accounting for genotyping errors;
- Allowing for the inference of parentage simultaneously with sibship;
- Estimating population allele frequencies simultaneously by taking into account of the reconstructed relationship in and among the 3 sub-samples, using Bayes' theorem;
- Accounting for genotyping errors and mutations in data for relationship reconstruction;
- Detecting genotyping errors and mutations in individual genotypes;
- Refining estimates of genotyping error rates at each locus;
- Inferring genotypes of individuals (offspring and inferred parents) who have no genotype data;
- Applicable to both diploid and haplodiploid species, to both dioecious and monoecious species;
- Applicable to a sample of diploid offspring, haploid offspring, or both;

- Allowing for and estimating inbreeding, and inferring selfing rate for monoecious species;
- Choices of the full likelihood method and a new likelihood score method (Wang 2012);
- Choices of different priors or no priors for sibship assignments;
- Estimating the current effective size of a population from sibship assignments;
- Using both co-dominant and dominant markers;
- Utilizing known relationships together with marker data;
- Allowing for parallel computation (by openMP and MPI) using multiple cores/CPUs;
- Simulating a 2- or 1-generation genotype dataset with known relationships for analysis by Colony or other pedigree reconstruction programs;
- Windows GUI;
- Batch run of multiple datasets.

1.3 Bug report

Colony (Copyright 2008 by Jinliang Wang) is available, free of change, for academic use only. It is downloadable from the website http://www.zsl.org/science/research-projects/software/. Any update of the program will also be put in the same website. A brief record of the updating history is available as a separate document for downloading, and is viewable in Colony. Every effort has been made to implement the methods correctly and efficiently, but there is no guarantee that the program is free of bugs. Reports of bugs are welcome, and should be sent to: jinliang.wang@ioz.ac.uk?subject=Colony.

2. Installation

Colony software has 3 packages in 3 zipped files on the website for Windows, Macintosh and Linux respectively. The Windows version is preferred because it has a GUI and the simulation module, and is more extensively tested. Macintosh users can also install a Windows simulator to run the Windows version of Colony.

Upon downloading and unzipping the zipped file for Windows, one obtains an installation file named "Colony2.msi". Double click on this file to start the installation. By default, Colony will be installed in "C:\ZSL\Colony". However, you can change the directory where Colony will be installed during the installation process. *It is suggested that it NOT be installed in the "Windows" directory or the "Program Files" directory*. Otherwise, due to windows security issues, subsequent input and output files of Colony are automatically moved to a folder in VirtualStore and, the simulation program may not run properly.

To run Colony after installation, just double click the Colony icon on your desktop, or alternatively, click **`Start`**→**`Programs`**→**`Colony`**.

Colony should run on a PC or server with Microsoft Windows operating system 2000/XP, Vista, 7, 8 and 10. It requires the .Net Framework 2.0 (or higher) installed on your computer. If you have a recent version of Windows, you probably already have .Net installed on your computer. You can check by clicking **`Start`** on your Windows desktop, selecting **`Control Panel`**, and then double-clicking the **`Add or Remove Programs`** icon. When that window appears, scroll through the list of applications to check whether Microsoft .Net Framework 2.0 (or higher) is listed. If it is not installed, you need first download it from the Microsoft website and install it before installing Colony.

For Windows 7 64bits, after installing Colony, you may need first to grant read-and-write access to the Colony program folder. For details, see "14.7 File access problems in Windows 7". If you have

problem running Colony in parallel with multiple threads, then you need to copy the two files in the subfolder "OS64" into the Colony program folder to replace the original 2 files there.

This PDF file of the user's guide is included in the package. I suggest you to print and read this file carefully before running Colony. This is because of the following reasons. (1) Several input files with specific format requirements are needed for setting up the project. These files must be prepared before running Colony. (2) The analysis results are in many output files, and it is important to correctly interpret the results.

It is also highly recommended that the user first test the program by running at least one of the example datasets provided. The analysis on one's own dataset using the program involves the following steps.

3. Empirical data input – Windows GUI

This section describes how to set up a Colony project of an empirical dataset, using Colony's Windows GUI. In the GUI, all input and output files are organized into "projects". The user gives a project name when setting up a new project, and a folder with this name is created in the directory where the Colony program is installed. All subsequent input files and, after running Colony, output files of the project are put in this project folder. The project folder cannot be moved to other locations for Colony to work on it.

The GUI has limited capacity to deal with large datasets, because it takes a lot of memory to display the data in the formatted table form. For most computers, it should have no problem to cope with a dataset with up to 2000 individuals and 2000 loci. Beyond the limits, one should consider inputting the data and running Colony in non-GUI mode, as described below.

Follow the steps below to set up a new project and input data into the project. It is suggested that all input files described below are prepared in the required format before running Colony to set up a new project. **These files must be in pure text file format, using either commas, tabs, or white space as delimiters**. Empty rows take no effect. The contents of a row required by Colony can appear in multiple consecutive lines, with a line continuation mark "&" at the end of a line. Therefore, the symbol "&" cannot be used for other purposes, such as in an individual ID. For example, a row for the ID and genotypes at 3 loci of an individual may look like

```
IndividualXXX    124 128    212  214   144  144
```

The row can be arranged in multiple continuous lines, like

```
IndividualXXX  &
  124 128  212  214  &
  144  144
```

Note, the line break mark "&" should NOT sit within a string (like `IndividualXXX`) or a numerical value (e.g. 124 or 12.54), and should always prefixed with one or more blank spaces.

3.1 Set up a new project

Click on **`File→New Project`** (alternatively, click the new project Tool Menu Button) to open up a new project set up wizard (Figure 1). You are asked to give a project name, which should be a string containing less than 40 letters and numbers (others such as space, comma, full stop, forward and backward slashes are not allowed in the project name). You are also asked about the project type, and

the “Empirical Data Analysis” should be chosen herein. When the OK button is clicked, a folder with the project name is created in the directory where Colony is installed. All input and output files wll be stored in this folder. All output files will use the same project name but different (self-explainable) extension names. The next time one runs Colony to load the project, one can use **`File→Open Project`** (alternatively, **`File→Recent Projects`**, if the project is a recent one) to open the project folder.

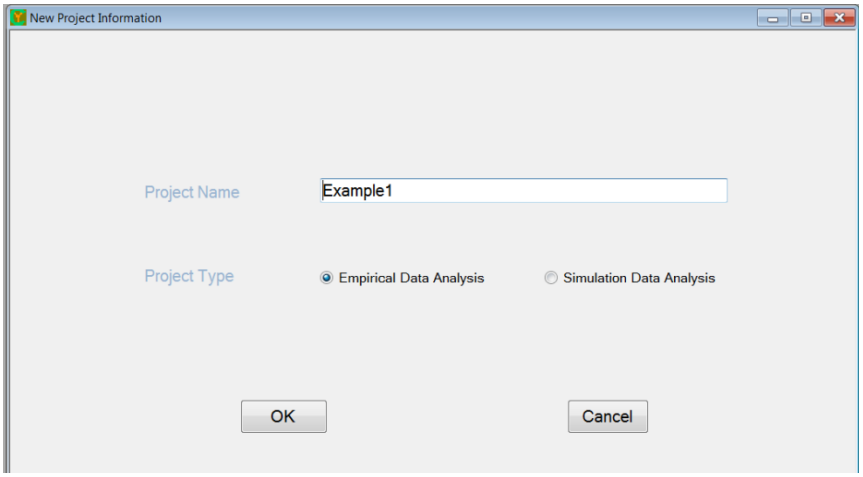


Figure 1: Setting up a new project. The user specifies the project name (here “example1”, in practice it is better to give a more meaningful name)

3.2 Input parameters

When the “OK” button is clicked in the previous step, a new window headed by “New Project Wizard: Input an empirical dataset” shows up. In the new window, 10 pages are provided to input your data. Inputs into these pages are sequential such that only when all previous pages are completed and checked can the following page be accessed. Similarly, if one goes back to a previous page and make any changes there, the following pages may lose the data already inputted, or left unchecked. This is because the data input in a previous page may affect the (validity of) data input in a following page. In case there is any problem in the data or the data format in a certain page, you may (1) exit the new project wizard, change the data using Colony’s built-in text file editor (in **`File→Open File`**), and rerun Colony; or more conveniently (2) use an external editor (such as notepad) to change and save the data and then continue setting up the project.

The new project wizard takes in the information given in each page, saves the data with a specific file name in the project folder (adding column headers if necessary), and assembles all of the data and parameters into a single (default) input file named “Colony2.dat”, which is saved in the project folder on completion of the data input process.

In page 1 (see Figure 2), a number of parameters are required to be set. In most cases, the default values of the parameters are fine.

(1) ***Mating system-I***: You are requested to specify the male and female mating system. Here in this specific context, male “monogamous” means that two offspring in the OFS sample must be fathered by 2 different males if they have separate mothers. In other words, *male “monogamous” specifies that no paternal halfsibs exist in the OFS sample*. Note that the mating system herein is defined with regard to the samples being analyzed, not to the population or species from where the samples are taken. For example, consider a population in which males mate singly with females in a

breeding season but mate with different females in different breeding seasons. An OFS sample with individuals taken from multiple breeding seasons may contain offspring from different mothers but from a single male (i.e. paternal halfsibs). Therefore, for the purpose of the Colony analysis, the male mating system should still be set as "polygamous". The female mating system is similarly defined. Note also that when both males and females are defined as polygamous, the markers are few and have genotyping errors and no sibship prior is used, the computation for the FL method can become very slow simply because all offspring (related or not) in the OFS can be inferred to be related in the pedigree (see Figure 2A, for example) and must be considered together in computing the likelihood of a configuration.

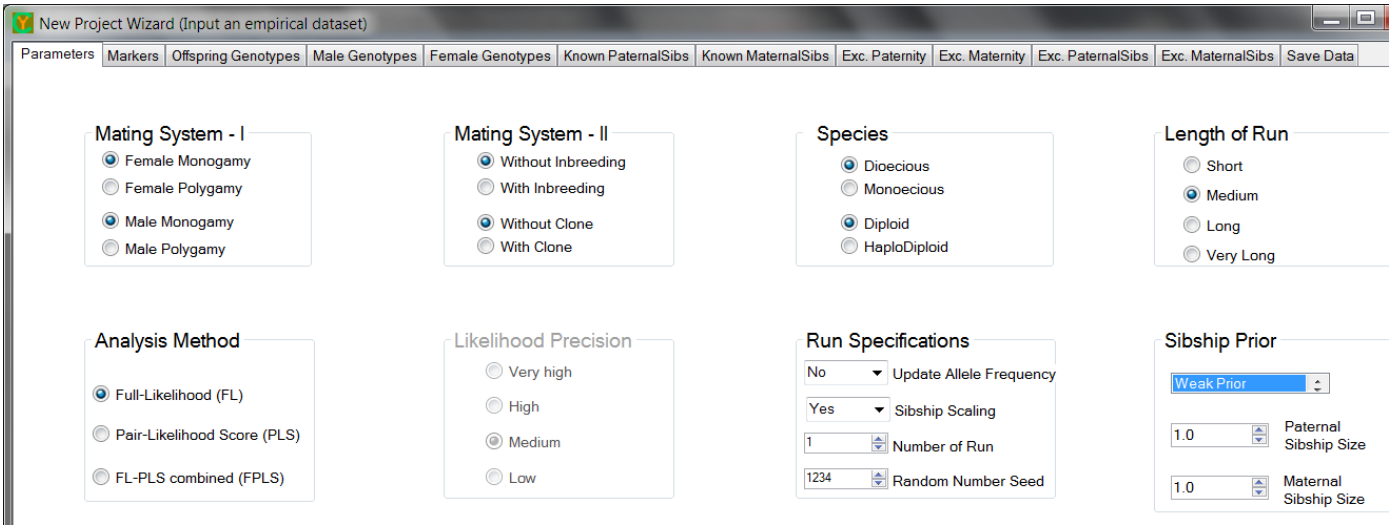


Figure 2: Parameters for the new project.

(2) ***Mating system-II***: You are also allowed to define whether there is inbreeding or not. When inbreeding is absent, the population is assumed to be in Hardy-Weinberg equilibrium and genotype frequencies are calculated using the HW law. In such a case, inbreeding is not inferred and all offspring are assumed to come from outbreeding for monoecious. When inbreeding is present, a population level (average) inbreeding coefficient, equivalent to Wright's F_{IS}, is inferred iteratively with other parameters (e.g. relationship, allele frequencies) and is used for calculating genotype frequencies. In such a case, inbreeding is inferred and offspring are assumed to come from both outbreeding and selfing for monoecious. Note that, for dioecious, the non-inbreeding model is recommended except when there is strong evidence of inbreeding and the inbreeding level is high. Otherwise, using the inbreeding model for data with low or no inbreeding could slow down the computation substantially with little or no improvement for the estimation of relationships.

You are also given the choice of inferring clones (duplicates) or not. For the choice of "Without Clone", all offspring multilocus genotypes are assumed to come from distinctive individuals who are not clone mates. In other words, all offspring multilocus genotypes are assumed to be different by default. Any identical multilocus genetypes must come by chance due to limited marker information, or must be due to mistypings, or both causes. With this choice, clones (duplicates) are not inferred.

For the choice of "With Clone", it is assumed that some offspring may have the same or similar multilocus genetypes because they are duplicated or come from the same clone, although they may have flightly different multiplocus phenotypes due to mistypings. With this choice, sibship is first

inferred. Then given an inferred full sibship, individuals in this full-sib family are partitioned into clone clusters by maximizing the likelihood of clone configuration. This two step process works very well, as checked by simulations (Wang 2016), for both sibship and clone inference when the size (i.e. the number of individuals) of a clone is small. However, when clone size is big (say, >10 individuals in a clone), the Mendelian segregation is distorted which leads to a possible split of a full sib family. If one cares only for the inference of clones, it is not a problem. Otherwise, two options are possible to overcome the problem. The 1st is to run the original dataset using the full likelihood method, allowing for duplicates. If large clones are detected, then full sibship might be split. One can remove all members but one of an inferred clone, and re-run the reduced dataset by the same method. If no large clones are detected, then there is no need to re-run the data. The 2nd is to use the likelihood score method, which is resilient to distortions of Mendelian segregation because it considers pairs of individuals.

(3) ***Species***: Colony is applicable to both dioecious and monoecious species for sibship and parentage assignments. In both cases, candidate parents (CMS, CFS) are allowed to be present or absent. For monoecious, CMS and CFS must be the same if present. For dioecious, the species is allowed to be either diploid or haplodiploid. In the diploid case (dioecious or monoecious), all individuals are assumed diploid. In the haplodiploid case, males and females are assumed to be haploid and diploid respectively (for species with diploid males and haploid females, you just need to swap the two sexes), offspring in OFS can be diploid, haploid or a mixture of both. For monoecious, the species is always assumed to be diploid. For polyploid species, the codominant marker genotype data can be transformed to persudo diploid dominant marker data before carrying out Colony analysis, as described by Wang & Scribner (2014).

(4) ***Length of run***: Longer runs consider more configurations in the simulated annealing algorithm to search for the best assignment with the maximum likelihood, and thus are more likely to find the maximum likelihood configuration, but take more time to do so. Four run length options are provided, Short/Medium/Long/VeryLong, increasing the running time by roughly 10 times (e.g. Long run takes about 100 times of the time for a short run). In most cases, a medium run is a good compromise between running time and accuracy. See FAQ 14.1 and 14.5 for more details.

(5) ***Analysis method****:* Four methods (full likelihood, FL; pairwise-likelihood score, PLS; FL and PLS combined, FPLS; pure pairwise likelihood, PPL) for sibship and parentage assignments are implemented in Colony2. The FL method is the most accurate one as verified by simulated and empirical data analyses (Wang 2012). The PLS method uses the same simulated annealing process as employed by FL to search for the best configuration. It, however, calculates and uses the sum of pairwise log-likelihoods instead of the full log-likelihood as the criteria to assess how plausible a configuration is. FPLS is similar to FL, except configurations are first screened by PLS to speed up the computation. When a new configuration is constructed, its PLS is calculated and compared with that of an old configuration. If the new configuration is abandoned based on the changes in PLS according to the Metropolis-Hastings algorithm, then there is no need to calculate the FL of the new configuration. FPLS works well, being similar in accuracy to but faster than FL, when sibship sizes are not very big and marker information is ample. Otherwise, it is slightly less accurate than FL, but still more accurate than PLS. The PPL method calculates the likelihoods of a pair of individuals under different candidate relationships independently of other individuals, as described in Wang & Santure (2009).

In general, FL is the most accurate, followed by FPLS and PLS, while PPL is the least accurate. However, PPL is the computationally fastest method, while FL is the slowest. For medium to large

datasets containing many markers (say, hundreds) and many individuals (say, thousands), a good compromise is to use the PLS method. In Colony2, the analysis using PPL is always conducted and presented. However, the user is asked to choose between FL, PLS and FPLS. The default method is FL.

(6) ***Likelihood precision****:* The option is valid only when FL or FPLS is chosen as the analysis method, and both male and females are designated as polygamous. As hinted before, the FL or (to a less extent) FPLS method can be very slow for a large dataset involving many offspring when genotyping errors are present and when both sexes are polygamous. Reducing the precision of the likelihood computation can reduce the running time, with a slight negative effect on assignment accuracy.

(7) ***Update allele frequency****:* Allele frequencies are required in calculating the likelihood of a configuration. These frequencies can be provided by the user (see below) or are calculated by Colony using the genotypes in OFS, CMS (optional) and CFS (optional). In the latter case, you can ask Colony to update allele frequency estimates by taking into account of the inferred sibship and parentage relationships during the process of searching for the maximum likelihood configuration. However, updating allele frequencies could increase computational time substantially, and may not improve relationship inference much if the genetic structure of your sample is not strong (i.e. family sizes small and evenly distributed, most candidates are not assigned parentage). I suggest not updating allele frequencies except when family sizes (unknown) are suspected to be large (relative to sample size) and highly variable.

(8) ***Sibship size scaling***: When a full sibship becomes large, containing say hundreds siblings, it might be falsely reconstructed into 2 or more full sibships by the full likelihood method, if marker information is not high enough (Wang 2013). To avoid such errors, full sibship size is scaled down based on the number of alleles and the genotyping error rate at a locus and the actual full sibship size (Wang 2013). Analyses of numerious simulated datasets and quite a number of empirical datasets verify that this scaling scheme can highly effectively reduce large sibship splitting without causing the merging of small sibships (i.e. the scheme keeps both false full-sibling rate and false non-full-sibling rate low). Therefore, the default option for Sibship Scaling is Yes.

Very accationally, however, the scaling scheme causes an excessive merging of small full sibships to produce a falsely large full sibship. Such a falsely large full sibship is characterized by an exceptionally high proportion of loci showing genotypes incompatable with a full sibship, such as displaying >4 alleles and >2 types of homozygotes at a codominant locus in diploid species. As far as I am aware, there is only one report of this problem. In case of such a problem, one could re-set up the project, taking the alternative option for Sibship Scaling, NO, and re-run the dataset.

In situations where you know the maximal full sibship size is small (say, < 20), then a single run with the alternative option NO is needed. In situations where you have no idea of the possible full sibship size range or you suspect some full sibship can be large, my suggestion is to take the default option YES to run your dataset. If the results indicate an excessive merging of small sibships because a reconstructed full sibship is too large or it contains too many incompatable genotypes, then a second run with the alternative Sibship Scaling option, NO, is required. Otherwise (I would assume this is the norm), no need for a 2nd run.

(9) ***Number of runs***. For the same dataset and parameters of a project, multiple runs can be conducted so that the best configuration with the maximum likelihood is more likely to be found and the

uncertainties of the estimates (see below) are more reliably assessed. However, it is very time consuming to do multiple runs. Furthermore, in typical situations a single run suffices for a point estimate.

(10) ***Random number seed***. Colony uses the simulated annealing algorithm to search for the ML configuration. It is a Monte Carlo method similar to MCMC, with a fine control of re-configuration acceptance rate though annealing "temperature". Starting from the initial configuration in which all individuals are set as unrelated except for those individuals with known relationships, a random change is made to part of the configuration to generate a new configuration. The likelihoods of the new and old configurations are then calculated and compared to determine whether the new one is to be accepted or rejected. If the new likelihood is larger than the old one, then the new configuration is accepted. Otherwise, an acceptance rate is calculated using the current temperature, the new and old likelihood values, and is compared with a random number drawn from a uniform distribution in the range of [0,1]. If the random number value is smaller than the acceptance rate, the new configuration is still accepted although it is inferior to the old one. This is intended to avoid the algorithm getting stuck on a local maximum in the likelihood surface. Therefore, the random number seed partially determines the searching path. With exactly the same data and parameter values, different runs using different random number seeds may give slightly different final best configurations and likelihood values. Such a case occurs occasionally when there is not enough information in the maker data to resolve the genetic structure, the actual genetic structure of the sample is extremely weak, or the sample size is very large (i.e. thousands of individuals). For example, when the number of markers is small, and/or the markers are not informative (few alleles with uneven frequency distribution), and/or most families are extremely small (e.g. one offspring per sibship), it is difficult to have replicate runs (using different random number seeds) converge to the same best configuration. One can do multiple runs for the same dataset by using different random number seeds to check/confirm the reliability of the analysis results. In the case replicate runs yield different results, the good news is that relationships reliably inferred are usually reconstructed consistently among runs, while dubious relationships are inferred inconsistently among the runs. One just needs to focus on those reliable, consistent relationships and ignore (abandon) those unreliable, inconsistent relationships in downstream analyses.

(11) ***Sibship size prior***. You can choose to use or not to use a prior distribution for the paternal and maternal sibship of the offspring.

In the case of both sexes polygamous where both paternal and maternal half sibship is to be inferred, some unrelated or loosely related individuals (such as cousins) may be inferred as half or full siblings because they have similar genotypes if the markers are not highly informative. Indeed, loosely related (e.g. cousins) or even unrelated individuals can have identical genotypes at a probability increasing with a decreasing amount of marker information (fewer markers, less polymorphisms, higher mistyping rates). The larger the sample size is and the less marker information the dataset has, the more severe is the problem. Sometimes all offspring in OFS are inferred to be linked in a two-generation pedigree either directly (by sharing the same parent or parent pair) or indirectly. This problem not only reduces inference accuracy, but also increases computational time dramatically. Figure 2A depicts the shape of a typical falsely inferred large pedigree in which all offspring in a sample can sit comfortably.

On the other hand, any information about the average paternal and maternal sibship sizes can be used in a prior to help in sibship and parentage assignments.

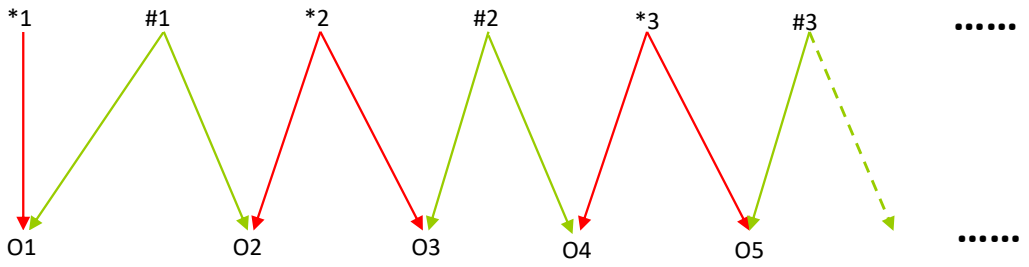


Figure 2A: A large loose pedigree. Inferred male and female parents (not included in candidate parent samples) are numbered with a prefix * and #, respectively. Offspring (included in OFS) are numbered with a prefix O.

Because of the above two considerations, I use Ewen's sampling formula as the prior for paternal and maternal sibship size distributions. Simulations showed that the prior can discourage loose and unnecessarily complex pedigrees to reduce false sibships and computational time, and can increase inference accuracy.

Suppose paternal sibship size distribution is **m**={m_1, m_2, …, m_n}, where m_i (i=1, ..., n) is the number of paternal sibships each consisting of exactly i offspring. The total number of offspring is $n = \sum_{i=1}^{n} i m_i$, and the average number of non-empty paternal sibships (= the number of contributing fathers) is $k = \sum_{i=0}^{n-1} \frac{\alpha}{i+\alpha}$, where α is a concentration parameter that determines the degree to which individuals are allocated to the same father. We can substitute k by n / n_p and solve numerically for α, where n_p is the average number of offspring per father in the sample. Given α, the prior probability of **m**={m_1, m_2, …, m_n} is $\text{Prb}(\mathbf{m}) = \frac{n!}{\prod_{i=0}^{n-1}(\alpha+i)} \prod_{i=1}^{n} \left(\frac{\alpha}{i}\right)^{m_i} \frac{1}{m_i!}$. The prior distribution of maternal sibship size is similarly defined.

The strength of the prior is accommodated by using $(\text{Prb}(\mathbf{m}))^x$ in the calculation, where x takes values of 0 (no prior), 0.25 (weak prior), 0.5 (medium prior) and 1.0 (strong prior). To use the prior, you are asked to provide the (estimated) average paternal (n_p) and maternal (n_m) sibship size, and the value of x. When one has no idea of the average sibship size, use x=0 (No prior) and you are not asked about the n_p and n_m values. Otherwise, use x=0.25, 0.5 and 1.0 when your confidence in the provided n_p and n_m values is low, medium and high respectively. The default setting is n_p=1 for paternal and n_m=1 for maternal sibships, and x=0.25. This default setting is designed to reduce false sibship assignments, and to reduce computational time.

A fifth option of the prior, "Known Ne", is also provided for the cases (1) a sample of individuals is taken at random from a population with a known effective size, N_e, and a roughly known sex ratio, R and (2) a random sample of individuals is used to estimate the N_e of the sampled population from sibship frequencies (Wang 2016). In case (2), the prior N_e and R may be assumed values or estimates from likeage disequilibrium or temporal changes in allele frequency. In both cases, with the provided prior N_e and R values, Colony will calculate and use an optimal sibship prior in making relationship inference and N_e estimation.

Note that the prior settings are related to other parameters. As a result, there are intuitively mutual constraints on the prior and other parameters. (1) When both sexes are monogamous, we have $n_p = n_m$. (2) When one sex is polygamous and the other sex is monogamous, then the average sibship size for the polygamous sex must be no smaller than that for the monogamous sex. (3) Both n_p and n_m must not be larger than the offspring sample size.

To minimize falsely large pedigrees like that shown in Figure 2A, n_p and n_m are better taken as the HARMONIC mean number of offspring per father and mather respectively. If most paternal sibships are small and only a few is large, then n_p should be small, closer to the smallest rather than the arithmetic mean paternal sibship size.

(12) ***Note to the project***. You can put anything in the text box, such as when you set up the project, notes to the dataset, etc. The GUI will append to your note with some basic information such as the date and time the new project is created.

3.3 Markers

In page 2 (see Figure 3), the information about the markers should be provided.

(1) *Number of loci genotyped*. Provide the maximum number of marker loci genotyped for the individuals in your sample. Note that marker loci have an implicit order which should be followed consistently in the entire input. For example, in offspring, candidate male and female genotype data, in allele frequency data and in marker type and genotyping error data, the same order of marker loci must be followed. In all these files, "the first locus" or "locus 1", for example, must refer to the same marker.

(2) *Marker type and error rate.* Click the load button under "Marker Types and Error Rates" to load the file. In the file, 4 values should be provided for each marker (in columns). The first value (on row 1) specifies the marker name or ID (consisting of a maximum of 20 letters/numbers, others such as space, comma, full stop, forward and backward slashes are not allowed in the name/ID). The second (on row 2) indicates the marker type, whether it is codominant (0) or dominant (1). The third and fourth values (on rows 3 and 4 respectively) give the allelic dropout rate and the rate of other kinds of genotyping errors (including mutations) of the marker. For more information about the models of genotyping errors, see Wang (2004).

Note that this file sets the order of marker loci that must be followed in all the following input. The first column is for locus 1, the second for locus 2, …

Note also that computation for the FL method becomes slow when markers suffer from genotyping errors. This is especially obvious when both males and females are specified as polygamous. The higher the error rate, the slower the program will run.

An example file of the "Marker Types and Error Rates" with 5 loci is shown below, and when loaded into Colony it looks like Figure 3 (lower pane). Note the column heads in Figure 3 (i.e. "Locus-1") should not be included in the file. The column heads are added by Colony automatically when loading the file. This is true for all of the following files loaded into Colony.

```
mk1     mk2     mk3     mk4     mk5
  0       0       0       0       0
```

```
0.0000  0.0000  0.0000  0.0000  0.0000
0.0001  0.0001  0.0001  0.0001  0.0001
```

The input can be simplified tremendously when all loci have the same generic marker name/ID (1st line) or when all loci have the same value (2-4 lines). Only one input item is needed per line in such a case. For example, the above example input is simplified to one item per line:

```
mk@
0@
0.0000@
0.0001@
```

The symbol @ indicates that the same value applies to all loci (lines 2-4), or the same generic name applies to all markers (line 1) and Colony will add the order of the marker to its generic name. The full and simplied input can be mixed for different lines. For example, the 2nd and 3rd lines can be

```
0    0    0    1    0
0.0000@
```

(3) *Allele frequency.* If population allele frequencies are unknown and are to be estimated from the current dataset within which relationships are being inferred, click the "unknown" radiobutton and Colony will estimate allele frequencies from the current samples. If population allele frequencies are known or have been estimated from another larger, more appropriate sample, click the "known" radiobutton and then the "load" button to load the allele frequency file. The file should be prepared, before loading, in the following format.

Each locus takes 2 consecutive rows. The first row lists the allele names/identifications (using a unique integer number, 1~999999999), and the second row lists the corresponding frequencies of the alleles. Alleles (or allele frequencies) on the same row should be separated by a comma or white space. The first two rows are for locus 1, the 3rd and 4th rows are for locus 2, …. Within a locus, allele names/identifications must be unique but are not necessarily ordered or sequential. Alleles at different loci are allowed to have the same identification number. The same allele names/identifications of a locus must be used in the genotype data for offspring and candidate parents.

An example file of the allele frequency data is shown below, and when loaded into Colony it looks like Figure 3 (upper panel).

```
1     2     3     4     5     6     7     8     9     10    11    12
0.100 0.060 0.035 0.070 0.070 0.140 0.150 0.055 0.055 0.025 0.135 0.105
0     0     0     0     0     0     0     0     0     0     0     0
1     2     3     4     5     6     7     8     9     10    11    12    13
0.055 0.100 0.095 0.040 0.030 0.055 0.060 0.095 0.105 0.080 0.125 0.090 0.070
0     0     0     0     0     0     0     0     0     0     0     0     0
1     2     3     4     5     6     7     8     9     10    11    12    13    14
0.035 0.035 0.115 0.090 0.055 0.110 0.080 0.020 0.040 0.115 0.055 0.100 0.110 0.040
0     0     0     0     0     0     0     0     0     0     0     0     0     0
1     2     3     4     5     6     7     8     9     10    11    12    13    14    15
0.080 0.065 0.060 0.030 0.125 0.080 0.050 0.060 0.090 0.065 0.125 0.070 0.050 0.020 0.030
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
1     2     3     4     5     6     7     8     9     10    11    12    13    14    15    16
0.090 0.150 0.020 0.050 0.055 0.075 0.085 0.065 0.060 0.100 0.040 0.055 0.030 0.040 0.020 0.065
0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

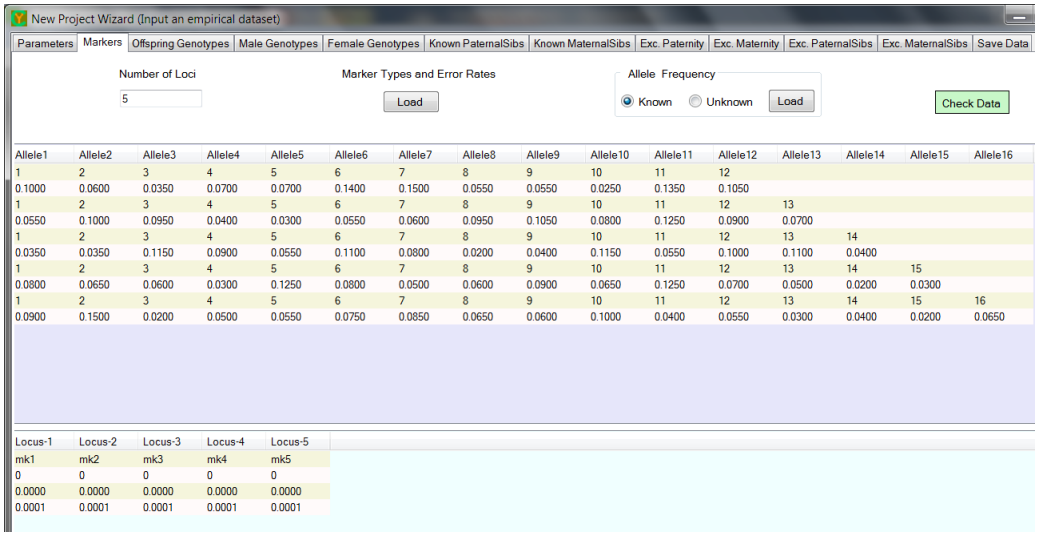


Figure 3: Page for inputting marker information

Note that when allele frequencies are specified as known, the allele frequency file loaded should contain all alleles found in the offspring and candidate genotypes. Otherwise, an error occurs in running Colony. Note also that for a dominant locus, only two alleles are allowed and they are always indexed as 1 to indicate the dominant allele (presence of a band) and 2 to indicate the recessive allele (absence of a band when homozygous). When population allele frequencies are misspecified, sibship and parentage might be overestimated and all offsping might be inferred to have the same (or very few) parents (see FAQ).

After inputting all required information, you need to click the "Check Data" button to check the validity of the input in the current page and the compatibility with previous pages. Only when this button is clicked and the check passed, are you allowed to go to the next page for input. The function of the check buttons in the following pages is similar.

3.4 Offspring genotypes

Information about offspring genotypes is required in Page 3.

(1) *Number of offspring*. Provide in the text box the number of offspring with genotypes included in the OFS sample. The minimum value is 1.

(2) *Offspring genotypes*. Load the offspring genotype file into the project by clicking the "Load Genotype" button. The file should contain the individual IDs and the genotypes at each locus (Figure 4). Each individual takes a single row. The first column gives the individual ID (a string containing a maximum of 20 letters and/or numbers, no other characters are allowed), the 2nd and 3rd columns give the alleles observed for the individual at the first locus, the 4th and 5th give the alleles observed for the individual at the 2nd locus, … An allele is identified by an integer, in the range of 1~999999999. If the locus is a dominant marker, then only one (instead of 2) column is required for the marker, and the value for the genotype should be either 1 (dominant phenotype, presence of a band) or 2 (recessive phenotype, absence of a band). Missing genotypes are indicated by 0 0 for a codominant marker and 0 for a dominant marker. Note that offspring IDs should be unique. They are case sensitive, which means that, for example, "offspring2" and "Offspring2" are treated as different. An offspring with missing genotypes at all loci (no marker information at all)

should not be included in the offspring genotype file. It is also recommended to exclude an individual who has little marker information (i.e. few loci with non-missing genotypes).

Information for a haploid offspring is the same as that for a diploid offspring as detailed above, except that the 2nd allele at each locus should be a fixed number of -99. The program reads in the genotype data of an offspring, and determines the ploidy of the offspring by examining the 2nd allele at each locus. If the 2nd allele is -99 at each codominant locus that has genotype data (i.e. the 1st allele is a positive number), then the offspring is deemed as a haploid. If the 2nd allele is a positive number at each codominant locus that has genotype data, then the offspring is deemed as a diploid. If the 2nd allele is -99 for some codominant loci and a positive number at other codominant loci that has genotype data, then the offspring cannot be determined for ploidy, and the program stops with an error message.

Part of an example offspring genotype file is shown below, and when loaded into Colony it looks like Figure 4. Note the column heads (e.g. Offspring) should not be included in the offspring genotype file. They are added by Colony automatically when loaded.

```
O1    11    11     8    11    11     3     9     2     1     2
O2    11    11     8     3    11     3     9    12     1     1
O3    11     1     8     3    11     4     9    11     1     8
O4    11     9     8     2    11     4     9    11     1     8
O5    12    12     3     1     4    12     2     3     2    16
O6    11    12     3     1     3     4     2     3     2    16
```

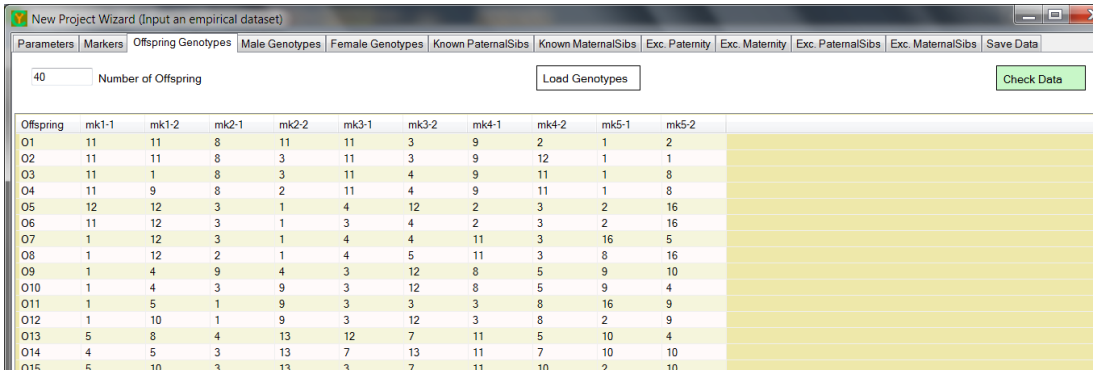


Figure 4: Page for offspring genotypes

The data (at 5 codominant loci) for a haploid offspring can be listed as

```
Ox   11  -99    3  -99    3  -99    2  -99    2  -99
```

In the special case that all loci are dominant and the species is haplo-diploid, an offspring genotype is displayed as a single number for each locus. In such a case, the offspring ploidy cannot be determined as described above, and has to be speficied explicitly. An offspring takes one row, with the 1st column showing the offspring ID (a string), the 2nd column showing the offspring ploidy (1 and 2 for haploid and diploid, respectively), the 3rd column showing the genotype (1 or 2 for dominant or recessive) at locus 1, …

Example data for two offspring at 5 dominant marker loci are

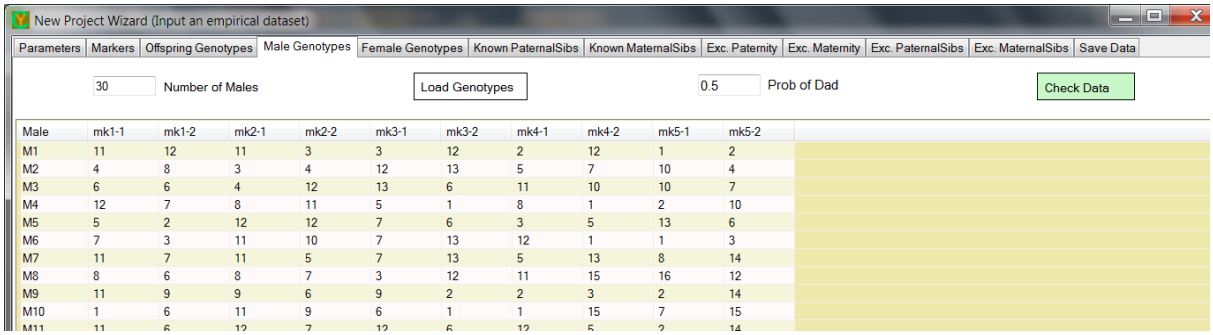
```
Ox   1  2  2  1  1  1
Oy   2  1  1  1  1  1
```

Offspring Ox and Oy are haploid and diploid (shown in red in column 2), respectively.

3.5 Candidate male genotypes

Page 4 reads in information about candidate males.

(1) *Number of candidate males*. Provide in the text box the number of candidate males included in the CMS sample. Note that known fathers are also included in the CMS sample. The minimum value is 0, in which case paternity is not inferred.



Male	mk1-1	mk1-2	mk2-1	mk2-2	mk3-1	mk3-2	mk4-1	mk4-2	mk5-1	mk5-2
M1	11	12	11	3	3	12	2	12	1	2
M2	4	8	3	4	12	13	5	7	10	4
M3	6	6	4	12	13	6	11	10	10	7
M4	12	7	8	11	5	1	8	1	2	10
M5	5	2	12	12	7	6	3	5	13	6
M6	7	3	11	10	7	13	12	1	1	3
M7	11	7	11	5	7	13	5	13	8	14
M8	8	6	8	7	3	12	11	15	16	12
M9	11	9	9	6	9	2	2	3	2	14
M10	1	6	11	9	6	1	1	15	7	15
M11	11	6	12	7	12	6	12	5	2	14

Figure 5: Page for candidate male genotypes

(2) *Male genotypes (optional).* When the number of candidate males is larger than 0, the user is asked to load a file containing the candidate male genotypes (Fig. 5). The format of male genotype file is the same as the offspring genotype file, except for 2 cases. One is that, for haplodiploid species (in which males are assumed to be haploid), each locus takes just one column rather than 2 columns, no matter the marker is codominant or dominant. The other is that, if an individual already included in the offspring genotype file is present in the candidate male file, it is not necessary to provide its genotypes, just the first column for its individual ID will do.

(3) *Probability of an actual father being included in candidates (optional).* Provide a guess (estimate) of the probability that the actual father of an offspring in the OFS is included in the CMS sample.

3.6 Candidate female genotypes

Page 5 reads in similar information about candidate females.

(1) *Number of candidate females*. Provide in the text box the number of candidate females included in the CFS sample. The minimum value is 0, in which case maternity is not inferred.

(2) *Female genotypes (optional).* When the number of candidate females is larger than 0, the user is asked to load a file containing the candidate female genotypes. The format of female genotype file is the same as the offspring genotype file, except that if an individual already included in the offspring genotype file is present in the candidate female file, it is not necessary to provide its genotypes, just the first column for its individual ID will do.

(3) *Probability of an actual mother being included in candidates (optional).* Provide a guess (estimate) of the probability that the actual mother of an offspring in the OFS is included in the CFS sample.

3.7 Known paternal sibship/paternity

In page 6, you can input information about any known paternal relationships to help infer unknown relationships.

(1) *Number of known paternal sibship/paternity*. Provide in the text box the number of known paternal sibships with or without known fathers included in the samples. The minimum value is 0. A known paternal sibship contains 1 or more offspring in the OFS sample who are known to share the same father no matter whether the father is known and included in CMS or not. For example, in the example shown in figure 6, there are 4 known paternal sibships. The 1st sibship contains 2 offspring, O1 and O2, who share a known and included father M1. The 3rd sibship contains 2 offspring, O23 and O25, whose father is unknown (i.e. not included in CMS) and thus indicated by "0" in the FatherID field.
(2) *Mismatch threshold (optional)*. If the number of known paternal sibship/paternity is larger than zero, you are asked to give the mismatch threshod, which should be an integer in the range [0, #Loci]. It is used to determine whether a known father-offspring dyad is accepted or not. If the pair of multilocus genotypes of a known father-offspring dyad show mismatches (Mendelian incompatibility) larger than the shreshold, then this presumed known father-offspring relationship is rejected. Otherwise, it is accepted and will not be inferred from genotype data.
(3) *Known paternal sibship/paternity (optional)*. If the number of known paternal sibship/paternity is larger than zero, then you are asked to load a file for the known paternal sibship/paternity. In the file, each known paternal sibship/paternity takes a row, with the first column containing the father ID/name if the male is known and included in CMS, or a value of 0 to indicate that the father is unknown or not included in CMS. From the 2nd column on, the ID/name of each member of the paternal sibship is listed.

An example is shown in Figure 6. Note again column heads should be excluded from the original file. In this example, the 1st row signifies that offspring O1 and O2 share the same father, M1, who is included in the CMS. The 3rd row signifies that offspring O23 and O25 are known to share the same father, who is unknown (indicated by 0). Note that 2 (or more) paternal sibships with different known fathers are never merged into a single paternal sibship; they are always kept distinctive in constructing relationship configurations. Offspring sharing the same unknown father are allowed to merge with a paternal sibship with known or unknown father. Note also that offspring within a single known paternal sibship with a known or unknown father will never be split into different paternal sibships.

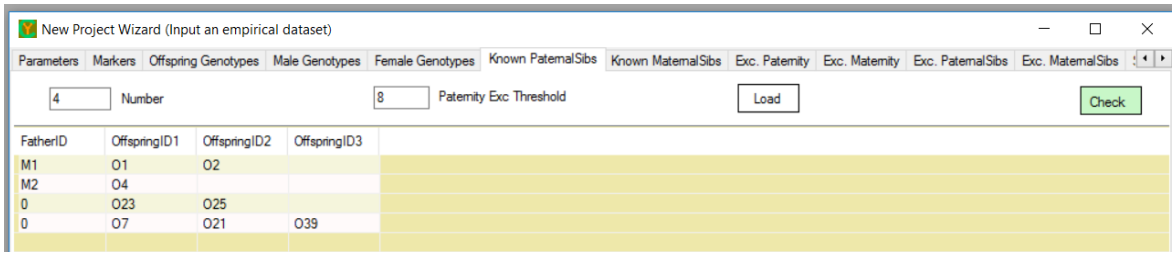


Figure 6: Page for known paternal sibship/paternity

3.8 Known maternal sibship/maternity

In page 7, you can input similar information about any known maternal relationships to help infer unknown relationships.

(1) *Number of known maternal sibship/maternity*. Provide in the text box the number of known maternal sibship or maternity included in the samples. The minimum value is 0.
(2) *Mismatch threshold (optional)*. The same as that defined in 3.7.
(3) *Known maternal sibship/maternity (optional)*. If the number of known maternal sibship/maternity is larger than zero, then you are asked to load a file for the known maternal sibship/maternity. In the file, each known maternal sibship/maternity takes a row, with the first column containing the mother ID/name if the female

is known and included in CFS or a value of 0 to indicate that the mother is unknown or not included in CFS. From the 2nd column on, the ID/name of each member of the sibship is listed.

3.9 Excluded paternity

In some cases, we know from the age or other information that some candidate males are definitely impossible to be the father of a specific offspring. Such information can be used as input to help infer parentage more accurately. In page 8, you are allowed to input, for each offspring, the candidate males that are excluded as the father.

(1) *Numbers of offspring with any excluded paternity.* Provide in the text box the number of offspring that each has at least one known excluded candidate male as its father. The minimum value is 0 (Fig. 7).
(2) *Excluded paternity (optional).* If the number of offspring with excluded paternity is larger than zero, then you are asked to load a file for the excluded candidate males. Each offspring with excluded paternity takes one row. The first entry of the row is the offspring ID/name, followed by the IDs of the candidate males that are excluded parentage of the offspring.

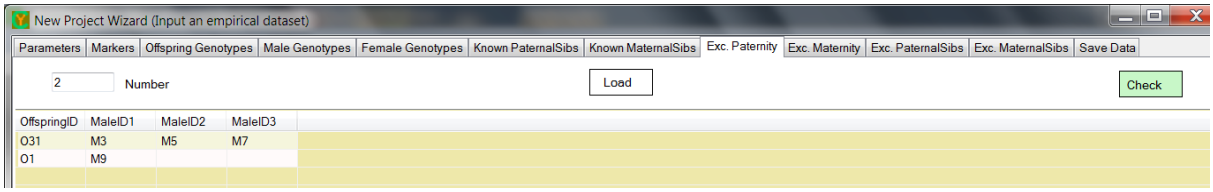


Figure 7: Page for excluded paternity

3.10 Excluded maternity

Similar to excluded paternity (see 3.9 above), known excluded maternity can also be inputted as information in the analysis in page 9.

3.11 Excluded paternal sibships

In some cases, we know that an offspring cannot possibly share the same father with one or more other offspring in the sample. Such information can be used as input to help infer sibships and parentage more accurately. In page 10, you are allowed to input, for each offspring, the offspring that are excluded as the paternal siblings. An offspring that has no excluded individuals as siblings should not be listed.

(1) *Numbers of offspring with any excluded paternal siblings.* Provide in the text box the number of offspring that each has at least one known excluded individual as its paternal sibling. The minimum value is 0 (Fig. 8).
(2) *Excluded paternal siblings (optional).* If the number of offspring with any excluded paternal siblings is larger than zero, then you are asked to load a file for the excluded paternal siblings. Each offspring with one or more excluded paternal siblings takes one row. The first entry of the row is the offspring ID/name, followed by the IDs of the offsprings that are excluded paternal siblings. In the example shown in Figure 8, O5, O3 and O4 are known to not share the same father with O1. However, it does not say anything about the relationship among O3, O4 and O5. They may and may not share the same father.

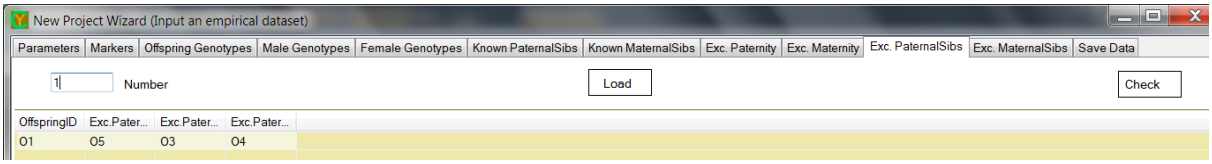


Figure 8: Page for excluded paternal sibship

3.12 Excluded maternal sibships

Similar to excluded paternal sibships (see 3.11 above), known excluded maternal sibships can also be inputted as information in the analysis in page 11.

4. Empirical data input – NO Windows GUI

If one chooses not to use the Windows GUI, and wants to run the program in MS-DOS mode on PCs or run the program in other platforms such as Mac and Linux, one just needs to prepare an input file called "Colony2.dat" which includes all of the parameter values and contains all of the data required by Colony program. **An input file with a name different from "Colony2.dat" is also acceptable, but must be defined in the command line invoking the Colony program (see section 8 below)**. The DOS mode is preferred to Windows GUI mode in the cases of a simulation study, and of analyzing a large dataset involving many loci or/and individuals.

For PC users, it is also possible to prepare and input the data using the GUI when the dataset is not very large and then run colony in DOS mode. This hybrid procedure combines the convenience of the GUI for data input and the running speed of the DOS mode. Once data input is complete, an input file called "Colony2.dat" is generated in the colony project folder. To run the data in DOS mode, copy Colony2.dat to the folder where the Colony program is installed (where Colony2p.exe is found). Then open and edit Colony2.dat file in the Colony program folder using a text editor, such as Notepad.exe. Change the indicator variable for monitor method on the line remarked by "`! 0/1=Monitor method by Iterate#/Time in second`", to 0. Change the value on the next line remarked by "`! Monitor interval in Iterate# / in seconds`", to 10000 or 100000. After saving the edited file, double click Colony2p.exe will start the run. More preferably, open a new DOS windows using Windows "command", and use the DOS command "cd" to navigate to the Colony program directory (folder). Type Colony2p.exe followed by return will start the run. Once finished, the results should be found in your colony project folder (where Colony2.dat was originally found), and you can use the GUI to view the results.

In the following, I specify how to prepare the Colony2.dat file manually (from scratch) using a text editor.

4.1 Data input

For no-GUI mode, the same data as for the Windows GUI mode described in section 3 are required. However, the difference is that all the data must be put in a single pure text input file in the following order and format. Also the delimiters in all data files must be either white spaces or commas. Empty rows take no effect. More details about the meaning and format requirement of each input item are explained in section 3. Here in this section, I only give details on those parts that are different in format (e.g. YES/NO in GUI, but 1/0 in no-GUI) between the GUI and no-GUI modes.

(1) *Project name (String)*. On the 1st line of the input file, give a project name, which should be a string containing less than 40 letters and numbers.

(2) *Output file name (String).* On the 2nd line of the input file, give an output file name which should be a string containing less than 40 letters and numbers. All output files of Colony will use the same output file name but different extension names.

(3) *Number of offspring in the OFS sample (Integer).* Provide this number on the 3rd line.

(4) *Number of loci (Integer).* Provide this number on the 4th line.

(5) *Seed for random number generator (Integer).* Provide this number on the 5th line.

(6) *Updating allele frequency (Boolean).* Provide on the 6th line an indicator of value 1 (or 0) to instruct Colony to (or not to) update allele frequencies during the simulated annealing process in searching for the ML configuration.

(7) *Dioecious/Monoecious (Integer).* On line 7 give a value of either 2 or 1 to indicate dioecious species or monoecious species.

(8) *Outbreeding/Inbreeding (Boolean).* On line 8 give a value of either 0 or 1 to indicate the absence or presence of inbreeding.

(9) *Ploidy (Boolean).* On line 9 give a value of either 0 or 1 to indicate diploid species or haplodiploid species. For monoecious, the indicator value should always be 0 to indicate diploid.

(10) *Mating systems for males and females (Boolean, Boolean).* On line 10, give 2 indicator values (0 or 1) to specify whether males and females are polygamous (=0) or monogamous (=1). The 1st value is for males and the 2nd is for females.

(11) *Clone inference (Boolean).* On line 11, give an indicator value (1 or 0) to specify whether clones (or duplicated individuals) are to be inferred (=1) or not (=0).

(12) *Sibship size scaling (Boolean).* On line 12, give an indicator value (1 or 0) to specify whether full sibship size is to be scaled (=1) or not (=0).

(13) *Sibship prior indicator (Integer), average paternal sibship size (Real, optional), average maternal sibship size (Real, optional).* On line 13 give an indicator value of 0, 1, 2 or 3 to indicate no sibship prior, weak sibship prior, medium sibship prior or strong sibship prior. The average paternal and maternal sibship sizes are optional when the prior indicator is 0. Otherwise, they are necessary.

The "*Sibship prior indicator"* can also take a value of 4 to indicate that an optimal sibship prior will be determined from the known prior values of effective population size, Ne, and sex ratio, R. In such a case, the 2nd and 3rd values are prior Ne and R values, in place of "*average paternal sibship size*" and "*average maternal sibship size*".

(14) *Population allele frequency indicator (Boolean).* On line 13 give an indicator value of either 1 or 0 to indicate whether population allele frequencies for each locus are known and are to be provided or not.

(15) *Numbers of alleles per locus (Integer, optional)*: These are required only when the *Population allele frequency* indicator is set to 1. List the numbers of observed alleles at each locus on a line,

starting from locus 1 in ascending order. Note that all alleles observed in the current samples being analysed for relationship must be included. When the number of loci is large (say, 100000 SNPs), the numbers of observed alleles at each locus are allowed to be listed in multiple lines. In such a case, no line note ("!" and anything after it) is allowed.

(16) *Alleles and their frequencies per locus (Integer, Real, optional)*: These are required only when the *Population allele frequency* indicator is set as 1. Each allele at a locus is represented by a unique identification number (1~999999999). These numbers are not necessarily ordered or sequential. Alleles at different loci are allowed to have the same identification number. For a given locus, list the observed alleles (i.e. their identification numbers) on one line and their corresponding frequencies on the next line. All genotyped loci are listed, starting from locus 1 in ascending order. Note that all alleles observed in the current sample being analysed for relationship must have frequencies greater than zero. Similar to (14), the observed alleles (or allele frequencies) at a single locus are allowed to be listed in multiple lines when the number of alleles is large.

(17) *Number of runs (Integer).* Provide a value for the number of replicate runs for the dataset (project).

(18) *Length of Run (Integer)*. Give a value of 1, 2, 3, 4 to indicate short, medium, long, very long run.

(19) *Monitor method (Boolean).* Give a value of either 0 or 1 to indicate monitoring the intermediate results by iterate number or running time. Always choose value 0 for run without Windows GUI.

(20) *Monitor interval (Integer).* Give a number to specify the interval by which intermediate results are directed to the monitor. The unit of the interval is number of iterates or seconds in running time when the Monitor method is set to 0 or 1, respectively. The suggested value is 10000 and 100000 for run with the full likelihood method (indicator value =1 or 2 in Analysis method below) and PLS method (indicator value =0 in Analysis method below).

(21) *Windows GUI (Boolean)*. Give an indicator number of 1 or 0 to indicate running Colony in the Windows GUI mode or not. Always set the value to 0 when running Colony in no-GUI mode.

(22) *Analysis method (Integer).* Give an indicated value of 0, 1 or 2 to tell Colony to use the pairwise-likelihood score (PLS), full likelihood (FL), or the FL and PLS combined (FPLS) method. More on these methods are explained above in the Windows GUI data input section.

(23) *Precision (Integer).* Give an indicated value of 0/1/2/3 to use Low/Medium/High/VeryHigh precision in calculating the full likelihood.

(24) *Marker IDs/Names (String)*. There are two options as described in "***3.3 Markers"***. With option 1, you need to provide the marker ID/name of each locus, starting from locus one in ascending order. With option 2, you just need to provide a generic marker name followed (without separator in between) by the symbol @. Colony will automatically name markers with the generic name and their order. For example, if the line input is MK@, then Colony will automatically name the markers as MK1, MK2, MK3, …

(25) *Marker types (Boolean).* Corresponding to each marker in the previous row, provide the type of each marker on a single line. Use indicator values of 1 and 0 to indicate dominant and codominant markers, respectively. If all markers are of the same type, then a single input item 1@ or 0@ (for dominant and codominant respectively) is acceptable, as described in "***3.3 Markers"***.

(26) *Allelic dropout rates (Real).* List the allelic dropout rates at each locus on a line, starting from locus 1 in ascending order. If all markers have the same rate, then a single input item x@ (where x is the dropout rate, such as 0.01) is acceptable, as described in "***3.3 Markers"***.

(27) *Other typing error rates (Real).* List the other typing error rates at each locus on a line, starting from locus 1 in ascending order. If all markers have the same rate, then a single input item x@ (where x is the *other typing error* rate, such as 0.01) is acceptable, as described in "***3.3 Markers"***.

(28) *Offspring IDs and genotype (String, Integer).* For each individual offspring in the OFS sample, list its ID or name followed by the two alleles for a codominant or the phenotype for a dominant marker locus. The format is the same as that in the Windows GUI. When the number of loci is large, the genotypes of an individual are allowed to be listed on multiple lines without line note and without line continuation mark "&".

(29) *Probabilities that the father and mother of an offspring are included in the candidate males and females (Real)*. Provide, on a single line, the estimates of the probabilities that a father and a mother of an offspring are included in the CMS and CFS samples, respectively. The two numbers must be provided even if there are no candidate males or/and females.

(30) *Numbers of candidate males and females (Integer)*. Provide on a single line the numbers of candidate males and females included in the CMS and CFS samples, respectively.

(31) *Candidate male IDs/names and genotypes (String, Integer, optional)*: If there are candidate males, give the ID/name and alleles at each locus of each male on a single line, the format being the same as specified in the Windows GUI section above.

(32) *Candidate female IDs/names and genotypes (String, Integer, optional)*: If there are candidate females, give the ID/name and alleles at each locus of each female on a single line, the format being the same as specified in the Windows GUI section above.

(33) *Number of known paternity and exclusion threshold (Integer, Integer)*. Provide the number of offspring in the OFS sample that have known fathers included in the CMS sample, and the maximum #mismatches allowed for a known father-offspring dyad. Both have minimum values 0.

(34) *Known offspring-father dyad (String, optional).* If the number of known paternity is larger than zero, then on each row list a single dyad containing the offspring ID followed by its known father ID.

(35) *Number of known maternity and exclusion threshold (Integer, Integer)*. Similar to (33) but for maternity.

(36) *Known offspring-mother dyad (String, optional)*. If the number of known maternity is larger than zero, then on each row list a single dyad containing the offspring ID followed by its known mother ID.

(37) *Number of known paternal sibships (Integer)*. Provide the number of known paternal sibships in the OFS sample that have unknown fathers. The minimum value is 0.

(38) *Paternal sibship size and members (Integer, String, optional).* For each known paternal sibship with an unknown father or a father not included in the CMS sample, provide on a single line the sibship size (number of offspring in the sibship), followed by the IDs of the offspring in the sibship. The order of offspring IDs has no relevance.

(39) *Number of known maternal sibships (Integer)*. Provide the number of known maternal sibships in the OFS sample that have unknown mothers. The minimum value is 0.

(40) *Maternal sibship size and members (Integer, String, optional).* For each known maternal sibship with an unknown mother or a mother not included in the CFS sample, provide on a single line the sibship size (number of offspring in the sibship), followed by the IDs of the offspring in the sibship. The order of offspring IDs has no relevance.

(41) *Number of offspring with known excluded paternity (Integer)*. Provide the number of offspring that each has at least one known excluded candidate male as its father. The minimum value is 0.

(42) *Excluded paternity (String, optional).* If the number of offspring with known excluded paternity is larger than zero, then you need to specify the excluded candidate males for an offspring. Each offspring with known excluded paternity takes one row. The first entry of the row is the offspring ID/name, the 2nd entry is the number of excluded males, followed by the IDs of these males excluded from parentage of the offspring. The order of male IDs has no relevance.

(43) *Number of offspring with known excluded maternity (Integer)*. Provide the number of offspring that each has at least one known excluded candidate female as its mother. The minimum value is 0.

(44) *Excluded maternity (String, optional).* If the number of offspring with known excluded maternity is larger than zero, then you need to specify the excluded candidate females for an offspring. Each offspring with known excluded maternity takes one row. The first entry of the row is the offspring ID/name, the 2nd entry is the number of excluded females, followed by the IDs of these females excluded from parentage of the offspring. The order of female IDs has no relevance.

(45) *Number of offspring with known excluded paternal sibships (Integer)*. Provide the number of offspring that are known to each has at least one excluded offspring as its paternal sibling. The minimum value is 0.

(46) *Excluded paternal siblings (String, Integer, String, optional).* If the number of offspring with known excluded paternal sibships is larger than zero, then you need to specify the excluded offspring as paternal siblings for an offspring. Each offspring with one or more known excluded paternal siblings takes one row. The first entry of the row is the offspring ID/name, the 2nd entry is the number of excluded offspring as paternal siblings, followed by the IDs of these offspring excluded from paternal siblings of the offspring. The order of entries 3, 4, … has no relevance.

(47) *Number of offspring with known excluded maternal sibships (Integer)*. Provide the number of offspring that are known to each has at least one excluded offspring as its maternal sibling. The minimum value is 0.

(48) *Excluded maternal siblings (String, Integer, String, optional).* If the number of offspring with known excluded maternal sibships is larger than zero, then you need to specify the excluded offspring as maternal siblings for an offspring. Each offspring with one or more known excluded

maternal siblings takes one row. The first entry of the row is the offspring ID/name, the 2nd entry is the number of excluded offspring as maternal siblings, followed by the IDs of these offspring excluded from maternal siblings of the offspring. The order of entries 3, 4, … has no relevance.

4.2 An example

An example "colony2.dat" input file is shown below. Anything after "!" in a line is just a note and is ignored by Colony when reading in the data from the file. When the content that needs to be listed in a line is big and thus the line becomes too long, the content can be listed in multiple lines. In such a case, however, no line note ("!" and anything after it) and line continuation mark "&" are allowed.

```
Example1  ! Project name
Example1  ! Output file name
40        ! Number of offspring in the sample
5         ! Number of loci
1234      ! Seed for random number generator
0         ! 0/1=Not updating/updating allele frequency
2         ! 2/1=Dioecious/Monoecious species
0         ! 0/1=no inbreeding/inbreeding
0         ! 0/1=Diploid species/HaploDiploid species
1  1      ! 0/1=Polygamy/Monogamy for males & females
0         ! 0/1=Clone inference =No/Yes
1         ! 0/1=Scale full sibship=No/Yes
0         ! 0/1/2/3=No/Weak/Medium/Strong sibship prior; 4=optimal sibship prior
0         ! 0/1=Unknown/Known population allele frequency
1         ! Number of runs
2         ! 1/2/3/4=short/medium/long/very long run
0         ! 0/1=Monitor method by Iterate#/Time in second
10000     ! Monitor interval in Iterate# / in seconds
0         ! 0/1=No/Yes for run with Windows GUI
1         ! 0/1/2=PairLikelihood score/Fulllikelihood/FPLS
2         ! 0/1/2/3=Low/Medium/High/Very high precision with Fulllikelihood

     mk1     mk2     mk3     mk4     mk5    !Marker Ids, 5 loci
       0       0       0       0       0    !Marker types, 0/1=Codominant/Dominant
  0.0000  0.0000  0.0000  0.0000  0.0000    !Allelic dropout rate at each locus
  0.0001  0.0001  0.0001  0.0001  0.0001    !Other typing error rate at each locus

   O1  11  11   8  11  11   3   9   2   1   2  !Offspring ID and genotypes at locus 1~5
   O2  11  11   8   3  11   3   9  12   1   1
   ……
  O39  12   5   8   3   5   6   5   7   7   6
  O40  12   2   3   3   6  10   5   7   7   6

0.5  0.5     !probabilities that the father and mother of an offspring are included in candidates
30    40     !Numbers of candidate males and females

   M1  11  12  11   3   3  12   2  12   1   2  !Candidate male ID and genotypes at locus 1~5
   M2   4   8   3   4  12  13   5   7  10   4
  ……
  M29   4   3   9  13   4  10   5   6   2  12
  M30   9  12   2   6  10  10   6  13  10  11

   F1  11  12  11   3   3  12   2  12   1   2  !Candidate female ID and genotypes at locus 1~5
   F2   4   8   3   4  12  13   5   7  10   4
   ……
  F39   4   3   9  13   4  10   5   6   2  12
  F40   9  12   2   6  10  10   6  13  10  11

3   5              !Number of offspring with known paternity, exclusion threshold
O33 M1             !IDs of known offspring-father dyad
O34 M1
 O1 M2

0   0              !Number of offspring with known maternity, exclusion threshold

2                  !Number of known paternal sibship
2  O23 O25         !Size of known paternal sibship, and IDs of offspring in the sibship
3  O7 O21 O39

2                  !Number of known maternal sibship
2  O15 O20         !Size of known maternal sibship, and IDs of offspring in the sibship
4  O11 O12 O19 O27

2                  !Number of offspring with known excluded paternity
O31 3  M3 M5 M7    !Offspring ID, number of excluded males, the IDs of excluded males
O1  1  M9

2                  !Number of offspring with known excluded maternity
O31  3 F3 F5 F7    !Offspring ID, number of excluded females, the IDs of excluded females
O1   1 F9

0                  !Number of offspring with known excluded paternal sibships
0                  !Number of offspring with known excluded maternal sibships
```

5. Simulation data input – Windows GUI

A simulation project generates simulated genotype data for a given parameter combination including sibship and parentage structure and marker properties (number of loci, number and frequencies of alleles at a locus, mistyping rates, …). The simulated data can then be analyzed directly by Colony, and can be imported into Excel and R to be formatted for analysis by other software.

Similar to that described in section 3, the contents of a line required and described below can appear in multiple consecutive lines, with a line continuation mark "&" at the end of a line.

5.1 Set up a new project

This is the same as described in section 3, except the "Simulation Data Analysis" option should be chosen herein.

5.2 Input parameters

In page 1, a number of parameters are required to be set. In most cases, the default values of the parameters are fine. Some parameters are required for simulation only, some for running Colony only, and the rest for both. These are indicated by (S), (C), and (SC) below, respectively, and the corresponding label texts are in blue, green and red in this page's GUI. When the mouse is resting on a label of the GUI, a tooltip explaining the corresponding parameter input will appear.

(1) Number of replicates (S): It specifies the number of replicate datasets to be simulated and analyzed. Note that although simulating data is very quick, analyzing data is slow, especially when both sexes are polygamous and markers have non-zero mistyping rates. Except when offspring sample size is very small or many cpus are used in parallel run, specifying a large number of replicates (say, 100) can take a long time to finish. For serial run with a moderate offspring sample size (~100), I suggest using around 10 replicates.

(2) Mating system-I (SC): It specifies the male and female mating systems, as explained in section 3. You can choose one of 4 options: "Male monogamy, Female monogamy", "Male monogamy, Female polygamy", "Male polygamy, Female monogamy", "Male polygamy, Female polygamy". The meanings of "monogamy" and "polygamy" in the context of Colony are explained in "(1) ***Mating system-I***" of section 3 above. Changing the mating system will erase any mating data already loaded in the next page. In such a case, you need to reload the relevant data.

(3) Mating system-II (C): The same as in section 3.1.2, except for the inference of clones which are always not inferred. Changing the mating system will erase any mating data already loaded in the next page.

(4) Species (SC): There are 3 species models to choose: "Dioecious, Diploid", "Dioecious, HaploDiploid", "Monoecious, Diploid". Changing species will erase any mating data already loaded in the next page.

(5) Analysis method (C): The same as in section 3.

(6) Likelihood precision (C): The same as in section 3.

(7) Number of runs (C): The same as in section 3.

(8) Length of run (C): The same as in section 3.

(9) Allele frequency (C): There are 3 options: "Known", "Unknown, No update", and "Unknown, Update". The 1st option specifies that the actual allele frequencies used in simulating genotype data are to be used as input data for Colony. The 2nd option specifies that no allele frequency data will be included in the Colony input file, and allele frequencies for pedigree reconstruction will be calculated by Colony from genotype data without updating. The 3rd option is the same as the 2nd, except that allele frequencies will be updated periodically by Colony by incorporating reconstructed pedigrees.

(10) Parental F (S): It specifies the inbreeding coefficient of parents, and is valid for dioecious only. It is used to generate parental genotypes.

(11) Parental selfing rate (S): It specifies the selfing rate of parents, monoecious only. It is used to generate parental genotypes. The selfing rate of the sampled offspring of each parent is determined by the mating matrix defined below.

(12) Sibship prior (C): The same as in section 3.

(13) Paternal sibship size (C): The average size (number of offspring) of a paternal sibship. Valid only when the sibship size prior is chosen (i.e. = 1, 2, 3) in (12).

(14) Maternal sibship size (C): The average size (number of offspring) of a maternal sibship. Valid only when the sibship size prior is chosen (i.e. = 1, 2, 3) in (12).

(15) Genetic map length (S): The length of the genome in Morgans on which the specified number of markers (below) will be generated. Markers are assumed to be equally spaced in the genome (more details see Wang 2004). For free recombination of unlinked markers, set the value =-1.

(16) Random number seed (C): The same as in section 3.

(17) Inferring clones (C): The same as in section 3.

(18) Sibship size scaling (C): The same as in section 3.

5.3 Matings

In page 2, the mating and family structure related information should be provided.

(1) Mating Structure (S): The mating (or family, or sibship and parentage) structure of the data to be simulated is specified by a mating matrix. The matrix has m ($\geq$1) rows and f ($\geq$1) columns, where m and f are the numbers of male and female parents contributing to the sibship structure. In the $m \times f$ matrix, element q_{ij} ($\geq$0) gives the number of full siblings of male parent (row) i (=1~m) mated with female parent (column) j (=1~f). The sum of values on a row (column) gives the total number of offspring produced by a male (female). Therefore, a male (female) is redundant if all values on the row (column) are zeros, because it has no offspring to be included in the simulated sample. Consider the following numerical example of a mating matrix.

```
5  0  2  1
3  1  1  0
```

It indicates that the first male (row 1) is mated with females (columns) 1~4, producing 5, 0, 2 and 1 (full-sib) offspring, respectively, to be included in the sample. The second male (row 2) is mated with females (columns) 1~4, producing 3, 1, 1, and 0 (full-sib) offspring, respectively, to be included in the

sample. From this mating matrix, it can be seen the maximal effective number of matings with 1 or more offspring is 3 for a male, and is 2 for a female. Therefore, both males and females should be designated as polygamous. If the mating system implied by the mating matrix is in conflict with the parameter option you choose in "Mating system – I", a warning message will be issued and the mating matrix will be erased.

The mating matrix is essential and is uploaded by left clicking the load button. If the matrix is loaded, then right clicking the load button will erase the matrix. For monoecious species, the number of rows and the number of columns must be equal, because the same set of parents are assumed to contribute male and female gametes.

An example of a mating matrix for dioecious monogamous males and females and variable full sib family sizes is

```
1   0   0   0   0
0   2   0   0   0
0   0   4   0   0
0   0   0   8   0
0   0   0   0   16
```

An example of a mating matrix for monoecious species with variable selfing rates of different parents is

```
1   1   1   1   1
1   2   0   0   0
0   0   4   0   0
0   0   0   8   0
0   0   0   0   16
```

This mating matrix signifies that female parent 1 (column 1) contributes 2 female gametes, which combine with a male gamete from the same adult (on row 1) to produce a selfing offspring and with another male gamete from another adult (on row 2) to produce an outbred offspring. Therefore, the selfing rate of female parent 1 is 0.5 (one out of 2 offspring is from selfing). Similarly, female parents 2, 3, 4, and 5 have a selfing rate of 2/3, 4/5, 8/9, and 16/17 respectively.

(2) # Mating Matrices (S): It specifies the number of (replicate) mating matrices explained above to be simulated. The total number of offspring in the simulated sample is equal to the product of the "# Mating Matrices", and the sum of offspring in a single mating matrix. It is possible to define a (1- or 2-generation) pedigree of an arbitrary size and complexity of a sample of individuals by using a single mating matrix. However, the matrix can become huge for a large sample. "#Mating Matrices" is used to avoid the mating matrix too big and too repetitive.

(3) # Dads in a Matrix (S): Specifies the number of dads that contribute to a single mating matrix. It must be equal to the number of rows in a single mating matrix. The total number of dads that contribute to the offspring in the simulated sample is the product of the "# Mating Matrices" and "# Dads in a Matrix".

(4) # Mums in a Matrix (S): Specifies the number of mums that contribute to a single mating matrix. It must be equal to the number of columns in a single mating matrix. The total number of mums that contribute to the offspring in the simulated sample is the product of the "# Mating Matrices" and "# Mums in a Matrix".

(5) # Candidate Males (S): Specifies the number of unrelated males with genotypes to be simulated and included in the candidate father list. The list will also include any known fathers and any unknown but sampled fathers, both being specified below.

(6) # Candidate Females (S): Specifies the number of unrelated females with genotypes to be simulated and included in the candidate mother list. The list will also include any known mothers and any unknown but sampled mothers, both being specified below.

(7) Prb. of Dad (optional) (C): This is the estimated (assumed) probability that an actual father is included in the candidate father list, as in section 3.

(8) Prb. of Mum (optional) (C): This is the estimated (assumed) probability that an actual mother is included in the candidate mother list, as in section 3.

(9) Numbers of offspring with known dad & mum (S): When left clicking the load button, you are allowed to load an $m \times f$ matrix, with element r_{ij} (≥ 0) giving the number of full siblings whose dad, male (row) i (=1~m), and whose mum, female (column) j (=1~f), are both known, and the known paternity and maternity will be included in the input file for Colony run. Obviously, $r_{ij} \leq q_{ij}$. Otherwise, an error message is displayed, and the newly loaded matrix for r_{ij} is erased. Right clicking the load button will erase the matrix already loaded. When the matrix is left unloaded, all values in the matrix are assumed zero.

(10) Numbers of offspring with known dad only (S): This matrix is similar to that of (9), except that the element s_{ij} (≥ 0) gives the number of full siblings whose dad, male (row) i (=1~m), is known and whose mum, female (column) j (=1~f), is unknown. The known paternity will be included in the input file for Colony run. Obviously, $s_{ij}+r_{ij} \leq q_{ij}$. Otherwise, an error message is displayed, and the newly loaded matrix for s_{ij} is erased. Right clicking the load button will erase the matrix already loaded. When the matrix is left unloaded, all values in the matrix are assumed zero.

(11) Numbers of offspring with known mum only (S): This matrix is similar to that of (9), except that element t_{ij} (≥ 0) gives the number of full siblings whose dad, male (row) i (=1~m), is unknown and whose mum, female (column) j (=1~f), is known. The known maternity will be included in the input file for Colony run. Obviously, $t_{ij}+s_{ij}+r_{ij} \leq q_{ij}$. Otherwise, an error message is displayed, and the newly loaded matrix for t_{ij} is erased. Right clicking the load button will erase the matrix already loaded. When the matrix is left unloaded, all values in the matrix are assumed zero.

(12) Sampled dads in a mating matrix (S): This checked list box allows you to select, among the m males in a mating matrix, which are (by checking the box) and which are not (by unchecking the box) sampled and included in the candidate male list. Any male that is checked herein or indicated as known in (9) and (10) will be included in the sample.

(13) Sampled mums in a mating matrix (S): This checked list box allows you to select, among the f females in a mating matrix, which are (by checking the box) and which are not (by unchecking the box) sampled and included in the candidate female list. Any female that is checked herein or indicated as known in (9) and (11) will be included in the sample.

5.4 Markers

In page 3, the information about the markers should be provided.

(1) # Loci (SC): The number of loci genotyped, the same as in section 3.

(2) Allele Frq Dist (S): Specifies the distribution of allele frequencies at each locus to be simulated. There are 4 options: Uniform, Equal, Triangular, Empirical. The first gives a uniform distribution of allele frequencies, drawn from the Dirichlet with all parameter values of 1. The second and third gives $p_i = 1/k$ and $p_i = i/[k(1+k)/2]$ respectively for allele i=1~ k, where k is the number of alleles at a locus. The fourth specifies that empirical allele frequencies are to be loaded (see below).

(3) G Missing Freq. (S): The frequency that a single locus genotype is missing in the simulated data. This frequency is a constant across loci, and missing genotypes occur independently across loci and individuals.

(4) Load Marker Types and Error (SC): There should be 5 lines (rows) and L columns in this text file, where L is the number of marker loci as specified in (1) above. For column l (l =1, 2, …, L), the values on lines 1~5 give the marker name (ID, string), marker type (0 for codominant, 1 for dominant), number of alleles (>1, integer), allelic dropout rate (≥0 and ≤1, float number), and false allele rate (≥0 and ≤1, float number) at marker locus l. Note that the file is similar to that as described in "*Marker type and error rate*" in section 3. However, it has an extra third line which specifies the number of alleles per locus. An example input looks like this,

```
     mk1      mk2      mk3      mk4      mk5
       0        0        0        0        0
       2        3        4        5        8
  0.0000   0.0000   0.0000   0.0000   0.0000
  0.0001   0.0001   0.0001   0.0001   0.0001
```

(5) Load Allele Frequency (SC): This button is enabled only when the allele frequency distribution is specified as Empirical (see above). Clicking this button will load an allele frequency file. In this file, the allele frequencies at locus i (i =1, 2, …, L, where L is #Loci defined in (1)) are listed in row i. Note that, for a given locus (row) i, the number of allele frequency values must be equal to the number of alleles at this locus defined in (4), each value should be in the range [0, 1], and the sum of the values must be in the range [0.99,1.01]; Otherwise, an error message is issued. An example of the allele frequencies for markers mk1 and mk2 as specified in (4) could be

```
0.1000 0.9000
0.0500 0.1405 0.8095
```

5.5 Save data

In page 4, only one control exists. By clicking the "Save Data" button, the program will first check all inputs in the previous pages. When any apparent errors are detected, it will stop and display the error message. In such a case you need to modify the data and input them again. After passing the check, the program saves the input into a file, called "Input3.Par", in the project folder.

6. Simulation data input – NO Windows GUI

If one chooses to run the simulation module in MS-DOS mode on PCs or run the program in other platforms such as Mac, one just needs to prepare an input file called "Input3.Par" which includes all of the parameter values and contains all of the data required by the simulation program Simu2.exe to generate and analyze simulated data. The DOS mode is preferred to Windows GUI in the cases of a simulation study of many large datasets involving many loci or/and individuals.

For PC users, it is also possible to prepare and input the data using the GUI when the dataset is not very large, and then run the simulation program Simu2.exe in DOS mode. This hybrid procedure combines the convenience of the GUI for data input and the running speed of the DOS mode. Once data input is complete, an input file called "input3.par" is generated in the colony project folder. To run the data in DOS mode, copy all executables and dynamic link library files (with extension name .dll) in Colony program folder into the new project folder. Double click Simu2.exe will start the simulation in a serial run. Once finished, the results should be found in your colony project folder (where Input3.Par was originally found), and you can use GUI to view the results.

In the following, I specify how to prepare the Input3.Par input file manually using a text editor.

6.1 Data input

For DOS mode or other platforms, the same data as for the Windows GUI described in section *5* are required. However, the difference is that all data must be put in a single pure text file, named "Input3.Par", in the following order and format. Also the delimiters must be either white spaces or commas. Empty rows take no effect. More details about the meaning and format requirement of each input item were explained in section *5*, and the meanings of (S), (C), and (SC) and the corresponding colours blue, green and red were also described in section *5*.

(1) ***Project name (String, SC)*:** On the 1st line of the input file, give a project name, which should be a string containing less than 40 letters and numbers. Most or all output files from the simulation and Colony analysis will use the same project name but different extension names.

(2) ***Number of replicates (Integer, S)*:** On the 2nd line of the input file, give the number (≥1) of replicates.

(3) ***Analysis method (Integer, C)*.** Give an indicated value of 0, 1 or 2 to tell Colony to use the pairwise-likelihood score (PLS), full likelihood (FL), or the FL and PLS combined (FPLS) method.

(4) ***Precision (Integer, C)*.** Give an indicated value of 0/1/2/3 to use Low/Medium/High/VeryHigh precision in calculating the full likelihood.

(5) ***Species (Integer, SC)*** and ***parental selfing rate (real, S)*.** Provide an indicator of value 1 or 2 for monoecous or dioecious species, and a real number (in range [0,1]) for the parental selfing rate. This rate is valid for monoecious only, but must be present for all cases.

(6) ***# Mating matrices (*Integer, *S):*** For more, see section *5*. It must be ≥1.

(7) ***# Dads and # Mums in a matrix (*Integer, *S):*** For more, see section *5*. Both values must be ≥1.

(8) ***Mating Structure (*Integer, *S):*** For more, see section *5*.

(9) ***Numbers of offspring with known dad & mum (*Integer, *S):*** See (9) of section *5*.

(10) ***Numbers of offspring with known dad only (*Integer, *S):*** See (10) of section *5*.

(11) ***Numbers of offspring with known mum only (*Integer, *S):*** See (11) of section *5*.

(12) ***Sampled dads in a mating matrix (Boolean, S):*** Provide, on a single row, *m* indicator values of 0 (1) to indicate the actual dad is (is not) sampled, genotyped and included in the candidate father list, where *m* is the number of dads in a single mating structure (matrix). For more see (12) of section *5*.

(13) ***Sampled mums in a mating matrix (Boolean, S):*** Provide, on a single row, *f* indicator values of 0 (1) to indicate the actual mum is (is not) sampled, genotyped and included in the candidate mother list, where *f* is the number of mums in a single mating structure (matrix). For more see (13) of section *5*.

(14) ***# Candidate males and # candidate females (****Integer, **S):*** see (5) and (6) of section *5*.

(15) ***Prb. of dad and Prb. of mum (Real, C):*** The two real numbers are necessary, but are useful only when the candidate father and mother lists are not empty. More see (7-8) of section *5*.

(16) ***Number of loci (SC, Integer)*****:** Provide this number (≥1) of loci to be simulated.

(17) ***G Missing Freq. (Real, S):*** See (3) in *5.4*.

(18) ***Drop rate at each locus (Real, SC):*** A number (= #loci) of real values (≥0 and <1) that specify the allelic dropout rate at each locus, see (4) in *5.4*.

(19) ***Other error rate at each locus (Real, SC):*** A number (= #loci) of real values (≥0 and <1) that specify the rate of other errors (including false alleles) at each locus, see (4) in *5.4*.

(20) ***Marker types (Boolean, SC):*** Specifies the type (0 and 1 for codominant and dominant) of markers at each locus.

(21) ***Number of alleles at each locus (Integer, S):*** For a dominant marker, the number of alleles is fixed at 2.

(22) ***Allele Frq Dist (Integer, S):*** It takes 4 values, 0-3 to indicate Uniform, Equal, Triangular, Empirical frequency distribution (see (2) in 5.4).

(23) ***Allele frequencies at each locus (Real, SC, optional):*** When the allele frequency distribution is specified as empirical in (22), you must provide the allele frequencies at each locus, as detailed in (5) in 5.4. Otherwise, no input is needed.

(24) ***Species ploidy (Integer, SC):*** Provide a value of 1 or 2 for HaploDiploid or Diploid species, respectively.

(25) ***Mating systems for males and females (Boolean, SC):*** Use values of 0 and 1 to indicate polygamous and monogamous, respectively, for males and females.

(26) ***Random number seed (Integer, C):*** The same as in section 5.2.

(27) ***Sibship prior indicator (Integer), average paternal sibship size (Real, optional), average maternal sibship size (Real, optional).*** Same as in section 5.

(28) ***Inferring clones (C):*** The same as in section 5.

(29) ***Sibship size scaling (****Boolean, C)****:* The same as in section 5.

(30) ***Allele frequency known or unknown (****Boolean, C)****:*** If the indicator is set as 1, then the allele frequencies simulated or inputted for each locus in generating the genotype data are included in the data input file, colony2.dat, and sibship/parentage reconstruction will use these frequencies as known. Otherwise if the indicator is set as 0, allele frequencies will be assumed unknown in colony analysis, and Colony will calculate allele frequencies from genotype data.

(31) ***Updating allele frequency (Boolean, C):*** A value of either 0 or 1 is needed, but is valid and useful only when the indicator value in (30) is set as 0. In such a case (i.e. allele frequency unknown), Colony will update allele frequencies by taking into account of reconstructed pedigrees if this indicator value is set as 1 (updating). Otherwise, no updating of allele frequencies.

(32) ***Number of replicate runs (Integer, C):*** It specifies the number of replicate runs for a single dataset. For simulation, always use 1.

(33) ***Length of Run (Integer, C)*****:** Give a value of 0, 1, 2, 3, 4 to indicate VeryShort, Short, Medium, Long, VeryLong run. Suggested values are 1 or 0.

(34) ***Genetic map length (S):*** See section 5.

(35) ***Parental F (S):*** See section 5.

(36) ***Outbreeding/Inbreeding (Boolean, C)*****:** Specify a value of either 0 or 1 to indicate the absence or presence of inbreeding in Colony analysis.

(37) ***Windows/DOS (Boolean, C)*****:** Give an indicator number of 0 or 1 to indicate running Colony in the Windows GUI or DOS mode. Always set the value to 1 when running Colony in DOS mode.

(38) ***Monitor interval (Integer, C)*****:** Give a number to specify the interval by which intermediate results are directed to the monitor. The unit of the interval is number of iterates and seconds in time for DOS and Windows GUI, respectively. Always set the value to 10000 or 100000 when running Colony in DOS mode.

6.2 An example

An example "Input3.Par" input file is shown below. Anything after "!" in a line is just a note and is ignored when reading in the data from the file. When the content that needs to be listed in a line is big and thus the line becomes too long, the content can be listed in multiple lines. In such a case, however, no line note ("!" and anything after it) is allowed. The following example dataset is included in the software package, in folder "Simulation Dataset 1", but the information is scattered in several different text files for easy input into the GUI for testing purpose.

```
SimuTest                        !Project (output file) name
2                               !Number of replicates
1                               !0/1/2=Pair-likelihood Score(PLS)/Full likelihood(FL)/FL-PLS combined
(FPLS) method
0                               !0/1/2/3 for low/medium/high/very high precision
2 0                             !2/1=Dioecious/Monoecious, Selfing rate for monoecious
1                               !Number of mating matrices
5 5                             !#dads & mums in a mating structure
```

```
1 0 0 0 0                       !#full siblings of dad i (row, 1-5) with mum j (col, 1-5)
0 2 0 0 0
0 0 4 0 0
0 0 0 8 0
0 0 0 0 16

0 0 0 0 0                       !#full siblings of known dad i (row, 1-5) and known mum j (col, 1-5)
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 1

0 0 0 0 0                       !#full siblings of known dad i (row, 1-5) and unknown mum j (col, 1-5)
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0                       !#full siblings of unknown dad (row, 1-5) and known mum j (col, 1-5)
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0

1 0 0 0 0                       !Dad 1-5 included (1) in or excluded (0) from candidate list
0 0 0 0 0                       !mum 1-5 included (1) in or excluded (0) from candidate list

10  10                          !# unrelated candidate males & females
0.50  0.50                      !Assumed prbs of fathers & mothers included in candidates
10                              !Number of Loci
0.05                            !Prob. of missing genotypes

.01 .02   0 0 0 0 0 0 0 0       !Drop rate for each locus
.01   0 .01 0 0 0 0 0 0 0       !OtherErrorRate for each locus
  0   0   0 0 0 0 0 0 0 0       !Marker types, 0/1=codominant/dominant
  9   9   9 9 9 9 9 9 9 9       !# alleles at each locus

0                               !0/1/2/3=Uniform/Equal/Triangular/other allele freq. distr.
2                               !1/n=HaploDiploid/n-ploid species
1  1                            !1/0=Monogamy/Polygamy for males & females
1234                            !Seed for random number generator
0  0.0  0.0                    !0/1/2/3=No/Weak/Medium/Strong sibship prior
0                               !B, 0/1=Clone inference =No/Yes
1                               !B, 0/1=Scale full sibship=No/Yes
0                               !1/0 (Y/N) for known allele frequency
0                               !1/0 for updating allele freq. or not
1                               !#replicate runs
0                               !0/1/2/3/4=VeryShort/Short/Medium/Long/VeryLong run
-1                              !Map length in Morgans. <0 for infinite length
0.0                             !Inbreeding coefficient of parents
0                               !0/1=N/Y for allowing inbreeding in Colony
0                               !0/1 for Windows DOS/GUI run, parameter in Colony
100000                          !#iterates/#second for Windows DOS/GUI run, parameter in Colony
```

7. Running empirical data analysis – Windows GUI

7.1 Start the run

Once the data input is complete, you can start running Colony by clicking on Run→Start Running. You are then asked how many threads you want to use for the computation of the present run. Even for a computer with a single CPU that has a single core, you can still use multiple threads. But due to communication cost between threads, there is no benefit in doing this. Actually, the computation would slow down significantly. The suggestion is, never set more threads than the total number of cores in your computer or the number of markers in your dataset, whichever is smaller.

In most cases, the computational bottleneck in Colony is the computation of the likelihood of a relationship configuration. Code for this section is thus parallelized by MPI (Message Passing Interface) to allow the use of multiple cores/CPUs. Parallelization is over loci, so that each thread calculates the log-likelihood for a subset of the loci, and the results are summed and broadcast to all

threads. Therefore, the best performance is achieved usually when each thread has an equal share of the computational load (number of loci). If you run a dataset with n loci on a computer with k cpus, a better performance is obtained by using k-1 rather than k cpus if n is a multiple of (k-1). For good performance, never set a number of threads larger than the number of loci or the number of cores in your computer, whichever is smaller.

If you are not sure about the cpus/cores of your computer, or about other things, choose the default value of 1 for a serial run. If you choose a value of 2 or above for a parallel run, the run may fail if you have not correctly configured the parallelization specific to your computer (see details in Installation).

Once the Colony program is running, in the Running Status window (Figure 9), the upper ProgressBar shows the progress in annealing temperature (assuming a maximum of 200 temperature reductions), while the lower ProgressBar shows the progress in iterates within a given temperature. Note that these ProgressBars act only as rough indicators to the progress of computation, especially the upper one. This is because it is not predictable exactly when the computation is to be finished. The intermediate results, explained below, are outputted into the text box.

Run : The replicate run number. Variable
Tmr : The number of temperature reductions so far within the run. Variable
Itr : The number of iterates (reconfigurations considered) so far within the run. Variable
NSucc : The number of successful (accepted) reconfigurations so far within the temperature. Variable
NSuccLmt : Maximum (Limit) number of successful reconfigurations allowed within the temperature. Constant
NFail1 : The number of reconfigurations since the last update of the best likelihood within the temperature. Variable
NFail1Lmt : Maximum (Limit) value of *NFail1* within the temperature. Constant
NFail2 : The total number of reconfigurations since the last update of the best likelihood within the run. Variable
NFail2Lmt : Maximum (Limit) value of *NFail2* within a run. Constant. The run terminates when *NFail2Lmt=NFail2* and the successful rate (see below) < 0.01
SucRate% : *=NSucc / Itr*. Variable
SucLmt% : *=NSucc / NSuccLmt*. Variable
FailLmt% : *=NFail1 / NFail1Lmt*. Variable
IterLmt% : = (Number of iterates) / (Maximum number of iterates) within a temperature. Variable
CrLogL : The log likelihood of the current configuration. Variable
#F1 : Current number of paternal sib families. Variable
#F2 : Current number of maternal sib families. Variable
#F3 : Current number of sib family clusters. Variable
#FS : Current number of full sib families. Variable
HSPair : Current number of half-sib dyads. Variable
FSPair : Current number of full-sib dyads. Variable
#AssgnC1 : Current number of candidate males that are assigned parentage. Variable
#AssgnC2 : Current number of candidate females that are assigned parentage. Variable
#AssgnP1 : Current number of offspring that have assigned paternity. Variable
#AssgnP2 : Current number of offspring that have assigned maternity. Variable

Similar outputs are given for the current best configuration (the line below *CrLogL*)

When multiple threads are instructed in running the dataset, by default only the local computer is used. To use multiple computers (each may have 1 or more cpus/cores), you have to use the DOS mode described below.

To use multiple cores/cpus in a DOS mode run, follow the following steps. First, change the data input file as described in **4**. **Empirical data input – NO Windows GUI**. Second, open a DOS window and go to the colony program folder (using DOS command cd) where "smpd.exe" file is found. Third, you need to install, with administrator's previlage, "smpd" with the command line "smpd -install". Fourth, type the command line "mpiexec -localonly -n x colony2p.exe" to start the colony run with x threads on the local computer only. For using multiple computers, consult your IT support team.

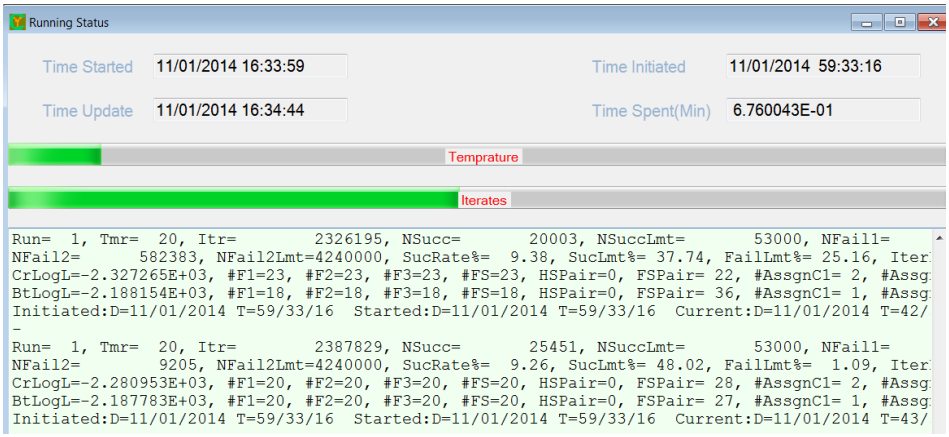


Figure 9: The running status window

7.2 Monitor progress by graph

By clicking on Run→ Show Running Status Plot, you can select to show in graph form the dynamic changes of an interesting quantity (e.g. LogLikelihood, number of full-sib dyads inferred) (Figure 10A). You can copy the graph by right clicking it.

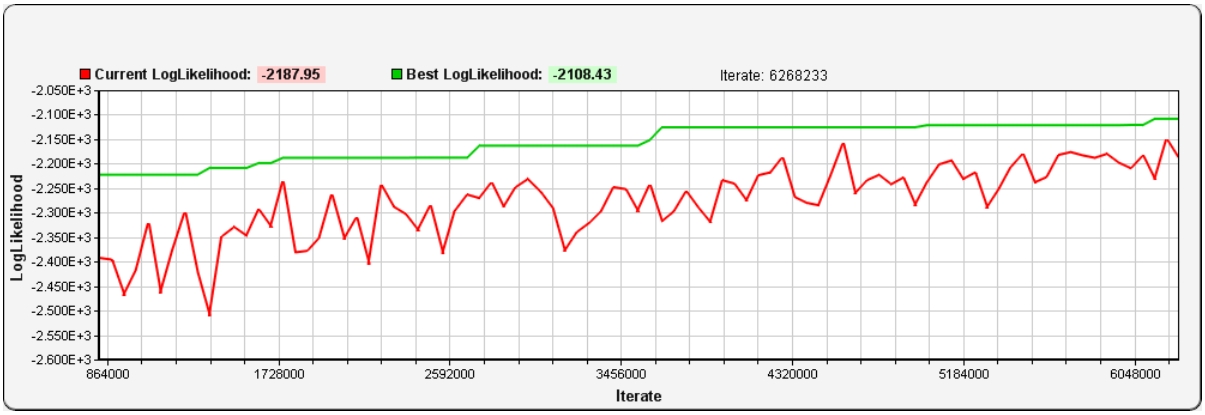


Figure 10A: The running status in real time chart window

Similarly, you can view the changes in the current assignments of sibship, paternity (if CMS exists) and maternity (if CFS exists) by clicking on Run→ Show Assignments Plot (Figure 10B). Pointing the mouse to an assigned dyad will show the indexes of the dyad (may not work very well in real-time assignment graph). Right click the graph to copy it.

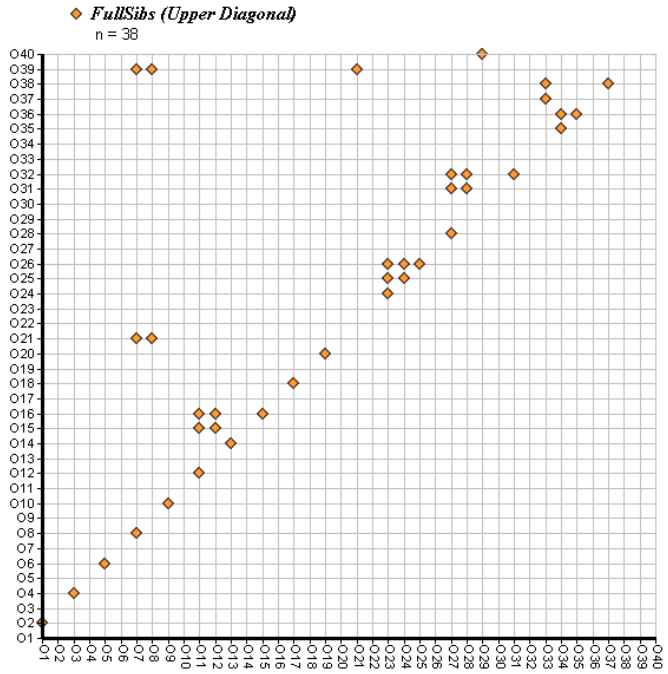


Figure 10B: The current sibship assignments.

7.3 Show running status

The computation could take a long time to finish, depending on your sample sizes, number of markers, typing errors, mating systems, etc. During the running process, you can minimize or close the Running Status window or any real time graph window. Then later when you want to check the running status again, you can re-open these windows. It is suggested that when you have finished viewing the running status or graphs, always close these windows to use CPU fully for computation.

7.4 Stop and restart running

You can stop running your dataset by clicking on Run→Stop Running. The most recent intermediate results are saved into files named "*.Midresult#" in your project folder, where #=1-5. In fact, the intermediate results are saved to your Colony project folder every 5 CPU minutes consumed by Colony program.

When next time you restart Colony by opening the project and running the dataset, you will be asked "Continue from last break?" If you take the alternative answer "No", the program will erase all intermediate results and restart a completely new run. Note that the action is NOT reversible and will cause the loss of all intermediate results! If take the default answer "Yes", the program will read in the intermediate results and continue the run from the break (see below for more options).

If take the default answer "Yes", you are further asked "Cut short the current run & output the current best results?" If you take the default answer "No", the program will read in the intermediate results and continue the run from the break. If you choose the alternative answer "Yes", the program will read in the intermediate results, finalize and output the best results and finish the run almost immediately. This option is useful in viewing the best results so far obtained. The intermediate results are not destroyed so that, after taking this option and viewing the best results, you can still restart Colony to make it continue the run from the breaking point next time.

Sometimes Colony can be very slow. A large dataset containing many individuals with a polygamous mating system of both sexes and containing markers with substantial mistyping rates can take weeks or even months to analyze. Initially, one may set a medium or long run of the dataset, and later find that it takes too long to finish and want to stop the run with the best results reported. In such a case, one can take the "Cut short the current run & output the current best results?" option. How close the results obtained from this cut-short run to the best possible results that would have been obtained without cutting the run short depends on how far the run was from the finishing point before the break. A useful indicator is the intermediate output SucRate% (see 4.1.1). In general, the smaller the value is, the closer the random searching process is close to the finishing point. However, when marker information is ample and the genetic structure is strong, the run can finish at a high value of SucRate%.

8. Running empirical data analysis – NO Windows GUI

Place the completed input file "colony2.dat" in the same directory where the Colony executable, Colony2p.exe, and a dynamic link library, libguide40.dll, are found. Then run the executable. You can stop and restart from where it stopped as in the Windows version. Additionally, running Colony2p.exe directly in DOS or other platforms (linux, Mac) allows you to use an input file with a name other than "colony2.dat". The command is

```
Colony2p.exe IFN:YourInputFileName
```

Similarly, you can cut the current run short and make colony output the current best results by command

```
Colony2p.exe RCS:1
```

You can combine the above two commands,

```
Colony2p.exe IFN:YourInputFileName RCS:1
```

where the order for options IFN and RCS takes no effect. Type

```
Colony2p ?
```

will show these options.

9. Running simulation data analysis – Windows GUI

Once the data input is complete, you can start running simulation by clicking on Run→Start Running. You are then asked how many threads you want to use for the simulation of the present run. Even for a computer with a single CPU that has a single core, you can still use multiple threads. But due to communication cost between threads, there is no benefit in doing this. For maximal efficiency, I suggest using the total number of cores in your computer or the number of replicates, whichever is smaller, as the number of threads for a parallel run. Unlike empirical data analysis described above, the parallization is at the entire replicate level in simulations using MPI. Each thread simulates and analyses (using Colony) an independent replicate dataset.

Similar to empirical data analysis, the parallel run for simulations may fail if you have not correctly configured the parallelization specific to your computer (see details in Installation).

Once the simulation program is running, the ProgressBar shows the progress towards completion, and the intermediate results from the master thread only are inputted into the text box for monitoring purpose.

You can stop the simulation by clicking on Run→Stop Running. The most recent intermediate results are saved into files so that the next time you restart the simulation, it will read in the intermediate results and continue running from there.

10. Running simulation data analysis – NO Windows GUI

Copy the executables, Simu2.exe and Colony2S.exe, and dynamic link libraries, impi.dll and libguide40.dll, into your simulation project folder where the complete input file "Input3.Par" is found. Then run the executable Simu2.exe. You can stop and restart from where it stopped as in the Windows GUI. To start a parallel run, see section **7.1.**

11. Batch run of multiple datasets

Colony can be used to analyse multiple datasets in a batch in both parallel (using MPI) and serial modes. This is especially useful for simulations, where many datasets need to be analysed.

A batch run is realized by running the executable ColonyBatchRun.exe in a DOS window (for Windows), or the executable ColonyBatchRun.out in the X-terminal (for Linux or Mac). The procedures are as follows.

11.1 *Prepare a text input file for ColonyBatchRun*. The file contains *n* ($n > 0$) rows, each row gives the name of a complete input file for Colony. Only English letters, numbers 0-9 and underscore can be used for the name of Colony input file, and the total length of the name must be smaller than 100. Note, the output file names in different Colony input files must be different. Otherwise, the same set of output files will be overwritten.
An example input file for ColonyBatchRun reads,

```
colonyInfile.1
colonyInfile.2
colonyInfile.3
…
```

11.2 *Serial run of ColonyBatchRun*. The command line is

```
ColonyBatchRun.exe colony2s.exe infile.txt          (for DOS)
./ColonyBatchRun.out colony2s-ifort.out infile.txt  (for X-terminal)
```

Where the 1st input argument is the relevant colony binary for your OS, which should always be the serial run version, and the 2nd input argument is the input file to be read by ColonyBatchRun. Both arguments are compulsory.

11.3 *MPI parallel run of ColonyBatchRun*. The command line is

```
mpiexec -n 4 ColonyBatchRun.exe colony2s.exe infile.txt          (for DOS)
mpirun -n 4 ./ColonyBatchRun.out colony2s-ifort.out infile.txt   (for X-terminal)
```

In the above examples, 4 parallel threads (cores) are used to analyse the set of *n* colony input files (such that each thread analyses *n*/4 files). Note for Linux, the command line for your cluster can be different from the above. Note also that the relevant MPI must be installed on your computer. If the executable ColonyBatchRun.exe or ColonyBatchRun.out does not work on your computer, you can compile the Fortran source code, ColonyBatchRun.f90, on your computer, invoking openmp and MPI in compiling options.

Note, the same dataset can be analysed in a number of *k* replicates by ColonyBatchRun. For this purpose, you need to prepare *k* Colony input files, which can have identical contents except for output file names and random number seeds. **Both must be unique among files**. Of course you can change other parameters such as run length. However, MPI parallel run may become inefficient for replicate datasets with different run lengths, because the thread for a short run length may have to wait for a long time for the thread for a long run length to finish.

12. Output from the program

When an empirical data analysis or a simulation has finished running, analysis results are directed to a number of pure text files with the same name but different self-explanatory extension names. For empirical data analysis, the name of the files is exactly the project name. For simulation, the name of the files is the project name with suffix "_*i*", where *i* (=1~*n*, *n* the number of replicates) indicates the *i*th replicate. In the following, the wild card * indicates the name of the files, remembering the difference between a simulation and empirical data analysis.

Additional simulated data files (see below) are also available for a simulation run. These files have different but self-explanatory names with suffix "_*i*", with the same extension name ".txt".

All these simulated data files and analysis output files are found in the project folder, and can be read by any text editor. They can be imported into Excel or any other test editors. Preferably, they can be viewed by Colony's GUI, showing the results in tables and graghs. These tables can be copied to clipboard, by selecting the cells and pressing "Ctrl C". Similarly, graphs can be copied to clipboard by right clicking the graphs.

12.1 Full-sib dyads

Full-sib dyads inferred by Colony are listed in a file named "*. FullSibDyad". The results can be loaded by clicking "View Results" → "Fullsib Dyad" in Windows. On each line, the IDs of the full-sib dyad are listed, followed by the probability of such a dyad. An example is shown below. In the Windows version, you can sort the dyads according to offspring ID or probability, by clicking the corresponding column head. The probability of a full-sib dyad is calculated as explained in FAQ 13.4. Note that for a sib pair of individuals A and B, only one of the two possible unordered dyads of {A, B} and {B, A} is listed.

OffspringID1	OffspringID2	Probability
O1	O2	0.973
O1	O5	0.339
O1	O6	0.339
O2	O3	0.026
O2	O4	0.026
O2	O5	0.327
O2	O6	0.327
O3	O4	1.000

Note haploid offspring have no fathers, and are either full sibs (sharing the same mother) or non-sibs (having different mothers). A haploid and a diploid offspring are either half sibs (sharing the same mother) or non-sibs (having different mothers).

12.2 Half-sib dyads

Half-sib dyads inferred by Colony are listed in a file named "*. HalfSibDyad". The results can be loaded for viewing by clicking "View Results" → "Halfsib Dyad" in Windows. On each line, the IDs of the half-sib dyad are listed, followed by the probability of such a dyad. The half-sib dyad shares the same mother, or the same father, but not both. If both males and females are specified as monogamous so that no half-sibs exist in the offspring sample, then the "*. HalfSibDyad" file is missing. Note that for a sib pair of individuals A and B, only one of the two possible unordered dyads of {A, B} and {B, A} is listed.

The information on full- and half-sib dyads can be used to draw a sibship graph, in which offspring indexes (or IDs) are on both *x* and *y* axes and the full-sib, half-sib, and non-sib dyads are indicated by different colors (red, green and blue respectively in Figure 11). The probability of a sibship is indicated by the colored band height. A graph for an example dataset is shown in Figure 11 (drawn by Mathematica).

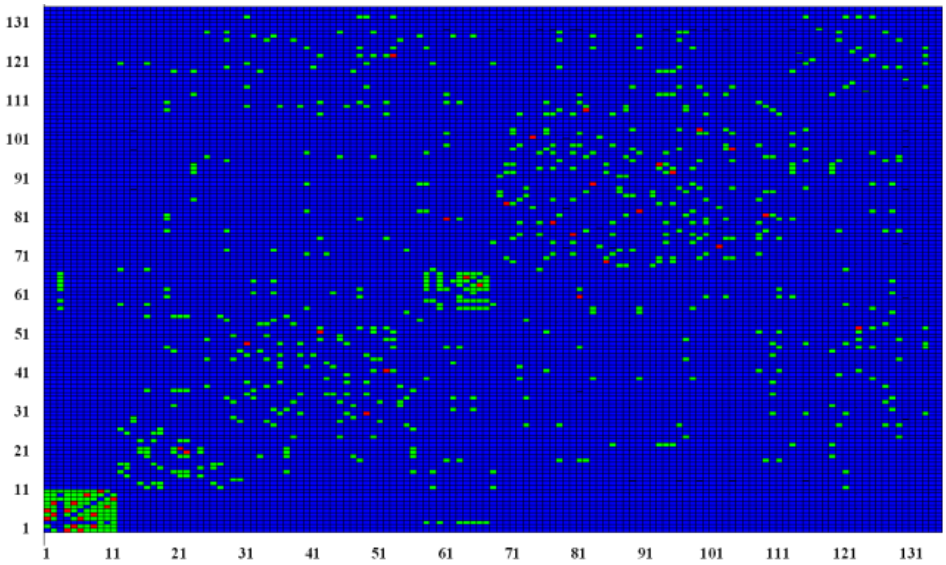


Figure 11: An example graph of full-sib dyad (red), half-sib dyad (green) and non-sib dyad (blue) among 135 offspring with indexes shown on both *x* and *y* axes.

12.3 Paternity

Paternity of each offspring inferred by Colony is summarized in a file named "*.Paternity". The results can be loaded for view by clicking "View Results" → "Paternity" in Windows. On each line, the ID of the offspring is shown on the 1st column, followed by the ID and probability of the 1st inferred father, the ID and probability of the 2nd inferred father,… The inferred fathers are in order of descending probabilities. Only candidate males with a minimum probability of 0.001 of being the father of the focal offspring are listed. Offspring without an inferred father found among the candidate males are excluded from the file. When there are no candidate males, the file "*.Paternity" does not exist. An example paternity file is shown below.

OffspringID	InferredDad1	ProbDad1	InferredDad2	ProbDad2
O1	M2	1.000		

O2	M2	0.856	O11	0.006
O9	M2	0.064		
O10	M2	0.064		

12.4 Maternity

Maternity of each offspring inferred by Colony is summarized in a file named "*.Maternity". The results can be loaded for view by clicking "View Results" → "Maternity" in Windows GUI. On each line, the ID of the offspring is shown on the 1st column, followed by the ID and probability of the 1st inferred mother, the ID and probability of the 2nd inferred mother,… The inferred mothers are in order of descending probabilities. Only candidate females with a minimum probability of 0.001 of being the mother of the focal offspring are listed. Offspring without an inferred mother found among the candidate females are excluded from the file. When there are no candidate females, the file "*.Maternity" does not exist.

12.5 Best(ML) configuration

The best configuration with the maximum likelihood obtained at the end of the computation is given in a file named "*.BestConfig". It can be loaded and viewed by clicking "View Results" → "Best (ML) Configuration" in Windows. In this file, each offspring takes one row in which columns 1~4 show the offspring ID, father ID, mother ID and cluster index (see 11.6 below for an explanation of a cluster). When the inferred father (mother) is not found in the candidates, the father (mother) ID is given an index (starting from 1) prefixed with "*" ("#"). Offspring sharing the same father ID (no matter whether the father is found in the candidate males or not) are paternal sibs, those sharing the same mother ID are maternal sibs. An example file is given below.

OffspringID	FatherID	MotherID	ClusterIndex
O1	M2	F1	1
O2	M2	F1	1
O3	*1	#1	2
O4	*1	#1	2
O5	*2	#2	3
O6	*2	#2	3
O7	*3	#3	4
O8	*3	#3	4
O9	*4	F2	5
O10	*4	F2	5
…			
O33	M1	#11	13
O34	M1	#11	13
O35	*12	#12	14
O36	*12	#12	14
O37	*13	#13	15
O38	*13	#13	15
O39	*3	#3	4
O40	*12	#12	14

Note haploid offspring have no fathers, and their father positions under Column "FatherID" are filled with "*-".

When clone is also inferred, the 1st to the 3rd columns are as shown above, 4th column shows the individual's clone index, and the 5th column shows the cluster index as above. Individuals with the same clone index are clone mates who should have the same multilocus genotype as inferred and reported in the output file *.OffGenotype.

The information in file "*.BestConfig" can be visualized and shown in a pedigree graph by many free software, such as Pedigree Viewer developed by Brian Kinghorn. The pedigree drawn by Pedigree Viewer from the above file is shown in Figure 12.

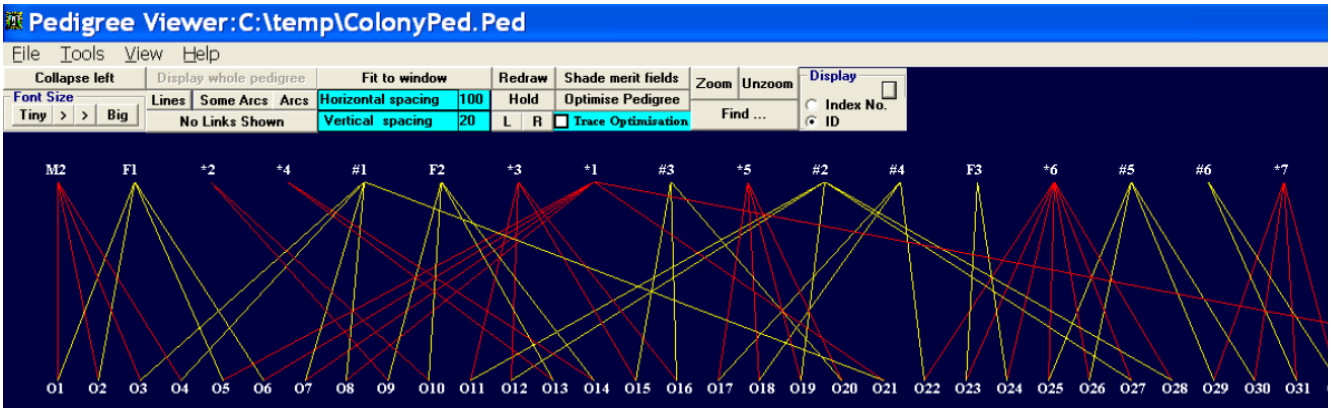


Figure 12: An example pedigree drawn by Pedigree Viewer from the best (ML) configuration file of Colony

The best sibship assignments, paternity assignments (if CMS exists) and maternity assignments (if CFS exists) can also be viewed as graphs.

For monoecious species, fathers and mothers are indistinguishable and given the same index and prefix "#" when they are inferred to be excluded from the candidate parent list. No matter the inferred fathers and mothers are assigned to candidate parents or not, an offspring is from self reproduction if its father and mother are inferred to be the same, and is from outbreeding if its father and mother are inferred to be different.

12.6 Best(ML) cluster

When both males and females are specified as polygamous, then some offspring who do not share parents at all are still linked in the pedigree. As illustrated in Figure 13, O1 and O3 share no parents but are linked through O2. In likelihood calculation, all offspring in a cluster must be considered jointly, while different clusters can be considered independently. The information in the best configuration with the maximum likelihood is also presented in a file named "*.BestCluster". It can be loaded and viewed by clicking "View Results" → "Best (ML) Cluster" in Windows. In this file, each offspring takes one row in which columns 1~5 show the cluster index, the probability of the cluster, offspring ID, father ID, and mother ID.

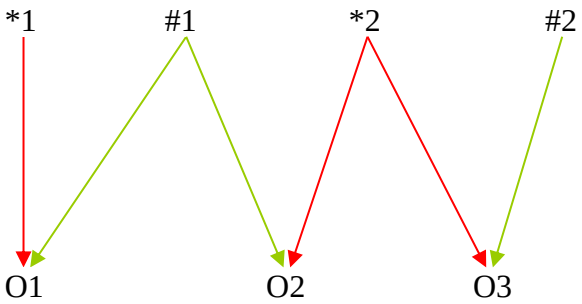


Figure 13: An example pedigree

12.7 Best(ML) full-sib family

Full-sib families inferred by Colony are listed in a file named "*. BestFSFamily". The results can be loaded by clicking "View Results" → "Best (ML) Fullsib Family" in Windows. Some full-sib families may have more than 2 members. In some cases, it is more convenient to show the entire full-sib family rather than pairs of full sibs. The information about the best full-sib families reconstructed by Colony is

contained in a file named "*.BestFSFamily". In this file, each family takes one row, with the 1st column showing the family index, 2nd and 3rd columns showing the inclusive and exclusive probabilities of this family (calculated as explained in FAQ 13.4), followed by the IDs of all the offspring members of this family. An example file is shown below.

FullSibshipIndex	Prob(Inc.)	Prob(Exc.)	Member-1	Member-2	Member-3	Member-4
1	0.8560	0.8435	O1	O2		
2	0.8615	0.8123	O3	O4		
3	0.9940	0.9090	O5	O6		
4	1.0000	0.9534	O7	O8	O21	O39
5	0.9350	0.9123	O9	O10	O13	O14
6	0.7941	0.6521	O11	O12	O19	O27
7	0.9953	0.9523	O15	O16	O20	

The 2nd column gives the probability that all individuals (listed from column 4 on) of a given fullsib family in the best configuration are fullsibs. Essentially this Inc.Prob. shows to which extent the full sib family is splitable. The lower this probability is, the higher is the likelihood that the family can be split into 2 or more families. Therefore, the probability of a best full-sib family containing a single individual is always 1, because it is not splitable.

The 3rd column gives the probability that all individuals (listed from column 4 on) of a given fullsib family in the best configuration are fullsibs, and no other individuals are fullsibs with this fullsib family. Therefore, this Prob(Exc.) is always smaller than or equal to Prob(Inc.) of the same family.

Comparing Prob(Inc.) with Prob(Exc.) sheds light on the possible sibship structure. If a sibship has a high Prob(Inc.) but a low Prob(Exc.), then it means the best full sibship is probably true, but incomplete (the sibship might be split).

12.8 Offspring genotypes

Given an offspring's observed genotypes and the genotyping error rates at each locus, Colony infers the underlying genotypes of the offspring conditional on the inferred relationships (family structures, or the genotypes of other individuals related to the offspring). The inferred offspring genotypes are contained in a file named "*.OffGenotype", and can be loaded by clicking "View Results" → "Inferred Offspring Genotypes" in Windows. The 1st and 2nd columns show the offspring ID and the marker ID. For a codominant diploid locus, the 3rd and 4th columns show the observed 2 alleles of the offspring at the locus, the 5th column shows whether a genotyping error is detected (=1) or not (=0), columns 6~8 show the 1st inference of the 2 alleles and the probability, columns 9~11 show the 2nd inference, … For a dominant locus, the observed genotype takes just 1 column with a value of either 1 (dominant type) or 2 (recessive type) or 0 (missing genotype). For each offspring-locus combination, a maximum of 5 inferences are given. Part of an example file is shown below.

OffspringID	MarkerID	ObsAllele1	ObsAllele2	TypeError	Est1-Allele1	Est1-Allele2	Prob1	Est2-Allele1	Est2-Allele2	Prob2
O1	mk1	11	11	0	11	11	1.0000	5	7	0.0000
O2	mk1	11	11	0	11	11	1.0000	5	7	0.0000
O3	mk1	1	11	0	1	11	0.9999	9	11	0.0001
O4	mk1	9	11	0	9	11	0.9998	1	11	0.0002
O5	mk1	12	12	0	12	12	0.9999	11	12	0.0001
O6	mk1	11	12	0	11	12	0.9999	12	12	0.0001
O7	mk1	1	12	0	1	12	1.0000	1	7	0.0000
O8	mk1	1	12	0	1	12	1.0000	1	7	0.0000
O9	mk1	1	4	0	1	4	1.0000	4	5	0.0000

Note that the offspring genotypes are not inferred when they are observed at loci that have no genotyping errors, because in such a case the observed genotypes are, by definition, always correct. They are also not inferred when the pair-likelihood score method is used and the reconstructed cluster of family is very large that it is difficult to calculate the likelihood. Also note that a genotyping error is

detected (error flag =1) when the observed genotype has a probability < 0.05 (i.e. significance level =0.05). Otherwise (error flag =0), the observed genotype has a probability > 0.05.
Note also that clone mates, if inferred, always have identical estimated genotypes at each locus.

12.9 Father genotypes

The genotypes at each locus of each inferred father (whether included in or excluded from the candidates) in the best configuration are also inferred by Colony, given in a file named "*.DadGenotype". The inference is made in a likelihood framework by calculating the probability of observing the genotypes of all of the offspring of an inferred father. The results can be loaded by clicking "View Results" → "Inferred Father Genotypes" in Windows. The file has the same structure as "*.OffGenotype", except that offspring ID in the 1st column is replaced by father ID. An unobserved genotype is indicated by "-1 -1". Part of an example file is shown below.

FatherID	MarkerID	ObsAllele1	ObsAllele2	TypeError	Est1-Allele1	Est1-Allele2	Prob1	Est2-Allele1	Est2-Allele2	Prob2
*3	mk1	-1	-1	0	1	5	0.6990	7	12	0.3007
*9	mk1	-1	-1	0	2	7	0.6262	6	6	0.2239
*4	mk1	-1	-1	0	1	5	1.0000	4	5	0.0000
*5	mk1	-1	-1	0	1	6	0.1665	6	10	0.1137
*6	mk1	-1	-1	0	6	10	0.3807	7	10	0.3561
M1	mk1	11	12	1	7	11	0.4992	7	12	0.4992
M2	mk1	4	8	1	4	11	0.5000	8	11	0.5000
*1	mk1	-1	-1	0	1	9	0.7285	11	11	0.1140

For haplodiploid species, the column headed by ObsAllele2 is shown as "NA". Note that father genotypes are not inferred in similar cases as described above for offspring genotypes. Also note that a genotyping error is detected (error flag =1) when the observed genotype has a probability < 0.05 (i.e. significance level =0.05). Otherwise (error flag =0), the observed genotype has a probability > 0.05.

Three points are worth noting.
(1) Only single locus parental genotypes are inferred. No attempt has been made to infer multilocus parental genotypes.
(2) Even for single locus inference, the inferred genotypes are accurate only in some limited cases in which father and mother genotypes can be distinguished. These include haplodiploid species, male and female parents having different numbers of mates, and a parent of one sex having known genotypes. Otherwise, the male and female genotypes are symmetrical and it is impossible to infer the genotypes of each parent accurately. Let us consider an example. Suppose an inferred full sib family contains 4 offspring, who have genotypes {A1,A1}, {A1,A2}, {A1,A3}, {A2,A3} at a locus. When both parents of the family are not in the candidate male and female lists and thus have no known genotypes, they are inferred to be {A1,A2} and {A1,A3}. However, it can be {A1,A2} for the mother and {A1,A3} for the father, or vice versa. The two cases have the same probability that is <=0.5. Therefore, none of the 2 inferences is more reliable than the other. In file "*.DadGenotype", up to 5 inferences are listed for each locus of each inferred father, with corresponding probabilities. Only when the best (the 1st) inference has a posterior probability much larger than that of the second best should we take it seriously. Otherwise, these inferences are just for the user's reference, with weightings indicated by the posterior probabilities.
(3) It is very difficult to infer multilocus genotypes accurately from single locus genotype inferences. Even when single locus genotypes are highly accurately inferred (as indicated by a much higher probability of the best inference than that of the 2nd best), multilocus genotypes still cannot be inferred reliably. Consider an example. Suppose the probability of the best genotype inference for each of a number of L loci is 0.9. Assuming independence among loci, the probability of the best multilocus genotype (which combines the L best single-locus genotypes) inference is 0.9^L, which decreases very rapidly with an increasing value of L. This means that it becomes quickly highly probable that the best inferred multilocus genotype is incorrect at one or more loci as L increases.

(4) For parental genotype combinations, see “13.3 Why the inferred maternal and paternal genotypes are always the same”

12.10 Mother genotypes

The genotypes at each locus of each inferred mother (whether included in or excluded from the candidates) in the best configuration are also inferred by Colony, given in a file named “*.MumGenotype”. The file has the same format as “*.OffGenotype”. The results can be loaded by clicking “View Results” → “Inferred Mother Genotypes” in Windows. The three points mentioned above in 11.9 apply to mother genotype inference. Note that mother genotypes are not inferred in similar cases as described above for offspring genotypes.

12.11 Distribution

The file named “*.Distribution” gives the distribution of 7 interesting quantities, which are the number of paternal families, the number of maternal families, the number of full-sib families, the number of children per father, the number of children per mother, the number of mates per male, and the number of mates per female. These distributions are calculated from the archives of plausible configurations with relatively high likelihood values. An example of the file is shown below. The first column lists the numbers, and the following columns lists the frequencies of these numbers. For example, the frequencies of 14 and 15 paternal families are 0.2004 and 0.5197, respectively.

Number	Fre(#PaternalFamily)	Fre(#MaternalFamily)	Fre(#FullSib-Family)	Fre(#Child/Father)	Fre(#Child/Mather)	Fre(#Mating/Male)	Fre(#Mating/Female)
1	0.0000	0.0000	0.0000	0.0487	0.0487	1.0000	1.0000
2	0.0000	0.0000	0.0000	0.4938	0.4938	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.2238	0.2238	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.2199	0.2199	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0137	0.0137	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

…

The results can be loaded by clicking “View Results” → “Distribution” in Windows.

12.12 Allele frequency

No matter whether Colony is instructed to update allele frequencies or not, it will infer allele frequencies by taking the ML family structure into account when population allele frequencies are unknown. The file containing the refined allele frequencies is named as “*.AlleleFreq”, and can be loaded by clicking “View Results” → “Inferred Allele Frequency” in Windows. In the file, columns 1~4 give the marker ID, the allele ID, the initial allele frequency without accounting for family structure, and the updated allele frequency calculated by taking reconstructed family structure into account. Part of an example file is shown below.

MarkerID	AlleleID	InitialFreq	UpdatedFreq
mk1	1	0.1000	0.0958
mk1	2	0.0600	0.0575
...			
mk2	1	0.0550	0.0558
mk2	2	0.1000	0.0896
...			

12.13 Archives of plausible configurations

During the process of hunting for the best configuration with the maximum likelihood, many configurations with high likelihoods (defined as log likelihood value larger than ML - 10) are generated. These plausible configurations with corresponding log likelihood values are archived and used on completion to analyze the uncertainties of a given inference (e.g. a full sibship). These archived configurations, listed in likelihood descending order, are in a file named “*.ConfigArchive”. The first archived configuration is the best one with the maximum likelihood. The configurations in the

archives have the same format as the best configuration in file "*.BestConfig". A maximum of 1000 unique configurations are listed in the file. The results can be loaded by clicking "View Results" → "Archived Configuration" in Windows.

12.14 SubStructure probability

The probabilities of a full-sib offspring dyad, a half-sib offspring dyad, a full-sib offspring family, a mother-offspring dyad (maternity), and a father-offspring dyad (paternity) have already calculated. Using the archived configurations, we can also find the probability of a particular substructure consisting of 2 or more offspring and their parents, using the method described in 13.4 below. This operation can be performed by clicking "View Results" → "SubStructure probability".

12.15 Intermediate results

Most of the intermediate summary results shown on the monitor during the simulated annealing process are also stored in a file called "*.Midresult", and can be loaded by clicking "View Results" → "Intermediate Result" in Windows. The column heads are "Date Time Run Tmr NumIterate CrLogL Cr#F1 Cr#F2 Cr#F3 Cr#FS CrHSPair CrFSPair Cr#AssC1 Cr#AssC2 Cr#AssP1 Cr#AssP2 BtLogL Bt#F1 Bt#F2 Bt#F3 Bt#FS BtHSPair BtFSPair Bt#AssC1 Bt#AssC2 Bt#AssP1 Bt#AssP2". Column "Date" shows the day and month of the year, "Time" gives the seconds, minutes, and hour of the time, "Run" gives the replicate run index, "Tmr" gives the temperature reduction steps within the replicate run, "NumIterate" gives the number of iterates within the replicate run, "CrLogL" gives the likelihood of the current configuration, and the other columns are explained in ***7.1 Start the run.*** "BtLogL" gives the likelihood of the best configuration so far, and the following columns are similar to those for the current configuration.

The results in file "*.MidResult" can be visualized by an external program (such as Excel) and used to monitor the changes of a quantity (e.g. number of fullsib dyads) within a replicate run, and compare these changes between different replicate runs. A plot of the changes in "CrLogL" among 3 replicate runs (lines in red, green and blue) for a hypothetical dataset is shown in Figure 14.

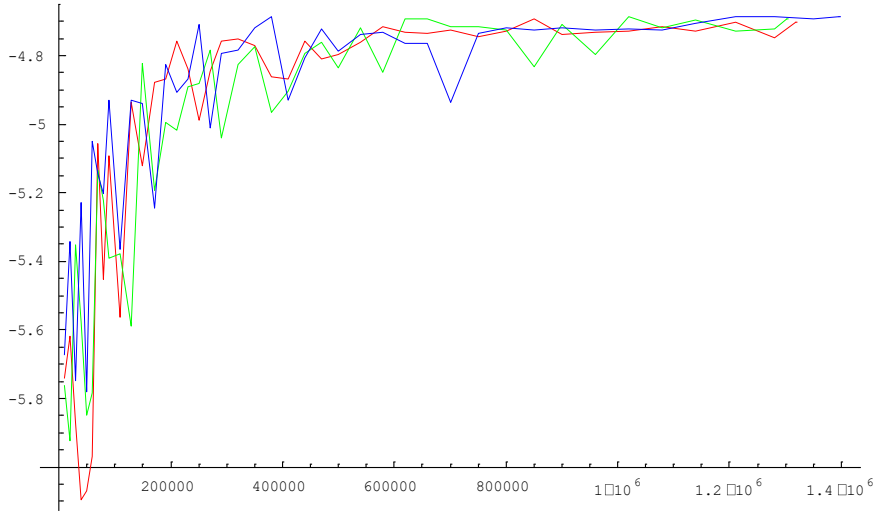


Figure 14: Changes of Log likelihood (*y* axis) as a function of the number of iterates (*x* axis) in 3 replicate runs (red, green and blue) of a hypothetical dataset.

12.16 Duplicated individual dyads from the pairwise approaches

When duplicates (or clone mates) are instructed to be inferred in the parameter setting, they are inferred by the pairwise approach (see below) and reported in an output file "*.PairwiseCloneDyad".

12.17 Fullsib dyads from the pairwise approaches

A pure pairwise likelihood method (PPL) is also implemented for estimating the sibship among offspring. In brief, a pair of offspring is considered in isolation for three possible candidate relationships, fullsib, halfsib, nonsib, and an additional relationship of identical twins (or duplicates, clone mates) if it is instructed to be inferred. The likelihood is calculated for each of the these relationships and the relationship with the maximum likelihood is returned as the best estimate. The method implemented in Colony uses the same genotyping error model (Wang 2004) and allows for haplodiploid species, missing genotypes, codominant and dominant markers.

The results are stored in a file named as "*.PairwiseFullSibDyad", and can be loaded by clicking "View Results" → "Fullsib Dyad (Pairwise Method)" in Windows GUI. On each line, the IDs of each fullsib dyad are listed. Note that for a sib pair of individuals A and B, only one of the two possible unordered dyads of {A, B} and {B, A} is listed.

12.18 Halfsib dyads from the pairwise approaches

For a brief description of the method, see the section above.
The results are stored in a file named as "*.PairwiseHalfSibDyad", can be loaded by clicking "View Results" → "Halfsib Dyad (Pairwise Method)" in Windows. On each line, the IDs of each halfsib dyad are listed. Note that for a sib pair of individuals A and B, only one of the two possible unordered dyads of {A, B} and {B, A} is listed.

12.19 Paternity from the pairwise approaches

In the pairwise approaches to assigning parentage to a single offspring, I follow closely the procedure of Marshall et al. (1998) in using the Δ statistic for resolving parentage with a certain level of confidence. In brief, Δ statistic is defined as the difference in LOD score between the most-likely candidate (as parent) and the second most-likely candidate. Its distribution is determined by analyzing simulated data mimicking the empirical dataset (Marshall et al. 1998). Given the distribution and a certain level of confidence (say, 95%), one can determine a threshold Δ value which can then be used in parentage assignments of the empirical dataset.

The method implemented in Colony allows for diploid and haplodiploid species, for codominant and dominant markers, and genotyping errors. Any known maternity is also utilized to infer the paternity of a given offspring.

The results are stored in a file named as "*.PairwisePaternity", can be loaded by clicking "View Results" → "Paternity (Pairwise Method)" in Windows. On each line, the IDs of each offspring and its inferred father are listed. Note that the level of confidence used in the analysis is 80% and 95%.

12.20 Maternity from the pairwise approaches

Similar to paternity, maternity is also obtained by the pairwise approach described in 11.19 above.

The results are stored in a file named as "*.PairwiseMaternity", can be loaded by clicking "View Results" → "Maternity (Pairwise Method)" in Windows. On each line, the IDs of each offspring and its inferred mother are listed. Note that the level of confidence used in the analysis is 95%.

12.21 Ne estimated from sibship assignments

Within a sample of individuals taken at random (with respect to kin) from a population, the frequencies of full and half sib dyads can be used to estimate the current effective size of the population. For details of the method, see the reference below

Wang J 2009. A new method for estimating effective population sizes from a single sample of multilocus genotypes. Molecular Ecology 18: 2148-2164.

The logic behind the method is simple. A small population (small Ne) will result in a high proportion of sibs. The smaller the Ne is, the higher is the probability that 2 individuals drawn at random from the same cohort of a population are sibs sharing the same father, or mother or both.

Two sets of Ne estimates from the sibship assignment method are provided. One set was obtained assuming a random mating population so that deviation from Hardy Weinberg equilibrium is negligible (Alpha = 0). The other set was obtained by using the Alpha value estimated from the genotype data of the current sample. When the number of sampled individuals is small, or the number of markers is small, the estimate of Alpha may be inaccurate and thus may lead to less accurate estimates of Ne. On the other hand, when there is substantial deviation from Hardy Weinberg equilibrium but is ignored (assuming Alpha = 0), Ne is also biased. The users may have some prior knowledge of Alpha of their populations. Using this knowledge together with the sample size considerations, one can determine which set of estimates are more appropriate.

The estimates of Ne from the sibship assignment method and the heterozygote excess method are stored in a file named as "*.Ne". The file can be loaded to view by clicking "View Results" → "Ne from Sibship Assignment" in Windows. For comparison, an estimate of Ne from the same data using the heterozygosity excess method is also given. The 95% confidence intervals are obtained from bootstrapping.

It should be noted that, like any other Ne estimation methods, the sibship assignment method makes several assumptions which could be violated in reality. The most critical assumption is that a sample of individuals is taken at random (with respect to kinship) from a single cohort of the population. If there are several cohorts in the sample, then it is possible that some sampled individuals are actually parents of some other sampled individuals. Without knowing the parent-offspring relationship, it is very difficult to infer full-sibship reliably as PO and FS dyads are very similar IBD (identity by descent) sharing. Colony (and any other relationship inference program) is limited in power in distinguishing the 2 relationships when sibship size is small and marker information is scarce. As a result, some PO relationships may be mistakenly inferred to be FS, increasing the estimate of FS frequency and decreasing the estimate of Ne. Although Colony always provides a Ne estimate from sibship assignment results, it is up to the user's discretion whether to discard or use the estimate.

12.22 Estimates of inbreeding

When the inbreeding model is adopted, Colony estimates the average inbreeding coefficient (and selfing rate for monoecious) iteratively with other parameters. The point estimates together with 95% confidence interval of inbreeding (and selfing rate for monoecious) are in the output file "*.Inbreeding", which can be loaded to view by clicking "View Results" → "Inbreeding" in Windows. When inbreeding is no allowed, no output for inbreeding is available.

12.23 Estimates of selfing

For monoecious species under the inbreeding model, Colony estimates the probability that each offspring comes from selfing fertilization. The results are in the output file "*.Selfer", which can be loaded to view by clicking "View Results" → "Selfing Individual" in Windows. For dioecious species or when inbreeding is no allowed, no output for selfing is available.

12.24 Inferred mistyping error rates

Given the user defined initial dropout and false allele rates of each marker locus, Colony infers the 1 (sibship only) or 2 (sibship and parentage) generation pedigree of the sampled individuals. Conditional on the best reconstructed pedigree, Colony re-estimates (refines) dropout and false allele rates of each marker locus. It also gives the upper (CI95UB) and lower (CI95LB) bounds of the 95% confidence interval of each estimated error rate. These results are listed in a file called "*.ErrorRate". The file can be loaded to view by clicking "View Results" → "Mistyping Rate" in Windows. Each row has 9

columns, giving in ascending order the MarkerID, the initial value, the inferred value, the inferred CI95LB and the inferred CI95UB of dropout rate, the initial value, the inferred value, the inferred CI95LB and the inferred CI95UB of false allele rate. Note the estimates are given only to loci with any non-zero rates of initial dropout rate, false allele rate, or both. No estimates are calculated and shown for loci with zero initial rates of both dropout and false alleles, or when the pairwise likelihood score method is used.

12.25 Inferred parent pairs

When both male and female candidate parent samples are included in an analysis, the inferred parent pairs for each offspring with their probabilities are obtained from the full likelihood method, and are listed in a file called "*.ParentPair". The file can be loaded to view by clicking "View Results" → "ParentPair" in Windows. Each row has 4 columns, giving the offspringID, the inferred fatherID, the inferred motherID, and the probability of the inferred parent pair. If an offspring has several inferred parent pairs, they are listed in the order of the probabilities. Any inferred parent pair that has a probability smaller than 0.01 will not be listed. If paternity (maternity) is not assigned to any candidate, the fatherID (motherID) is indicated by "*" ("#"). For monoecious, unassigned parentage is indicated by "#".

When any or both male and female candidate parent samples are missing, then the file "*.ParentPair" is also missing.

12.26 Best(ML) clones

Clones inferred by Colony using FL or PLS method are listed in a file named "*. BestClone". The probabilities are calculated similarly to those of a full sib family in "*.BestFSFamily".

12.27 Output files from the simulation program

All file names have suffix "_*i*" to indicate outputs from simulation replicate *i*, as explained above.

(1) Offspring IDs and genotypes: The data are in a file named "**OffspringGenotype_i.txt**". The 1st line is the header, and the *i*th (*i*>1) line gives the ID and genotypes of the *i*-1th offspring. Each offspring takes one row, with the first column being the offspring ID, and 2nd and 3rd columns being the genotype at locus 1, … The offspring ID implies the simulated relationships among all individuals in the simulated dataset. It consists of 3 parts. The first part is the offspring's father ID, which starts with letter "M" (indicating male parent), followed by a number *x*, which is the father's unique rank order. If the offspring is produced by male *i* in the *l*th mating matrix, then $x=(l-1)m+i$, where *m* is the number of males in a single mating matrix. The second part is the offspring's mother ID, which is similarly coded as the first part, except it starts with letter "F" (indicating female parent). The 3rd part is the offspring ID, which starts with letter "C" (indicating child), followed by a number *z*, which is the rank order of the offspring within the fullsib family to which the offspring belongs. For example, an offspring ID of "M2F3C4" indicates that the offspring is the 4th child from the mating between the 2nd father and the 3rd mother (each is continuously counted across all mating matrices). Two (or more) offspring are full siblings if they have the same father ID and mother ID, are paternal (maternal) half siblings if they have the same father (mother) ID but different mother (father) IDs, are non-sibs if they have different father IDs and different mother IDs. Parentage and non-parentage relationships between an offspring and a candidate parent are similarly recognized.

Note that if a locus is dominant, then its genotype takes just one column, with 1 and 2 denoting the dominant (band presence) and recessive (band absence) respectively. Also note that for monoecious species, both the mother ID and the father ID of an offspring starts with letter "M".

(2) Allele frequencies: If "Allele frequency" in section 5.2 is designated as known, then the allele frequencies used in simulating genotype data will be output to a file "**AlleleFrequency_i.txt**", and also

to Colony's input file "Colony2_i.dat". The data format in "AlleleFrequency_i.txt" is similar to that in (3) *Allele frequency* of section 3.3, except for an additional header line.

(3) Marker types and mistyping rates: The file is named "**MarkerTypeErrorRate_i.txt**", and contains the information for the ID, type (codominant or dominant), dropout rate, and other error rate of each marker. The data format is similar to that in (2) *Marker type and error rate* of section 3.3, except for an additional header line.

(4) Male IDs and genotypes: If there is any known or sampled male parent, or the number of male candidates is greater than zero, then a file called "**MaleGenotype_i.txt**" will be generated in the project folder. It is similar in format to "OffspringGenotype_i.txt", with the first column giving the candidate male ID, and the rest of the columns giving genotypes. For a true father designated as known, its ID starts with "M" followed by its unique rank order, as described above. For the *i*th unrelated candidate male, its ID starts with "m" followed by number *i*. Known male parents (if any) are listed first, followed by unrelated candidate males. Note for haploDiploid species, males are haploid and each single locus genotype takes just 1 column, not 2 as is true for diploid.

(5) Female IDs and genotypes: Similar to (4) above, and the file name is "**FemaleGenotype_i.txt**". Note for monoecious, female ID starts with "M" (instead of "F" as for dioecious) or "m" (instead of "f" as for dioecious). In fact, males and females are exactly the same for monoecious.

(6) Known paternity: If any fathers are known in the input data for the simulation project, then a file called "**KnownPaternity_i.txt**" will be generated in the project folder. In the file, each row (except for the first row which is the header) starts with the ID of a known father, followed by IDs of all of its offspring whose fathers are defined as known.

(7) Known maternity: Similar to Known paternity above, and the file name is "**KnownMaternity_i.txt**".

(8) True father IDs and genotypes: The simulated true genotypes of each father (dad) with its ID (coded as in (1) above) as the first column are included in a file called "**TrueDadGenotype_i.txt**".

(9) True mother IDs and genotypes: Similar to (8) above, and the file name is "**TrueMumGenotype_i.txt**".

(10) True offspring IDs and genotypes: The simulated true genotypes of each offspring with its ID (coded as in (1) above) as the first column are included in a file called "**TrueOffspringGenotype_i.txt**". For each offspring, any difference between the true genotype and the observed genotype in (1) is caused by mistypings or missing data. Where a true allele is different from an observed allele at any locus of any offspring, the true allele is indicated by a sign "-". When loaded into the DataGridView of Colony GUI, the true allele which is different from the observed allele is highlighted in red with a yellow background.

(11) True male IDs and genotypes: The same as (4), except the true rather than observed genotypes are included for each candidate male parent in a file called "**TrueMaleGenotype_i.txt**". Where a true allele is different from an observed allele at any locus of any candidate male, the true allele is indicated by a sign "-". When loaded into the DataGridView of Colony GUI, the true allele which is different from the observed allele is highlighted in red with a yellow background.

(12) True female IDs and genotypes: Similar to (11) above, and the file name is "**TrueFemaleGenotype_i.txt**".

(13) All data for Colony: The complete input file for Colony, "**Colony2_i.dat**", is generated in the project folder.

(14) All data for simulation: The complete input file, "Input3.Par", contains all information needed for running simulation program Simu2.exe.

***(15) Accuracy*:** The file name is "*_i.accuracy". The file lists the counts of (X|Y), where X and Y are inferred and true fullsib, halfsib and nonsib pairs (or inferred and true parentage and non-parentage pairs) in simulation replicate *i*. For monoecious, it also lists the counts of (X|Y), where X and Y are inferred and true selfing and outcrossing individuals. Accuracy is listed for two methods, the FL, PLS or FPLS method as chosen in setting up the simulation project, and the pure pairwise method.

***(16) Total accuracy*:** The file name is "*_0.accuracy". It gives the overall accuracy across replicates.

Note output files in (1-12) are comma delimitated pure text files and have headers as the first line, and can be imported into Excel directly. They can also be imported into R using command 'data<-read.csv("c:\\temp\\data.txt",header=T)', where "temp" should be the path and "data.txt" should be any output file name in (1-12). These files can be loaded and viewed within the Colony environment once the simulation has completed.

13. Example datasets

Four expirical and 2 simulation example datasets are included in the software package to illustrate how to use Colony program and how to interpret the analysis results. It is suggested that at least one example dataset is run before the user prepares and analyzes his/her own dataset. These datasets are found in folders "Example Dataset X" (where X=1~4) and "Simulation Dataset Y" (where Y=1~2). Each folder contains several input files with self-explanatory names. To setup a new project using any of these datasets, follow the steps described above.

Four complete projects using the four example datasets are also included in the package. The project names are "exampleX" (where X=1~4). To show the analysis results of each project, just load the project into Colony. Once it is loaded, you can also run the dataset.

13.1 Empirical example 1 : A simulated dataset

The dataset is generated by simulations. It is a small dataset used to show the data format required and the output given by Colony. It contains an OFS sample of 40 offspring, a CMS sample of 30 candidate males and a CFS sample 30 candidate females. Each diploid individual in the samples is genotyped for 5 codominant loci. Both males and females are assumed to be polygamous. The genotype data of the 3 sub-samples are analyzed to infer sibships within the OFS sample, and simultaneously to infer the paternity and maternity of the offspring using the candidate male and female samples. The dataset, in a folder named "Example Dataset 1", has the following files.

13.1.1 Data files

(1) *MarkerTypeErrorRate.txt* : It contains the ID, type (codominant or dominant), allelic dropout rate and other error rate at each locus. This marker locus order is used consistently throughout the project, including all input and output files. The file reads like this,

```
   mk1      mk2      mk3      mk4      mk5
     0        0        0        0        0
0.0000   0.0000   0.0000   0.0000   0.0000
0.0001   0.0001   0.0001   0.0001   0.0001
```

The first column gives the ID/name (row 1), type (codominant/dominant=0/1, row 2), allelic dropout rate (row 3) and other error rate (row 4) of the first marker locus.

(2) *AlleleFrequency.txt* : It contains the allele IDs and frequencies at each locus. It reads like this,

```
     1       2       3       4       5       6       7       8       9      10      11      12
0.1000  0.0600  0.0350  0.0700  0.0700  0.1400  0.1500  0.0550  0.0550  0.0250  0.1350  0.1050
     1       2       3       4       5       6       7       8       9      10      11      12      13
0.0550  0.1000  0.0950  0.0400  0.0300  0.0550  0.0600  0.0950  0.1050  0.0800  0.1250  0.0900  0.0700
     1       2       3       4       5       6       7       8       9      10      11      12      13      14
0.0350  0.0350  0.1150  0.0900  0.0550  0.1100  0.0800  0.0200  0.0400  0.1150  0.0550  0.1000  0.1100  0.0400
     1       2       3       4       5       6       7       8       9      10      11      12      13      14      15
0.0800  0.0650  0.0600  0.0300  0.1250  0.0800  0.0500  0.0600  0.0900  0.0650  0.1250  0.0700  0.0500  0.0200  0.0300
     1       2       3       4       5       6       7       8       9      10      11      12      13      14      15      16
0.0900  0.1500  0.0200  0.0500  0.0550  0.0750  0.0850  0.0650  0.0600  0.1000  0.0400  0.0550  0.0300  0.0400  0.0200  0.0650
```

The first 2 rows give the IDs and corresponding frequencies of all alleles at the first locus, …, the last 2 rows give the IDs and corresponding frequencies of all alleles at the last locus. This data file is optional. When you do not have extra information about the population allele frequencies, this file is omitted and Colony can use the focal sample for relationship inference as well as allele frequency estimation.

(3) *OffspringGenotype.txt*: It contains offspring IDs (in the first column) and genotypes (from second column on, columns 2 & 3 give the genotype at the first locus, 4 & 5 at the second locus, …). The marker loci follow the same order as before (and below). It reads like this,

```
  O1   11   11    8   11   11    3    9    2    1    2
  O2   11   11    8    3   11    3    9   12    1    1
  …… O3~O39 are omitted
 O40   12    2    3    3    6   10    5    7    7    6
```

(4) *MaleGenotype.txt*: It contains candidate male IDs and genotypes, following the same order and format as that in the offspring genotype file above. It reads like this,

```
 M1          11   12   11    3    3   12    2   12    1    2
 M2           4    8    3    4   12   13    5    7   10    4
 …… M3~M29 are omitted
 M30          9   12    2    6   10   10    6   13   10   11
```

The probability that a dad is included in the male candidate pool is assumed to be 0.5.

(5) *FemaleGenotype.txt*: Similar to the MaleGenotype.txt file, it contains candidate female IDs and genotypes. It reads like this,

```
 F1          11   12   11    3    3   12    2   12    1    2
 F2           4    8    3    4   12   13    5    7   10    4
 …… F3~F39 are omitted
 F30          9   12    2    6   10   10    6   13   10   11
```

The probability that a mum is included in the female candidate pool is assumed to be 0.5.

(6) *KnownPaternity.txt*: It contains known paternity/paternal sibship data. It reads like this,

```
M1 O33  O34
M2  O1
0  O23  O25
0   O7  O21  O39
```

For example, the first row shows that candidate male M1 has 2 known offspring, O33 and O34, while the 3rd row shows that offspring O23 and O25 have the same father who is unknown (may or may not be included in CMS).

(7) *KnownMaternity.txt*: Similar to *KnownPaternity.txt*, the file contains known maternity/maternal sibship data. It reads like this,

```
  0  O15  O20
  0  O11  O12  O19  O27
```

(8) *ExcludedPaternity.txt*: It contains known excluded paternity data. It reads like this,

```
  O31 M3 M5 M7
   O1 M9
```

For example, the first row shows that candidate males M3, M5 and M7 are excluded as the father of offspring O31.

(9) *ExcludedMaternity.txt*: Similar to *ExcludedPaternity.txt*, the file contains known excluded maternity data. It reads like this,

```
  O31 F3 F5 F7
   O1 F9
```

The above 9 files should be loaded into the project in Windows version, and are included in the input file "Colony2.dat" (below) in other platforms.

(10) *Colony2.dat* : This is the complete input file for Colony program in other platforms. It reads like this (anything after ! is just a note),

```
example1        !Dataset name
example1        !Output file name
40              ! Number of offspring in the sample
5               ! Number of loci
1234            ! Seed for random number generator
0               ! 0/1=Not updating/updating allele frequency
2               ! 2/1=Dioecious/Monoecious species
0               ! 0/1=Diploid species/HaploDiploid species
0  0            ! 0/1=Polygamy/Monogamy for males & females
0               ! 0/1=Clone inference =No/Yes
1               ! 0/1=Scale full sibship=No/Yes
0               ! 0/1/2/3=No/Weak/Medium/Strong sibship prior; 4=optimal sibship prior with known Ne
1               ! 0/1=Unknown/Known population allele frequency
12 13 14 15 16  !Number of alleles per locus

    1      2      3      4      5      6      7      8      9     10     11     12  !Allele IDs at 1st locus
0.1000 0.0600 0.0350 0.0700 0.0700 0.1400 0.1500 0.0550 0.0550 0.0250 0.1350 0.1050  !Allele freq at 1st locus
    1      2      3      4      5      6      7      8      9     10     11     12     13
0.0550 0.1000 0.0950 0.0400 0.0300 0.0550 0.0600 0.0950 0.1050 0.0800 0.1250 0.0900 0.0700
    1      2      3      4      5      6      7      8      9     10     11     12     13     14
0.0350 0.0350 0.1150 0.0900 0.0550 0.1100 0.0800 0.0200 0.0400 0.1150 0.0550 0.1000 0.1100 0.0400
    1      2      3      4      5      6      7      8      9     10     11     12     13     14     15
0.0800 0.0650 0.0600 0.0300 0.1250 0.0800 0.0500 0.0600 0.0900 0.0650 0.1250 0.0700 0.0500 0.0200 0.0300
    1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16
0.0900 0.1500 0.0200 0.0500 0.0550 0.0750 0.0850 0.0650 0.0600 0.1000 0.0400 0.0550 0.0300 0.0400 0.0200 0.0650

1          ! Number of runs
1          ! Length of run
0          ! 0/1=Monitor method by Iterate#/Time in second
10000      ! Monitor interval in Iterate# / in seconds
0          ! 0/1=No/Yes for run with Windows GUI
1          ! 0/1/2=PairLikelihood score/Fulllikelihood/FPLS
2          ! 0/1/2/3=Low/Medium/high/very high precision with Fulllikelihood

     mk1     mk2     mk3     mk4     mk5  !Marker IDs of locus 1~5
       0       0       0       0       0  !Marker type (0/1=Codominant/Dominant) of locus 1~5
  0.0000  0.0000  0.0000  0.0000  0.0000  !Allelic dropout rate of locus 1~5
  0.0001  0.0001  0.0001  0.0001  0.0001  !OtherError rate oof locus 1~5

   O1  11  11   8  11  11   3   9   2   1   2   !Offspring ID and genotypes of locus 1~5
   ……   ! rows for offspring O2~O39 are as in (3) OffspringGenotype.txt shown above
  O40  12   2   3   3   6  10   5   7   7   6

0.5  0.5             !Prob. the father and mother of an offspring are included in candidates
30   30              !Numbers of candidate males and females
```

```
   M1  11  12  11   3   3  12   2  12   1   2  !Candidate male ID and genotypes of locus 1~5
   ……   ! rows for male M2~M29 are as in (4) MaleGenotype.txt shown above
  M30   9  12   2   6  10  10   6  13  10  11

   F1  11  12  11   3   3  12   2  12   1   2  !Candidate female ID and genotypes of locus 1~5
   ……   ! rows for female F2~F29 are as in (5) FemaleGenotype.txt shown above
  F30   9  12   2   6  10  10   6  13  10  11

3   3       !Number of offspring with known father, exclusion threshold

O33 M1      !Offspring ID and known father ID
O34 M1
O1  M2

0   3       !Number of offspring with known mother, exclusion threshold

2           !Number of known paternal sibships
2  O23 O25  !Sibship size, and member IDs
3  O7 O21 O39

2           !Number of known maternal sibships
2  O15 O20  !Sibship size, and member IDs
4  O11 O12 O19 O27

2               !Number of offspring with known excluded fathers
O31  3 M3 M5 M7 !Offspring ID, number of excluded fathers, and excluded father IDs
O1   1 M9

2               !Number of offspring with known excluded mothers
O31  3 F3 F5 F7 !Offspring ID, number of excluded mothers, and excluded father IDs
O1   1 F9

0               !Number of offspring with known excluded paternal sibships
0               !Number of offspring with known excluded maternal sibships
```

13.1.2 Output of Colony analysis

It takes about 75 minutes to finish running this dataset on a PC with a 2.16 GHz CPU, using the full likelihood method. On completion of running Colony, the following output files are found (the project name is “Example1”).

(1) *Example1.FullSibDyad*

```
OffspringID1         OffspringID2   Probability
          O3                   O4         0.922
          O5                   O6         1.000
……
         O37                  O38         1.000
         O39                  O40         0.071
```

Only fullsib dyads with a probability >=0.001 are listed in the file.

(2) *Example1.HalfSibDyad*

```
OffspringID1         OffspringID2   Probability
          O1                   O2         1.000
          O1                   O5         1.000
... many more dyads omitted...
         O38                  O39         0.071
         O38                  O40         0.999
         O39                  O40         0.663
```

(3) *Example1.Paternity*

```
OffspringID         InferredDad1     ProbDad1
         O1                   M2        1.000
         O9                   M2        1.000
…
```

(4) *Example1.Maternity*

```
OffspringID         InferredMum1     ProbMum1
         O1                   F1        1.000
         O2                   F1        1.000
…
```

(5) *Example1.ParentPair*

```
OffspringID   InferredMum   InferredDad   Probability
O1            M2            F1            0.5124
O1            M2            #             0.4876
O2            *             F1            0.9909
O3            *             #             0.9909
O4            *             #             0.9909
O5            *             F1            0.9999
```

(6) *Example1.BestConfig*

```
OffspringID     FatherID     MotherID  ClusterIndex
         O1           M2           F1             1
         O2           *1           F1             1
         O3           *1           #1             1
         O4           *1           #1             1
         O5           *2           F1             1
……
        O35           M1           #9             1
        O36           M1           #9             1
        O37           *8           #8             1
        O38           *8           #8             1
        O39           *2           #9             1
        O40           *8           #9             1
```

(7) *Example1.BestCluster*

```
ClusterIndex Probability  OffspringID     FatherID     MotherID
           1      0.1247           O1           M2           F1
           1      0.1247           O2           *1           F1
           1      0.1247           O3           *1           #1
           1      0.1247           O4           *1           #1
           1      0.1247           O5           *2           F1
……
           1      0.1247          O38           *8           #8
           1      0.1247          O39           *2           #9
           1      0.1247          O40           *8           #9
```

(8) *Example1.BestFSFamily*

```
FullSibshipIndex Probability     Member-1     Member-2
               1      1.0000           O1
               2      1.0000           O2
               3      0.9223           O3           O4
               4      1.0000           O5           O6
               5      0.5482           O7           O8
               6      1.0000           O9          O10
               7      1.0000          O11          O12
               8      1.0000          O13          O14
               9      1.0000          O15          O16
              10      0.9999          O17          O18
              11      1.0000          O19
```

(9) *Example1.OffGenotype*

Part of the inferred offspring genotypes are given below.

OffspringID	MarkerID	Obs.Allele1	Obs.Allele2	TypeError	InferType1-1	InferType1-2	Prob1	InferType2-1	InferType2-2	Prob2
O1	mk1	11	11	0	11	11	1.0000	5	7	0.0000
O2	mk1	11	11	0	11	11	1.0000	5	7	0.0000
O3	mk1	1	11	0	1	11	1.0000	9	11	0.0000
O4	mk1	9	11	0	9	11	0.9997	1	11	0.0003

O5	mk1	12	12	0	12	12	1.0000	11	12	0.0000

(10) *Example1.DadGenotype*
Part of the inferred father genotypes are given below (-1 indicates that the observed genotype is not available).

FatherID	MarkerID	ObsAllele1	ObsAllele2	TypeError	InferType1-1	InferType1-2	Prob1	InferType2-1	InferType2-2	Prob2
M2	mk1	4	8	0	4	8	1.0000	4	11	0.0000
*3	mk1	-1	-1	0	5	10	0.9998	10	10	0.0001
*1	mk1	-1	-1	0	11	11	0.6424	9	11	0.2773
*4	mk1	-1	-1	0	7	7	0.7777	1	7	0.1216
*5	mk1	-1	-1	0	6	7	0.8036	2	7	0.1963
*7	mk1	-1	-1	0	11	11	0.5707	2	2	0.2509
M1	mk1	11	12	1	7	11	0.4996	7	12	0.4996
*8	mk1	-1	-1	0	12	12	0.8000	7	12	0.1000

(11) *Example1.MumGenotype*
Similar to *Example1.DadGenotype.*

(12) *Example1.Distribution*

Number	Fre(#PaternalFamily)	Fre(#MaternalFamily)	Fre(#FullSib-Family)	Fre(#Child/Father)	Fre(#Child/Mather)	Fre(#Mating/Male)	Fre(#Mating/Female)
1	0.0000	0.0000	0.0000	0.0279	0.0690	0.0617	0.2591
2	0.0000	0.0000	0.0000	0.0345	0.2068	0.4934	0.4008
3	0.0000	0.0000	0.0000	0.2063	0.1139	0.3161	0.2488
4	0.0000	0.0000	0.0000	0.4005	0.3956	0.1258	0.0657
5	0.0000	0.0000	0.0000	0.1540	0.1234	0.0030	0.0256
6	0.0000	0.0000	0.0000	0.1476	0.0119	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0292	0.0794	0.0000	0.0000
8	0.0105	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.3761	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.6133	0.0736	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0001	0.8856	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0406	0.0000	0.0000	0.0000	0.0000	0.0000

(13) *Example1.AlleleFreq*
This file is unavailable because the population allele frequencies are known so that they are not re-estimated from the current samples.

(14) *Example1.ConfigArchive*

```
Archived Configurations with corresponding log likelihood
Config#=    1 LogLikelihood=   -2.0223725258E+03
  OffspringID     FatherID     MotherID  ClusterIndex
           O1           M2           F1             1
           O2           *1           F1             1
           O3           *1           #1             1
           O4           *1           #1             1
.......
          O36           M1           #9             1
          O37           *8           #8             1
          O38           *8           #8             1
          O39           *2           #9             1
          O40           *8           #9             1
Config#=    2 LogLikelihood=   -2.0226302084E+03
……
```

(15) *Example1.MidResult*

Date	Time	Run	Tmr	NumIterate	CrLogL	#F1	#F2	#F3	#FS	HSPair	FSPair	#AssgnC1	#AssgnC2	#AssgnP1	#AssgnP2
08/04	40/23/13	1	1	10014	-2.464203E+03	24	22	14	34	35	6	0	2	0	2
08/04	41/23/13	1	1	20063	-2.417492E+03	24	22	14	32	34	8	0	1	0	2
08/04	42/23/13	1	1	29873	-2.430777E+03	25	27	16	36	29	4	2	1	3	2
08/04	43/23/13	1	1	39927	-2.417769E+03	20	24	12	33	42	7	0	0	0	0
.....															
08/04	03/38/14	1	29	9506879	-2.024448E+03	10	11	1	25	106	15	0	2	0	6
08/04	52/40/14	1	29	9749847	-2.023396E+03	9	11	2	23	107	18	0	2	0	6
08/04	21/42/14	1	30	9907375	-2.023396E+03	9	11	2	23	107	18	0	2	0	6

13.2 Empirical example 2 : An ant (Leptothorax acervorum) dataset
13.2.1 Source of the example data set

This example data set is kindly provided by Hammond, Bourke and Bruford, from a study on the mating frequency of an ant species, *Leptothorax acervorum* (Molecular Ecology 10:2719-2728). The dataset consists of 377 ant workers (diploid) sampled from 10 known colonies, each headed by a single monogamous queen (diploid). Males are also monogamous. Therefore, the sampled workers are either fullsibs from the same colony or nonsibs from different colonies. Candidate males and females are not available. These 377 ant workers are genotyped at up to 6 microsatellite loci, which have a number of observed alleles varying between 3 to 22. The genotypes of the sampled workers are used alone in reconstructing the sibships (colonies). The rates of genotyping errors are unknown, and are assumed in the analysis to be 5% for both allelic dropout and other kind of errors at each locus. Use of error rate values varying over several orders gives the same results (Wang 2004). The data files are as follows.

13.2.2 Input files

(1) *MarkerTypeErrorRate.txt* :

```
marker1    marker2   marker3   marker4   marker5   marker6
      0          0         0         0         0         0
   0.05       0.05      0.05      0.05      0.05      0.05
   0.05       0.05      0.05      0.05      0.05      0.05
```

(2) *OffspringGenotype.txt* : The 1st column gives the offspring ID, followed by the observed genotype at each of the 6 loci. Missing genotypes are denoted by "0 0".

```
   1     0   0   135 143   171 171     0   0     0   0   133 139
   2   264 330   135 143   171 171     0   0     0   0   133 139
   3   264 270   135 143   171 203     0   0     0   0   137 139
   4   264 330   133 135   171 203     0   0     0   0   133 139
   5   264 270   133 135   171 203     0   0     0   0   137 139
   …… (Individual 6~376 ommited herein)
 377   250 252   131 145   167 169   115 115   185 187   133 133
```

(3) *Colony2.dat* : This is the complete input file for Colony program in other platforms. Note that several input items are meaningless in some cases but do need values as place holders. For example, in this dataset, there are no candidate males and females so that the probabilities of the father and mother of an offspring being included in candidates are meaningless. However, 2 arbitrary values (in the range 0~1) are needed as input.

```
Lacervorum_SD99  !Dataset name
Example2         !Project (output file) name
377       ! Number of offspring in the sample
6         ! Number of loci
1234      ! Seed for random number generator
0         ! 0/1=Not updating/updating allele frequency
2         ! 2/1=Dioecious/Monoecious species
1         ! 0/1=Diploid species/HaploDiploid species
1  1      ! 0/1=Polygamy/Monogamy for males & females
0         ! 0/1=Clone inference =No/Yes
1         ! 0/1=Scale full sibship=No/Yes
0         ! 0/1/2/3=No/Weak/Medium/Strong sibship prior; 4=optimal sibship prior of known Ne
0         ! 0/1=Unknown/Known population allele frequency
1         ! Number of runs
2         ! Length of run
0         ! 0/1=Monitor method by Iterate#/Time in second
10000     ! Monitor interval in Iterate# / in seconds
0         ! 0/1= No/Yes for run with Windows GUI
1         ! 0/1/2=PairLikelihood score/Fulllikelihood/FPLS
2         ! 0/1/2/3=Low/Medium/high/very high precision with Fulllikelihood


  marker1  marker2  marker3  marker4  marker5  marker6  !Marker IDs of locus 1~6
        0        0        0        0        0        0  !Marker type (0/1=Codominant/Dominant) of locus 1~6
     0.05     0.05     0.05     0.05     0.05     0.05  !Allelic dropout rate of locus 1~6
     0.05     0.05     0.05     0.05     0.05     0.05  ! OtherError rate of locus 1~6
```

```
  1    0    0  135  143  171  171    0    0    0    0  133  139  !Offspring ID & genotypes of locus 1~6
  2  264  330  135  143  171  171    0    0    0    0  133  139
  3  264  270  135  143  171  203    0    0    0    0  137  139
    …… !individual 4~376 are not listed herein
377  250  252  131  145  167  169  115  115  185  187  133  133


0.0  0.0   !Prob. the father and mother of an offspring are included in candidates
0    0     !Numbers of candidate males and females

0  0       !Number of offspring with known father, exclusion threshold
0  0       !Number of offspring with known mother, exclusion threshold
0          !Number of known paternal sibships
0          !Number of known maternal sibships
0          !Number of offspring with known excluded fathers
0          !Number of offspring with known excluded mothers
0          !Number of offspring with known excluded paternal sibships
0          !Number of offspring with known excluded maternal sibships
```

13.2.3 Output files

For a medium length of run using the full likelihood method, it takes about 190 minutes to finish running this dataset on a PC with a 2.16 GHz CPU.

Using the genotype information of sampled workers, the program completely reconstructed the sibships of the sampled 377 workers. All the 377 workers are correctly assigned to the 10 colonies, without a single worker being assigned an incorrect relationship with any other worker. The same 100% successful assignments were obtained with a wide range of possible typing error rates used in the analyses. However, if typing errors are ignored by setting error rate as zero, the best estimate is 14 colonies. The 4 extra colonies are due to typing errors, which result in a split up of several colonies. Indeed, 8 offspring are identified to have single locus genotype errors (mostly at marker 4). These errors can be verified by checking the original data (e.g. worker genotypes at a locus from a single colony displaying 4 or more alleles, which is impossible for a fullsib family of haplo-diploid species).

Colony also infers the parental genotypes at each locus for each reconstructed family. In total, 67 single-locus parental phenotypes are available from this data set (not listed in input data and not used for sibship inference). If these observed phenotypes are completely correct, the numbers of correctly, incorrectly and partially (i.e. only one allele correctly inferred for a queen) recovered single-locus parental genotypes are 63, 2 and 2, respectively. The two incorrectly inferred genotypes are at the same locus of a queen and its mate, and the queen is a homozygote. The posterior probability of these two inferred genotypes is 0.54, and that of the alternatively inferred genotypes, which are in full agreement with observations, is 0.46. The two partially recovered parental genotypes occur in the smallest family containing seven offspring in the sample. Although we cannot check how accurate the inferred genotypes for those parents and loci without observed genotypes, they seem to be quite reliable as indicated by the large posterior probabilities for most loci and families.

With this dataset, we can run several different replicate analyses using independent seeds for the random number generator to see if the same results are obtained. It turns out that all replicates give identical results, indicating the annealing procedure adopted is powerful and well converged. The changes in log likelihood and the number of fullsib pairs as a function of the number of iterates during the annealing process are shown in the figure 15 below for 5 independent runs. All replicate runs converge to the same configuration with the same (maximum) likelihood.

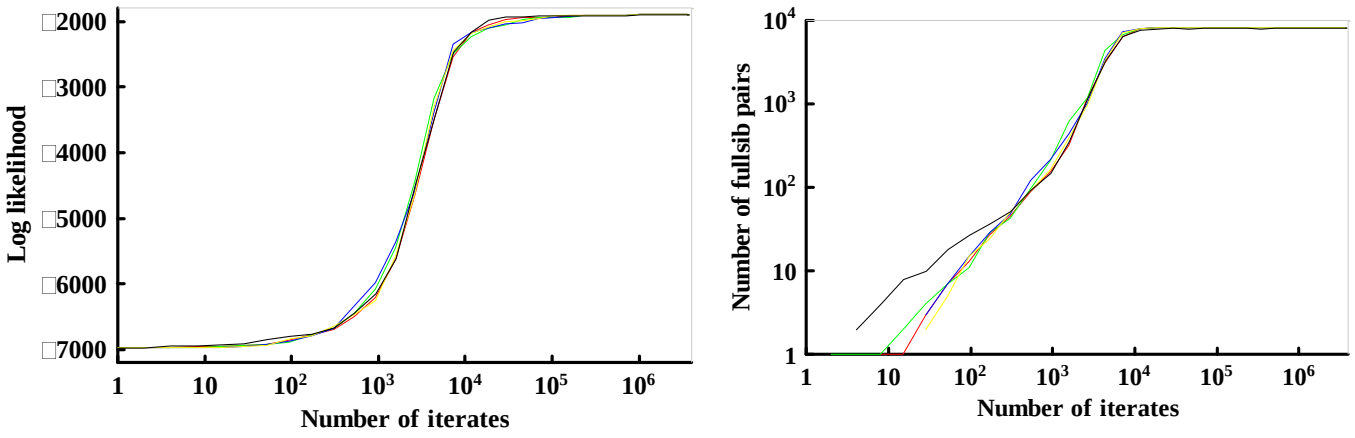


Figure 15: Changes in Log likelihood and number of inferred full-sibships with iterates in 5 independent replicate runs.

13.3 Expirical example 3: CEPH data with known sexes and generations of individuals

13.3.1 Source of the example data set

This is a subset of the CEPH (Centre d'Etude du Polymorphisme Humain) data, maintained by the Fundation Jean Dausset laboratory. The CEPH dataset, in its current version V10 available online (http://www.cephb.fr/cephdb/php/), contains genotypes of individuals from 65 families at 32 356 genetic marker loci. These include 9900 microsatellite markers and 21 480 bi-allelic markers, of which 17 512 are SNPs. Within each family, genotypes are available for the father, mother and a variable number of full-sib children. Some families also have a variable number of grandparents (1–4) genotyped as well.

The subset of CEPH data contains an OFS sample of 343 offspring taken from 59 families. The full sibship size (number of full sibs) in the OFS sample varies from 1 to 12, with the corresponding counts of {1, 4, 4, 8, 13, 9, 7, 4, 4, 2, 2, 1}. The subset also contains 105 candidate males in a CMS sample, and 119 candidate females in a CFS sample. Most of the actual fathers and grand fathers of the 59 families are included in the CMS sample, while a few fathers with genotypes missing at 2 or more loci are excluded. Female candidates are selected similarly. Each individual has genotypes at a maximum of 5 SSR loci. About 5% of the individual-locus genotypes are missing. Among the 5 loci, 1 displays 40 alleles, 1 displays 25 alleles and the remaining three each has 26 alleles. The data are analysed assuming monogamy for both sexes, a probability of 0.5 that the parent of an offspring is included in the candidates, no genotyping errors. Allele frequencies are assumed unknown and calculated by Colony using the 3 samples.

13.3.2 Data files

The following data files are included in the folder "Example Dataset 3".

(1) *MarkerTypeErrorRate.txt*

```
marker1 marker2 marker3 marker4 marker5
      0       0       0       0       0
 0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000  0.0000  0.0000  0.0000  0.0000
```

(2) *OffspringGenotype.txt*: It contains the IDs and genotypes of 343 offspring. The offspring ID (also candidate parent ID) is composed of 19 digits. The first 5 digits give the family ID, digits 6~7 give the individual ID within the family, digits 8~9 and 10~11 give the father ID and mother ID within the family, digits 12~13, 14~15, 16~17, and 18~19 give the paternal grandfather, grandmother, maternal grandfather, and grandmother IDs within the family. Any individual without genotypes (or unsampled) has a 2-digit ID of "00". The file reads like this,

```
 0132803010200000000   3   4   4  11   2   3   2   3   3   7
 0132804010200000000   1   2   4   8   1   3   1   4   5   6
 0132805010200000000   1   2   4   8   1   3   2   3   5   6
 0132806010200000000   3   4   4   8   1   4   1   2   3   7
 0132807010200000000   2   3   0   0   1   4   1   4   3   6
 0132808010200000000   2   3   4  11   2   3   2   3   5   7
 0132809010200000000   3   4   4   8   1   3   1   2   3   7
 0132810010200000000   2   3   4  11   2   4   1   4   5   7
 0132811010200000000   2   3   4   8   1   4   3   4   3   6
 1328103010206070809  15   7   4   9  12  11  11  14   9   7
 ……(Offspring 11~342 are omitted herein)
 0158210010211121314   2  10  15   9   8   7  10   5   6   3
```

(3) *MaleGenotype.txt*: It contains the IDs and genotypes of 106 candidate males. It reads like this,

```
 0132801000000000000   1   3   8  11   3   4   2   4   3   5
 1328101060700000000  15   9   6   4  11   5  11   8   7  11
 1328106000000000000   9   9   6   4  11   3  11  13  11  11
 0132901000000000000  20  26   5  11   3   6   2   6   1   3
 1329101101100000000  18  27   8  11  23  26   1   4   8  11
 ……(Offspring 6~104 are omitted herein)
 0158201111200000000  10   6  10   9  10   7   7   5  10   6
```

The probability that a dad is included in the candidate males is assumed to be 0.5.

(4) *FemaleGenotype.txt*: It contains the IDs and genotypes of 119 candidate females. It reads like this,

```
 0132802000000000000   2   4   4   4   1   2   1   3   6   7
 1328102080900000000  15   7   9   7  12   5  14   3  10   9
 1328107000000000000  15   3   6   8   5   5  11   8   8   7
 1328109000000000000   7  15   9   4  13   5  10   3  10   6
 0132902000000000000  23  24   6   8   6   7   4   5   1   2
 ……(Offspring 6~118 are omitted herein)
 0158212000000000000  11  10  10   5  10   6   7   4  10   5
```

The probability that a mum is included in the candidate females is assumed to be 0.5.

(5) *Colony2.dat* : The single input file for Colony in other platforms looks like this,

```
HumanCEPHKnownSex !Dataset name
example3          !Project (Output file) name
343       ! Number of offspring in the sample
5         ! Number of loci
1234      ! Seed for random number generator
0         ! 0/1=Not updating/updating allele frequency
2         ! 2/1=Dioecious/Monoecious species
0         ! 0/1=Diploid species/HaploDiploid species
1  1      ! 0/1=Polygamy/Monogamy for males & females
0         ! 0/1=Clone inference =No/Yes
1         ! 0/1=Scale full sibship=No/Yes
0         ! 0/1/2/3=No/Weak/Medium/Strong sibship prior; 4=optimal sibship prior with known Ne
0         ! 0/1=Unknown/Known population allele frequency
1         ! Number of runs
2         ! Length of run
0         ! 0/1=Monitor method by Iterate#/Time in second
10000     ! Monitor interval in Iterate# / in seconds
0         ! 0/1=No/Yes for run with Windows GUI
1         ! 0/1/2=PairLikelihood score/Fulllikelihood/FPLS
2         ! 0/1/2/3=Low/Medium/high/very high precision with Fulllikelihood

marker1  marker2  marker3  marker4  marker5  !Marker IDs of locus 1~5
      0        0        0        0        0  !Marker type (0/1=Codominant/Dominant) of locus 1~5
 0.0000   0.0000   0.0000   0.0000   0.0000  !Allelic dropout rate of locus 1~5
```

```
 0.0000   0.0000   0.0000   0.0000   0.0000  !OtherError rate of locus 1~5

0132803010200000000   3   4   4  11   2   3   2   3   3   7  !Offspring ID and genotypes of locus 1~5
0132804010200000000   1   2   4   8   1   3   1   4   5   6
  …… (Offspring 3~342 omitted herein)
0158210010211121314   2  10  15   9   8   7  10   5   6   3


0.5  0.5  !Prob. the father and mother of an offspring are included in candidates
105  119  ! Numbers of candidate males and females

0132801000000000000   1   3   8  11   3   4   2   4   3   5  !Candidate male ID and genotypes of locus 1~5
1328101060700000000  15   9   6   4  11   5  11   8   7  11
   ……
0158201111200000000  10   6  10   9  10   7   7   5  10   6


0132802000000000000   2   4   4   4   1   2   1   3   6   7  !Candidate female ID and genotypes of locus 1~5
1328102080900000000  15   7   9   7  12   5  14   3  10   9
   ……
0158212000000000000  11  10  10   5  10   6   7   4  10   5

0   0      !Number of offspring with known father, exclusion threshold
0   0      !Number of offspring with known mother, exclusion threshold
0          !Number of known paternal sibships
0          !Number of known maternal sibships
0          !Number of offspring with known excluded fathers
0          !Number of offspring with known excluded mothers
0          !Number of offspring with known excluded paternal sibships
0          !Number of offspring with known excluded maternal sibships
```

13.3.3 Output of Colony analysis

For a medium length of run, it takes about 30 minutes to finish running this dataset on a PC with a 2.16 GHz CPU, using the full likelihood method.

On completion of running Colony, the following output files are found, **.FullSibDyad, *.BestCluster, *.BestConfig, *.BestFSFamily, *.Maternity, *.Paternity, *.ConfigArchive, *.DadGenotype, *.MumGenotype, *.OffGenotype, *.AlleleFreq, *.Distribution, *.MidResult*, where "*" is the project name "example3".

In summary, there are in total 998 full-sib dyads (among the 343 offspring), 645 parent-offspring dyads, and 133842 non-fullsib, non-parent-offspring dyads. Colony completely reconstructed the genetic structure of the samples without a single dyad assigned an incorrect relationship. The power comes from the 5 high polymorphic markers and relatively large sibship sizes.

13.4 Empirical example 4 : CEPH data with unknown sexes and generations of individuals

13.4.1 Source of the example data set

In example 3, we assume the generations (whether offspring or parents) and sexes of the individuals are known. In some cases, however, either sex, generation or both are unknown to some or even all of the sampled individuals. Such cases are common for non-invasive sampling (using hairs, feces, etc.) from populations in the wild. The CEPH data are reanalyzed in example 4 to demonstrate that sibship and parentage can still be inferred by Colony when the sexes and generations of all individuals are unknown, although the power is compromised. With an increasing informativeness of marker data and sibship size, the knowledge of sex and generation of the individuals becomes decreasingly less important in the analysis.

The 3 sub-samples (343 offspring, 106 candidate males, 119 candidate females) are pooled into a single sample of 567 individuals which acted as the OFS, CMS, and CFS samples. In the known excluded parentage files, however, we need to exclude an individual as a candidate parent of itself. The same 5 markers as example 3 are used in the analysis.

13.4.2 Data files

The data files are similar to those in example 3, except now OFS, CMS, and CFS are the same containing 567 individuals, and there are two additional files containing excluded candidate parents. The 2 files, named as "ExcludedPaternity.txt" and "ExcludedMaternity.txt", are identical for this example, and read like

```
 0132801000000000000    0132801000000000000
 0132802000000000000    0132802000000000000
……
 0158210010211121314    0158210010211121314
 0158212000000000000    0158212000000000000
```

13.4.3 Output of Colony analysis

For a medium length of run using the full likelihood method, it takes about 380 minutes to finish running this dataset on a PC with a 2.16 GHz CPU.
On completion of running Colony, the following output files are found, **.FullSibDyad*, **.BestCluster*, **.BestConfig*, **.BestFSFamily*, **.Maternity*, **.Paternity*, **.ParentPair*, **.ConfigArchive*, **.DadGenotype*, **.MumGenotype*, **.OffGenotype*, **.AlleleFreq*, **.Distribution*, **.MidResult*, where "*" is the project name "example4".

In summary, there are in total 998 full-sib dyads (among the 343 offspring), 748 parent-offspring dyads, and 158715 non-fullsib, non-parent-offspring dyads. There are more parent-offspring dyads and non-fullsib, non-parent-offspring dyads than example 3 because the 3 samples are lumped and there are some parent-grandparent dyads. Colony completely reconstructed most of the 59 families. Most of the incorrectly reconstructed families are either small, or have grandparents included in the sample. Across families, Colony correctly inferred 987 full-sib dyads, 608 parent-offspring dyads, and 158589 non-fullsib, non-parent-offspring dyads out of totals of 998, 748 and 158715, respectively. It seems that parentage assignment is most badly affected by unknown sexes and generations of the sampled individuals.

13.5 Simulation Dataset 1

The entire input file, Input3.Par, is listed in ***6.2 An example.*** The pieces of information required for setting up a simulation project in Windows GUI are included in a folder named "Simulation Dataset 1". The dataset is about a dioecious species with monogamous males and females, with variable full-sib family sizes and with some parents known.

13.6 Simulation Dataset 2

The dataset is about a monoecious species with polygamous males and females and with selfing.

14 Frequently asked questions (FAQ)

14.1 How to make Colony run faster for my dataset?

There are several factors that determine how fast Colony runs and thus how much time it takes to analyze your data.

(1) *Analysis method*. The 2 analysis methods built in the current version of Colony, the full likelihood (FL) and the pair-likelihood score (PLS) method, incur very different computational intensities. This is especially obvious when the sample size is large, both sexes are polygamous, and the marker information is insufficient. In such a case, a family cluster can become large (involving many offspring in a complex family structure), making the computation of its likelihood very time consuming. With the PLS method, however, the computational time is little affected by family cluster sizes. Therefore, in the worst scenarios for the FL method, PLS can be several orders faster than FL. However, PLS usually

yields less accurate results than FL. With an increasing amount of marker information and an increasing family size in data, the difference in accuracy between the two methods diminishes (simulation results not shown). FPLS runs faster than FL, slower than PLS. It is a combination of the FL and PLS methods. It first screens the configurations by PLS, and only those configurations that have passed the screening are examined for their full likelihood. As a result, a proportion of the configurations are never calculated for the full likelihood, and therefore the program runs faster. The side effect is it risks throwing away good configurations with high likelihood, especially when sibship size is large but marker information is insufficient.

(2) *Sample size*. The larger the sample size, the more possible configurations to consider in searching for the maximum likelihood one. It is mainly the offspring sample size that matters. Candidate male and female sample sizes have minimal effect on the running time. Even in the most simple case of monogamy for both sexes and no candidate male and female samples available (so only full-sibship is inferred), the number of possible configurations increases at roughly an exponential rate (see Figure 16). With more complicated genetic structure involving half-sibship and parentage, the number of configurations increases with sample size at a much faster rate. In general, more possible configurations means more searching time required to find the ML configuration. Colony's full likelihood method has been tested on datasets containing ~2000 offspring (assuming both sexes monogamous) and 20 SSR with no genotyping errors. It takes about a week to finish the run on a PC. However, when there are genotyping errors, Colony runs much slower. To accomplish the run within a reasonable time limit, it may be necessary to split a large sample and analyze each segment separately. The principle in splitting a large sample is that sibship should be kept intact as much as possible. For example, if the individuals are taken from several locations and there is little migration for both sexes between locations, then we can analyze the sample comprising individuals from a single location.

Sample size affects the computational time of both full likelihood and pair-likelihood score method.

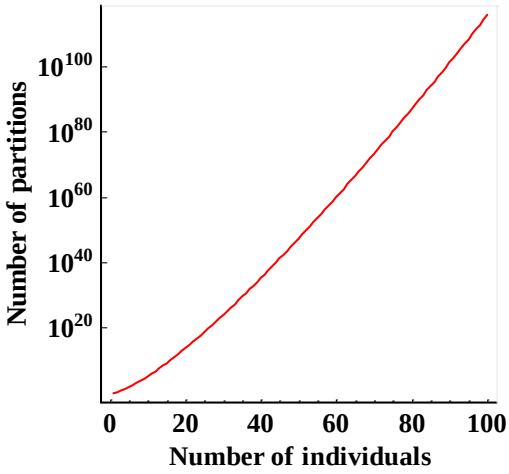


Figure 16: Number of possible configurations (partitions) as a function of the number of individuals in a sample. Individuals in the sample are either full-sibs or non-sibs.

(3) *Genotyping errors*. In calculating the likelihood of any given configuration, different possible parental genotype combinations need to be considered. Without genotyping errors, only feasible parental genotype combinations that are compatible with offspring genotypes are actually considered. However, when genotyping errors are allowed for, many more possible parental genotype combinations are feasible to generate the observed offspring genotypes. As an example, consider a full-sibship

containing an observed genotype {AA}. When there are no genotyping errors at the locus, only 3 parental genotype combinations are feasible, which are {AA} × {AA}, {AA} × {AX}, and {AX} × {AX}, where X represents any allele other than {A}. When genotyping errors are allowed for, 5 parental genotype combinations are feasible, which are, in additional to the 3 above, {AA} × {XX} and {XX} × {XX}. Similarly, if the offspring is observed as {AB}, there are 9 and 21 possible parental genotype combinations when genotyping errors are absent and present, respectively. For more complicated family structure involving many parents, the number of feasible parental genotype combinations increases dramatically by allowing for genotyping errors. Therefore, allowing for typing errors increases computational time substantially, especially when combined with insufficient marker data, large sample size and polygamy for both sexes. Furthermore, the higher the genotyping error rate per locus, the greater the computational time.
To increase computational speed, therefore, any locus that is regarded as highly unlikely to suffer from genotyping errors should be set a zero error rate.

Genotyping errors affect the computational time of the FL method only.

(4) *Mating system*. Allowing for polygamy increases computational load dramatically. First, the configuration space is enlarged by polygamy. Second, a family cluster can become very complicated and big, involving many parents and offspring in the pedigree. Mating system affects the computational time of both full likelihood and pair-likelihood score methods.

(5) *Markers*. The number and polymorphism of markers affect the computational time in a rather complicated way. When marker information is insufficient so that unrelated individuals may have similar genotypes compatible with a sibship, then they are falsely assigned into the same sibship, causing family clusters to become too large and computational load too high. In such a case, increasing the number of loci and marker polymorphism overcomes the problem and reduces computational time. However, when marker information is sufficient so that the actual genetic structure is fully recovered, any additional markers would cause pure computational burden and increase computational time. Markers affect the computational time of the full likelihood method more than that of the pair-likelihood score method.

(6) *Updating allele frequencies*. Colony can refine allele frequency estimates by taking the reconstructed genetic structure into account, and then use better allele frequency estimates to better estimate the genetic structure. This iterative process requires more computational time than the simple case of not updating allele frequencies.

(7) *Monitoring in graph*. When you monitor the computational progress in graph (especially the relationship assignment graph), about 2~10% (depending on the size of OFS) of CPU is devoted to graph drawing and presenting. So if your computer has a single CPU, then computation is slowed down by that proportion. In this case it is better to close or minimize the graphing window when you have finished viewing the graph. The CPU and memory usage of a particular running program can be viewed through Windows “Task Manager”.

(8) *The parameter value for Length of run*. See 13.5 below.

(9) *Choice in prior*. I suggest not using any sibship prior when any sex is monogamous and when little information about average sibship size is available. Otherwise, I suggest using the sibship prior with small average sibship sizes to effectively reduce the chances that unrelated or loosely related individuals being falsely inferred as siblings. The prior discourages a reconstructed family cluster

becoming excessively large because of the inclusion of unrelated or loosely related individuals. By doing so, the prior can speed up computation substantially and usually increase inference accuracy as checked by analyzing simulated data.

14.2 Do I need replicate runs?

Replicate runs can be conducted both within a project and between projects.
Within project replicate runs automatically use different random number seeds to initiate the simulated annealing processes. The results from different runs are combined for both the point estimates (ML estimates) and uncertainty estimates. When the marker information is insufficient and the genetic structure is weak, different runs may not converge to exactly the same configuration. In such a case, slightly more reliable results, especially uncertainty estimates, are obtained by using multiple runs.

Between project replicate runs require the user to set up different projects with the same data and parameter values, but different random number seeds and project names. By comparing the maximum likelihoods and the best configurations from these replicate projects, you can assess whether the program converges for your data set or not. When you have lots of informative markers, or your genetic structure is strong (large sibship size), the program converges reliably. However, when you have few informative markers and your sample's genetic structure is weak and your sample size is large, sometimes different runs give you slightly different answers in the best configuration and its likelihood values. Even in such an unfavorable case, you can still trust the bulk of the results because they are still consistent across runs. The small part of the runs varying across runs is unreliable and is thus better abandoned in downstream analysis.

Ideally, you can set up 2 (or more) replicate projects. Within a project, you can set the number of replicate runs to 1 to save time. If you find that these replicate projects give you essentially the same results, then you are assured that you have sufficient marker information to infer the genetic structure reliably. If not, it means either your marker information is insufficient, or your genetic structure is weak, or both. In such a case, you could obtain slightly better results by setting up a project with either a single long run or several replicate runs of medium (or long) length.

14.3 Why the inferred maternal and paternal genotypes are always the same?

Colony infers the paternal and maternal genotypes of each sib family in the best configuration with the maximum likelihood. The inference is conditional on the reconstructed relationship structure, the observed offspring genotypes and the observed genotypes of the assigned candidates (if exist). The genotypes are inferred for each parent separately, not for the parent combinations. Therefore, for the case of a pure fullsib family without assigned parentage in a diploid species, the inferred paternal and maternal genotypes are always the same. This is simply because, in this case, the definition of paternal and maternal is arbitrary and it is impossible to specify exactly the genotype of a single parent in isolation of the other. For example, father and mother genotype combination {AA, BB} has the same likelihood as {BB, AA} to give the observed offspring genotype {AB}, and thus paternal (and maternal as well) genotype is inferred to be AA and BB with the same marginal probability. If you are interested in the parental genotype combination rather than single parent genotypes, then you need an extra step to assemble single parental genotypes into a parental genotype combination. Let's consider an example.

Suppose a full-sib family containing 2 offspring of the same genotype {AB} at a locus with no genotyping errors. The frequencies of alleles A and B are 0.5 and 0.3, respectively. Both paternal and maternal genotypes are inferred by Colony as {AA}, {BB} and {AB}. The probabilities of {AA}, {BB} and {AB} under HWE are 0.25, 0.09 and 0.15, respectively. There are 9 possible father-mother genotype combinations (denoted by C1~C9), which are {AA, AA}, {AA, BB}, {AA, AB}, {BB, AA},

{BB, BB}, {BB, AB}, {AB, AA}, {AB, BB}, {AB, AB}. The prior probabilities (P1~P9) of C1~C9 are simply the product of the probabilities of the parental genotypes, and are 0.0625, 0.0225, 0.0375, 0.0225, 0.0081, 0.0135, 0.0375, 0.0135, and 0.0225, respectively. Given the parental genotype combination, the probability of obtaining the observed offspring genotypes can be calculated by Mendelian law. For C1~C9, the probabilities (R1~R9) are 0, 1, 0.25, 1, 0, 0.25, 0.25, 0.25, 0.25. To obtain the probability of the combination *n* (1~9), use the formula $\frac{Pn * Rn}{\sum_{i=1}^{9} Pi * Ri}$. For example, the best combinations are {AA, BB} and {BB, AA}, both having the same probability of 0.3.

For haplodiploid species, or a halfsib family, or a parentage assigned fullsib family, the inferred genotypes and probabilities are usually different for paternal and maternal parents. However, again an extra step similar to the example above is required to obtain the parental genotype combination and its probability. With an increasing number of parents involved in an offspring cluster, the parental genotype combination becomes more difficult to infer. At the moment, Colony infers single parent genotypes only. However, if there is enough interest in parental genotype combinations, I will build into Colony this function in the future.

14.4 How do I find the uncertainties of a particular sub-structure?

Colony gives the uncertainty estimates (in probabilities) of the most common relationship structure, such as a pair of individuals (whether full- half-sib dyad, or parent-offspring dyad) or a full-sib family. It is impossible to give the uncertainty estimates of all possible substructures because of they are too numerous even for a small sample. For example, one may be interested in a sub-structure involving 3, 4, 5 … individuals, or even the entire sampled individuals. For the simple case of a particular triad (say, A, B and C), there could be still many sub-structures, such as all 3 sharing a single parent (half-sibs), both parents (full-sibs), no parents (unrelated), A and B sharing both parents and sharing 1 parent with C, … The sub-structures are combinatorial and they are impossible to enumerate even if they involve a small number of individuals (say, 10). However, you can still find out the uncertainty of a particular sub-structure of your interest using the output in a file named *.ConfigArchive.

Consider an example. Suppose that offspring A, B and C are inferred to share a single parent of one sex and to have different parents of the other sex (half-sibs) in the best configuration with the maximum likelihood. Now you are interested in the uncertainty of this point estimate. Open file "*.ConfigArchive" you will find a number of *n* (maximal value is 1000) archived good configurations, **c**={c_1, c_2, …, c_n}, with corresponding log likelihoods. Searching through the *n* configurations, you may find a set of *m* configurations, **d**={d_1, d_2,…, d_m}, that contains the sub-structure. The probability of this sub-structure is then calculated by $\sum_{i=1}^{m} e^{L_i} / (\sum_{i=1}^{n} e^{L_i})$, where L_i is the log likelihood of configuration *i* in set **c** or **d**.

14.5 How do I determine the length of a run?

It is difficult to set the length of a run suitable for a specific dataset. Long runs are safer to ensure the convergence of the algorithm and the best configuration to have the maximum likelihood. However, they are at a cost of your (computer) time and patience. Other thing being the same, the more marker information the dataset has (more loci, more polymorphism) or/and the stronger the genetic structure in the sample, the more likely that the ML configuration is found in a shorter period of time. Because these factors are usually unknown, I suggest the following strategy.

Set up 2 (or more) projects with exactly the same dataset, but different random number seeds. The length of run for each project is set to "short". Run the two projects and compare the results obtained. If they are similar, then it means the program converges even on short runs for this dataset, and you do not need to do medium or long runs. If not, you need to try medium or long runs. Roughly speaking, the time of a medium (long) run is about 10 times that of a short (medium) run.

14.6 Why the genotypes of some offspring or parents are not inferred?

The single locus genotypes of offspring and parents (no matter included in candidate lists or not) are inferred from their phenotypes (if available) and the genotyping error rates given the inferred relationship structure, using Bayes theorem. However, no genotype inferences are provided when the pair-likelihood score (PLS) is adopted which leads to the best configuration has a full likelihood value of zero. This happens when the configuration is the best if pairs of individuals are considered (PLS method) but is incompatible all individuals in it are considered jointly (FL method). To ensure genotype inferences by the PLS method, a locus with no genotyping errors should be set a non-negligible value (say, 0.001).

14.7 File access problems in Windows 7

If Colony is installed in the "Programs Files" folder, then you need first to grant your read and write access to the folder called Colony (which contains program and other files, and the examples) after you have installed Colony on a computer running windows 7. Otherwise, when you run Colony the first time, and try to open an example project, you may get an error message something like "Error in reading project Information". When you close the project, you may get another error message like "System.UnauthorizedAccessException: Access to the path...". To solve the problem, right click the folder "colony", select "properties", then "Security", then "Edit", then allow the users to full control and modify, then click "Apply".

The simple solution is to avoid installing Colony in the "Programs Files" folder, or the "Windows" folder.

14.8 Why the paternity of an offspring is unassigned when a candidate male has a multilocus genotype compatible as the father of the offspring?

The paternity of an offspring may remain unassigned by Colony although a candidate male may have a multilocus genotype completely compatible as the father of the offspring. However, a simple pairwise (parent-offspring) genotype compatibility is not the justification for paternity assignment for at least three reasons. (1) There could be two or more sampled males that have completely compatible genotypes as the father of the offspring. This happens often especially when marker information is insufficient. One may use many polymorphic markers in the analysis. However, the particular offspring, male or both may have a high rate of missing data, so the inference may have to be based on just a few loci. (2) Even when there is only one candidate male that has a compatible genotype, it may still NOT be inferred as the father. This occurs when the offspring is inferred to have one or more paternal siblings and some of them are not happy to have the male as the father (due to genotype incompatibility). More generally, we do not consider genotype compatibility, which is the exclusion based approach. Instead, we calculate the likelihood of various possible relationship (including parentage and sibship) configurations and take the one with the maximum likelihood as the inference. (3) A prior probability of the inclusion of a father in the sampled males is used together with genotype data for paternity assignments. If this prior probability is much smaller than the actual value and marker information is insufficient (so that prior has a large impact on assignments/unassignments), a true father included in the candidate males may still be unassigned.

14.9 Why the paternity (maternity, sibship) assignment probability is so high even when marker information is scarce?

Colony uses simulated annealing algorithm to search for the best configuration with the maximum likelihood. In comparison with other "greedy" but speedy methods, the algorithm is slow but is robust when the global maximum is hidden in many local maxima in the likelihood landscape. In a single run (even for a long run), some good configurations with high likelihood values may still not be found by the algorithm which is designed to find the maximum likelihood configuration rather than all good configurations. As a result, the estimated uncertainties about the best configuration are too low, and the probabilities listed in *.paternity, *.maternity, *.FullSibDyad, *.HalfSibDyad, *.BestFSFamily may be too high to be true. To get more plausible results, multiple runs are needed by setting ***Number of runs*** (in section 3.2) to 5 or more.

14.10 Does Colony infer mating system directly?

No. There is no formal way built in Colony to test directly for the mating system. Let's consider 3 mating systems

A: both sex monogamy; B: one sex polygamy and the other sex monogamy; C: both sex polygamy

For the same genotype data, the maximum likelihood (ML) for C is always not smaller than B, and which in turn is always not smaller than the ML for A. One may naively think of using BIC or AIC to find the best model (mating system). However, it is difficult to count the number of parameters in a mating system (because we are estimating a complex structure rather than simply some parameters), and thus difficult to use the criterion.

When marker information is sufficient (so Colony analysis is highly robust and powerful), then ML(C)=ML(B)=ML(A) when the actual mating system is A, ML(C)=ML(B)>ML(A) when the actual mating system is B, and ML(C)>ML(B)>ML(A) when the actual mating system is C. Comparison of multiple runs of the same data under different mating systems (A, B, C) would indicate the most likely mating system.

When marker information is insufficient, however, such a simple relationship no longer exist. For example, you may observed ML(C)>ML(B)>ML(A) which suggests C being the most plausible mating system, although the actual mating system is A (which means there should be no half siblings). The false inference of some half siblings is due to sampling errors, and does not indicate necessarily that C is the most plausible mating system.

14.11 Why all sampled offspring are inferred to be siblings when allele frequencies are inputted as known?

The option of "Known Allele Frequency" in Colony is intended to deal with the difficult situations of a small offspring sample, or an offspring sample that might contain highly variable sibship sizes. In both cases, the simple allele counting method assuming unrelatedness will yield poor estimates of allele frequencies and thus poor relationship inferences. The latter situation can be dealt with in Colony by allowing the update of allele frequencies, re-estimating them by taking into account of the reconstructed pedigrees. However, the 1st situation can only be dealt with by providing the allele frequencies estimated from a large sample of individuals external to the focal dataset being analysed for relationship. When these frequencies are provided to Colony as known, they will be used in relationship inference and will not be re-estimated using the focal sample.

In the latter situation, it is critically important to ensure that the known allele frequencies do represent the population from which the focal sampled individuals come. Otherwise, Colony may overestimate sibship and parentage. For example, the known allele frequencies may be estimated for a highly differentiated subdivided population, but the focal offspring sample may come from just a single subpopulation. In such a case, obviously the unknown focal sample (or subpopulation) allele frequencies will be differentiated from the known allele frequencies which represent the entire population. When Fst is high (say, >0.1), then all sampled individuals are related by at least that extent and are thus wrongly inferred to be siblings.

14.12 How do I analyse a dataset with few and large families?

Sibship inference in a sample of offspring who are likely to share a single father, or a single mather, or a single pair of parents is difficult, if no other information (such as candidate parent genotypes, allele frequency information, known sibship) is available to Colony. This is because, in this situation, Colony has to estimate both allele frequencies and sibship from the same offspring genotype data. Colony has to estimate allele frequencies by assuming all offspring are unrelated (because their relationships are unknown), and the estimated frequencies are then used in estimating sibship. Obviously, the frequencies of alleles in the shared parent (or parents) at each locus will be grossly overestimated. This will result in sibship being split; Instead of a single shared parent, Colony will return several or many inferred parents. Increasing marker loci does not solve the problem. Updating allele frequencies by accounting for inferred sibship structure also has limited power, because the inferred sibship is always far away from the actual sibship atructure.

There are two solutions. One is to use another sample of individuals who are mostly unrelated in estimating allele frequencies. These externally estimated allele frequiencies are then provided to Colony as known information so that Colony does not have to infer allele frequencies from the offspring genotype data. When no other information is available, the alternative solution is to use the sibship prior by setting a large value (say, = sample size) for the mean sibship size of the relevant sex. For example, if all of the offspring in a sample are likely to share the same mother who may have mated with multiple males, then the mean maternal sibship size should be equal (or not much smaller than) the sample size. Responding to this prior setting, Colony will take 2 actions. One is to calculate allele frequencies from the sample by penalizing the most common alleles. The other is to encourage large maternal sibships.