Pipeline to perform quality control procedures on canine SNP data

---

Author: Marilijn van Rumpt - marilijn@live.nl (2024)

Project description:

To ensure clean and good quality data, quality control steps need to be performed. This command line tool can be used to perform multiple quality control checks: sample call rate, sex check, duplicate sample check, breed check.

This command line utility can be used for the following platforms:

- Embark
- Neogen 170K array (illumina array)
- Neogen 220K array (illumina array)
- Lupa 170K array
- Wisdom
- MyDogDNA
- WGS canfam 3
- WGS canfam 4
- Affymetrix
- Merged datasets
- Other (bed bim fam files from other sources)

Quality control steps performed by this tool

- Check for duplicate sample IDs (always performed)
- Sample call rate check (always performed, except for platform 'merged')
- Removal of bad quality Y SNPs in Neogen 220K and embark
- Optional checks:
    - Sex check
    - Duplicate/relationship check
    - Breed check
        - based on phylogenetic characterization

Command line utility quality_control.sh

Input files should be in same folder as quality_control.sh script. In this folder should also be the convert_files folder.

Usage: bash quality_control.sh [-i|e|a|o|f|m|s|d|b|p|h]

Standard performed quality step for all platforms except for 'merged': sample call rate check to remove bad quality samples. Optional quality steps to perform: sex check, duplicate check, breed check.

- Syntax options:
    - -i <filename.bim> Specify full name of .bim file, use -i, e, a together
    - -e <filename.bed> Specify full name of .bed file
    - -a <filename.fam> Specify full name of .fam file
    - -o <prefix_filename> Specify prefix for output files. Obligatory.

    - -f <prefix_filename> Specify prefix for .bed + .fam +. bim file, can be used instead of -option -i, e, a
    - -p <platform> Specify platform, options: embark, neogen170, neogen220, lupa170, mdd, wisdom, vcf3, vcf4, merged, other. Obligatory.
    - -s Execute sex check
    - -d Execute duplicate check within input file
    - -m <prefix_filename> Specify prefix for .bed + .fam + .bim file. And execute duplicate check between specified file in -m and file in option -f or -i, e, a. Use in combination with -d.
    - -b <method_tree_construction> Execute breed check Specify method to construct tree, options are: phylip and biopython
    - -h Print the help overview
- Examples:
    - bash quality_control.sh -f prefix_inputfile -p neogen170 -o newfilename
    - bash quality_control.sh -f prefix_inputfile -p embark -s -d -o newfilename
    - bash quality_control.sh -f prefix_inputfile -p embark -s -d -m prefix_second_inputfile -o newfilename
    - bash quality_control.sh -a inputfile.fam -i inputfile.bim -e inputfile.bed -p mdd -o newfilename
    - bash quality_control.sh -f prefix_inputfile -p neogen220 -b phylip -o newfilename
- Dependencies needed:
    - python3
        - when biopython is chosen as tree construction: package biopython
        - when git bash version is used and biopython: packages numpy, scipy, ete3, PyQt5, biopython (only for building trees)
    - plink 1.9 (included in this tool)
    - plink 2 (included in this tool)
    - Phylip's programs neighbor (included in this tool) if chosen tree construction method is phylip

Additional information of options:

- If no optional options (s, d, m, b) are used, only check for sample call rate is performed
- For all platforms:
    - For specifying input files, use either -i, e, a together, or only -f.
- -p merged:"
    - WARNING: when input is a merged dataset, call rate of samples is NOT checked. If merged dataset contains bad quality samples, the check for sex, breed and duplicates/relatedness is not reliable. Always perform quality control steps on each individual dataset before merging.
- When using -b biopython in the git bash .sh script, a png image is made of the tree. This is not done when the linux or windows version is used.

Operating system and used program versions

- quality_control.sh
    - This command line utility operates in Linux and is tested in Ubuntu, so it is adviced to use Ubuntu. If you have a Windows system, a Windows Subsystem for Linux (WSL) should be used. Installation information can be found on the ubuntu website.
- quality_control_GB.sh
    - This utility operates in git bash.
    - With this utility a png image is made if -b biopython is used, this is not done for the scripts that operate in Linux or Windows.
- quality_control_W.sh
    - This utility operates in the Windows Powershell
- Used versions to test the tools:
    - Ubuntu 22.04.2 LTS
        - GLIBC 2.35-0
    - Git bash 2.40.1.windows.1
    - Windows Powershell PSVersion 5.1.22621.2506
    - Python 3.10.12
        - numpy 1.25.2
        - scipy 1.11.4
        - biopython 1.81
        - ete3 3.1.3

        - PyQt5 5.15.10
    - programs neighbor and consense from phylip 3.698

File descriptions:

- YSNPsFemales.list files for Embark and Neogen 220K
    - These files contain the bad quality Y SNPs that are often wrongly called in females. See 'sample call rate check' for more information.
- GetSexY.py
    - Python script to determine sex based on number of Y calls
    - Is used for Embark and Neogen 220K data
- GetSexX.py
    - Python script to determine sex based on X snp homozygosity
    - Is used for platforms without Y data: lupa, neogen 170K, affymetrix, wisdom, vcf, merged files
- ExtractKinshipScores.py
    - Extracts the kinship scores out of the .kin0 file for samples between two input files, and does not extract the kinship scores which are between samples within the first input file
- CheckDuplicateIDs.py
    - Checks if there are duplicate IDs between two input files
- GetDuplicateInfo.py
    - Makes a new txt file with the number of snps per duplicate sample, and their kinship
- GetInnerJoin.py
    - Makes a file with the innerjoin of SNPs (SNPs in common) between the breed database and the input file
- MakeTree.py
    - Makes a phylogenetic tree newick file from a distance matrix, using biopython
    - In case the git bash .sh script is used, this script also makes a png image of the tree
- ReformatDist.py
    - Reformats the distance matrix and makes temporary sample IDs, so the matrix can be used by the PHYLIP package
- UpdateSampleIDs.py
    - Changes the temporary sample IDs in the newick file to the original sample IDs
- Directory temp_files
    - In this directory the temporary files made by the tool are placed
    - There should be no files in this folder after running the tool. However, if an error occurred or if the run was stopped prematurely, there can be files in this folder. These should be removed to prevent build-up of files. This folder gets automatically emptied at the start of a new run of the tool.
- In breed_database directory:
    - bed bim fam file of a SNP dataset containing many dog breeds
        - SNP dataset with multiple breeds (289 breeds, and wolves and coyotes)
        - This dataset contains only autosomal snps and the common SNPs between platforms from which the data comes.
        - Per breed 5 dogs are present, except for coyotes, of which 4 animals are present
        - The samples are selected based on lowest kinship, to minimalize the number of first or second degree relationships between samples.
    - A file with sample IDs in tree and their corresponding original sample IDs
        - This was made, so the breeds in the tree are easier to recognize based on the sample name
    - Breeds_tree.txt is a file with breeds in the SNP dataset

Plink settings

- --chr-set 38 is used in command (not --dog)
    - By doing this, the chromosome coding will remain the same (all in numbers from 1 to 42).
    - Dog has 38 autosomes
- --allow-no-sex
    - prevents errors because of missing sex in .fam or .ped file
- --mind 0.1
    - to exclude samples that have a missing sample call rate higher than 10%
- --missing sample-only 'scols=maybefid,nmiss,nobs,fmiss'

    - to get information about amount of missing SNPs
- --sample-counts 'cols=maybefid,hetsnp'
    - to get information about amount of heterozygous SNPs
- --make-king-table
    - to create a txt file with kinship between samples
- --king-table-filter 0.1875
    - to filter for only kinship scores higher than 0.1875 (=first degree relation)
- --king-table-require
    - to only make a kinship file with certain sample combinations
- --distance triangle 1-ibs
    - for making a distance matrix with lower-triangular format
    - 1-ibs is used to express distances as genomic proportions (1 minus the identity-by-state value)
- --distance square 1-ibs
    - for making a distance matrix with square symmetric format
    - 1-ibs is used to express distances as genomic proportions (1 minus the identity-by-state value)

Output (file) descriptions per check

- Log file: contains output of performed checks and the plink log's.
- Duplicate ID check
    - Reports which sample IDs are present multiple times in the input .fam file
- Sample call rate check
    - bed, bim, fam _bad_samples with the removed bad quality samples, if present
    - Reports: sample call rate for the bad quality samples
    - bed, bim, fam files with the samples that passed this check, and of which the bad quality Y SNPs are removed (if present).
- Sex check
    - file_sex_changed.txt with samples for which the sex differs between input .fam file and snp sex:
        - Family ID
        - Sample ID
        - original_sex: sex in input .fam file
        - SNP_sex: sex based on homozygosity of X snps
        - nr_nonmissing_X_SNPs: number of X SNPs on which sex check is based
        - percentage_homozygous: percentage of homozygous X SNPs
    - new .fam file with the updated sex based on snp sex
    - Reports: for how many samples the sex was different between sex in .fam file and sex based on SNPs
    - Reports: for which samples the sex could not be determined based on SNPs
    - Reports: for which samples the sex check is based on less than 500 X SNPs. (Sex could be less reliable if based on low number of X SNPs.)
- Duplicate check
    - _kinship.kin0 file with kinship scores (higher than 0.1875) between samples. Produced by plink --make-king-table.
        - FID1: family ID sample 1
        - IID1: individual ID sample 1
        - FID2: family ID sample 2
        - IID2: individual ID sample 2
        - NSNP: Number of variants considered (autosomal, neither call missing)
        - HETHET: Proportion/count of considered call pairs which are het-het
        - IBS0: Proportion/count of considered call pairs which are opposite homs
        - KINSHIP: kinship score between both samples
    - _duplicate_summary.txt file with the detected duplicates
        - Family_ID1
        - Sample_ID1
        - SNP_count_ID1: number of successful SNPs for this sample
        - Family_ID2
        - Sample_ID2
        - SNP_count_ID2: number of successful SNPs for this sample
        - Kinship_score: kinship score between both samples
        - Sample_ID_most_SNPs: which sample has the most successful SNPs (can be used to select which sample to use for further research)
    - if option -m is used (same file format as the two files above):

        - Reports which sample IDs are found in both input files (if present)
        - Reports the new temporary unique IDs in the first file, if duplicate sample IDs are present
            - If duplicate IDs are present, the ID of the first input file gets temporarily changed to a unique ID, because the two files need to be merged before kinship scores can be calculated. You don't want two files with the same IDs to be merged, because it merges the two samples, but it is unknown still if these are the same samples.
        - _between_files_temp_kinship.kin0 file with kinship scores between samples of the two input files. Produced by plink --make-king-table.
        - _between_files_duplicate_summary.txt file with the detected duplicates between the two input files.
- Breed check by phylogenetic tree
    - _tree.newick file
        - contains the tree in newick format
        - this tree can be visualized by programs such as ITOL (online tool), dendroscope, figtree etc.
    - _tree_annotation.txt
        - this file can be used to color the new dogs added to the tree
        - works only for ITOL online tool
        - after loading the newick file, in the control panel shown on screen, in the datasets tab, this annotation file can be loaded
    - _tree.png (only produced by tool for Git Bash)
        - Tree image

Checks performed by quality control tool:

Duplicate sample ID check

The input .fam file is checked for duplicate sample IDs, and if these are present, duplicate IDs are reported and an error is given. Duplicate IDs should not be present, because they can cause confusion of the identity of the sample, and quality control steps cannot be performed correctly. Make sure all sample IDs are unique.

Sample call rate check

Consists of two parts:

1. The **removal of unreliable Y SNPs**. This is only performed on embark and neogen 220K datasets, because these platforms have Y SNPs and for those platforms it is known which Y SNPs are unreliable. These Y SNPs are often called in females and are thus unreliable. These SNPs are removed in both males and females. See 'sex check' for more information.
2. The next obligatory step is checking the **sample call rate**: samples with a call rate of less than **90%** of SNPs are removed, and their call rate is reported. This is necessary because sex, breed, and duplicate checks cannot be performed reliably on bad quality samples. This check is not performed on merged SNP datasets, because datasets of different sizes can be merged. Because of this, smaller datasets would be removed by the sample call rate check, even though the quality of these samples is not bad. However, be sure to do quality control checks before you merge different datasets!

Note: Because Embark uses saliva swabs to retrieve DNA, the sample call rate can be lower compared to other platforms

Steps performed by the quality control command line utility:

1. Plink using --mind 0.1 for sample call rate and --exclude for bad Y SNPs (only for embark and neogen220k)
    - To remove samples with call rate under 90% and bad Y snps
    - plink --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --make-bed --mind 0.1 --exclude Neogen220YSNPsFemales.list --allow-no-sex --chr-set 38 --out new_file
2. If there were samples removed because of low sample call rate, plink2 --missing sample-only is used to calculate the sample call rate of these samples.

    - plink2 --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --make-bed --missing sample-only 'scols=maybefid,nmiss,nobs,fmiss' --keep inputfile.irem --allow-no-sex --chr-set 38 --out outputfile
3. Sample call rate of bad quality samples is reported

Sex check

In this check, the sex of a sample is determined based on X or Y SNPs, and compared to the sex in the .fam file. If these sexes are different, this is reported. A new .fam file is made with the sex based on SNPs. The samples for which the sex could not be determined, and the samples for which the sex is determined based on less than 500 X SNPs, are reported. The sex can be determined based on X SNPs or Y SNPs, depending on the platform.

Type of sex determination per platform:

- Sex check based on number of Y SNP calls:
    - embark and neogen220
- Sex check based on proportion homozygous SNPs on X chromosome:
    - neogen170, lupa170, wisdom, affymetrix, vcf3, vcf4, merged, other

General information sex check:

- In the .fam file sex is shown as:
    - 1 = male
    - 2 = female
    - 0 = unknown
- X chromosome SNPs can only be heterozygous in females. Males only have 1 X chromosome, so their alleles should always be homozygous. Based on how homozygous the X chromosome of an individual is, the sex can be determined. However, even female dogs are highly homozygous, and X SNP errors occur, so a small part of the samples will be assigned the wrong sex. Due to this, dogs with homozygosity between 97% and 98.5%, will be assigned a sex of 0 (=unknown). Dogs with homozygosity below 97% are assigned 2 (=female) and above 98.5% are assigned 1 (=male).
- Y chromosome SNPs can only be called in males, because females don't have this chromosome. Based on the number of Y alleles per sample, the sex can be determined. Samples with a high number of Y calls are male, and samples with a low or 0 Y calls are female.
- Y chromosome SNPs that are often called in females are bad quality SNPs, because this should not be possible. In embark and neogen 220K there are a number of SNPs (46 and 49 respectively) that are often called in females, and hence should be removed, because they are not reliable. This is performed simultaneously with the sample call rate check.

Overview differences between arrays

These numbers are after using convert.sh, and only non-pseudoautosomal X SNPs

Platform	Embark	Neogen 170K	Neogen 220K	Lupa 170K	Wisdom	Affymetrix
Nr Y SNPs	258	0	260	0	0	0
Nr Y SNPs removed	46	0	49	0	0	0
Nr X SNPs	6788	5056	6406	5075 - 5117	611 - 616	5967

Steps performed by the quality control command line utility for sex check based on Y SNPs:

1. Plink2 using --missing sample-only --chr 40 to get number of Y alleles called per sample
    - plink2 --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --missing sample-only 'scols=maybefid,nmiss,nobs,fmiss' --chr-set 40 --chr 40 --out new_file
    - Note: --chr-set 40 instead of 38 is used, because plink will otherwise not calculate the number of Y calls, because it is a sex-chromosome.
2. GetSexY.py to check (and change) sex based on Y calls
    - Assigns sex based on the number of Y calls per sample (high in males, low in females)
        - Limit embark: under 100 = female, above 100 = male
        - Limit neogen 220K: under 130 = female, above 130 = male
    - Reports for which samples the sex was changed
    - Creates a new .fam file with sex based on SNP data

Steps performed by the quality control command line utility for sex check based on X SNPs:

1. Plink using --missing sample-only --sample-counts --chr 39 to get number of homozygous X SNPs
    - plink2 --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --missing sample-only 'scols=maybefid,nmiss,nobs,fmiss' --sample-counts 'cols=maybefid,hetsnp' --chr-set 40 --chr 39 --allow-no-sex --out new_file
    - Note: --chr-set 40 instead of 38 is used, because plink will otherwise not calculate the number of X calls, because it is a sex-chromosome.
2. GetSexX.py to check (and change) sex based on X chromosome homozygosity
    - Assigns sex based on proportion of homozygous X SNPs per sample (high in males, lower in females)
        - Limit: under 0.97 = female, above 0.985 = male, inbetween: sex unknown
    - Reports for which samples the sex was changed and on how many X SNPs the sex is based
    - Reports for which samples the sex was based on less than 500 X SNPs
    - Reports for which samples the sex could not be determined based on SNPs
    - Creates a new .fam file with sex based on SNP data

Duplicate/relationship check

In this check, the relationships between samples in the input file is checked, or if the -m option was used, the relationships between samples of 2 files is checked. It is important to identify duplicate samples, because you generally don't want to use the same sample twice in an analysis. Duplicate samples and close family relationships between samples can cause bias. This check reports duplicates and relationships of first degree.

NOTE: no duplicate samples are removed from the input file. Based on the given information, the user should decide further actions for duplicate samples.

Summary

- The duplicate check is based on kinship scores between individuals. The scores are generated by plink2 --make-king-table.
- The highest possible kinship score is 0.5, these are duplicate samples or monozygotic twins.
- Kinship scores of ~0.25 are siblings or parent-child.
- Kinship scores of ~0.125 are second degree relationships: Grandparent-grandchild, aunt/uncle, niece/nephew, half-sibling.
- Kinship scores can be negative, this indicates an unrelated relationship
- The cutoff used for the first degree relationships is 0.1875 (mean of 0.125 and 0.25)
- The cutoff used for duplicates is 0.4. Above this value, samples are duplicates.
- When duplicate samples are found, the SNP count per sample is calculated, so the sample with the most SNPs can be selected for further research purposes.

Steps performed by the quality control command line utility for the duplicate/relationship check:

1. Plink2 using --make-king-table and --king-table-filter 0.1875 to get kinship scores of first degree relationships and duplicates.

    - plink2 --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --make-king-table --king-table-filter 0.1875 --chr-set 38 --out new_file
    - Produces a .kin0 file with the kinship scores
2. Filter out the duplicates (kinship > 0.4) out of the produced .kin0 file in step 1.
3. Plink2 using --missing sample-only to get number of successfully genotyped SNPs per duplicate sample
    - plink2 --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --missing sample-only 'scols=maybefid,nmiss,nobs,fmiss' --keep duplicates.list --chr-set 60 --allow-no-sex --out new_file
    - Note: --chr-set 60 instead of 38 is used, because otherwise plink will not always incorporate the number of Y SNPs, based on the sex in the .fam file. Any number above 42 can be used for this.
4. GetDuplicateInfo.py to make a summary of the duplicate samples, containing:
    - sample IDs, the number of successfully genotyped SNPs per sample, kinship score between samples, and the sample with the most genotyped SNPs If -m option was used:
5. Check for duplicate sample IDs in the file given with the -m option
    - Reports duplicate IDs if present.
6. GetDuplicateIDs.py to check for duplicate IDs between the first input file in option -f or -i,a,e, and the second input file given in option -m.
    - If duplicate IDs are present, the ID of the first input file gets temporarily changed to a unique ID, because the two files need to be merged before kinship scores can be calculated. You don't want two files with the same IDs to be merged, because it merges the two samples, but it is unknown still if these are the same samples.
7. Plink to merge the two files.
    - plink --bim firstfile.bim --fam firstfile.fam --bed firstfile.bed --allow-no-sex --bmerge secondfile.bed secondfile.bim secondfile.fam --chr-set 38 --make-bed --out new_file
8. Plink2 to get kinship scores of samples in merged file using --make-king-table and --king-table-filter 0.1875 and --king-table-require
    - plink2 --bim mergedfile.bim --fam mergedfile.fam --bed mergedfile.bed --make-king-table --king-table-filter 0.1875 --king-table-require firstfile.fam --chr-set 38 --out new_file
    - By using --king-table-require firstfile.fam, only kinship scores are calculated between samples, of which at least 1 sample is in the first input file.
    - Produces a .kin0 file with the kinship scores
9. ExtractKinshipScores.py to extract sample pairs between the two input files
    - To only report kinship scores between samples of the two input files, the kinship scores between samples within the first input file should not be included.
10. Plink2 using --missing sample-only to get number of successfully genotyped SNPs per duplicate sample
    - plink2 --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --missing sample-only 'scols=maybefid,nmiss,nobs,fmiss' --keep duplicates.list --chr-set 60 --allow-no-sex --out new_file
    - Note: --chr-set 60 instead of 38 is used, because otherwise plink will not always incorporate the number of Y SNPs, based on the sex in the .fam file. Any number above 42 can be used for this.
11. GetDuplicateInfo.py to make a summary of the duplicate samples, containing:
12. sample IDs, the number of successfully genotyped SNPs per sample, kinship score between samples, and the sample with the most genotyped SNPs

Breed check by phylogenetic tree

In this check, a phylogenetic tree is made. The new input samples are added to a tree in which many breeds are already present. By doing this, you can check where the new dogs are placed in the tree, and thus check if the known breed for this dog is correct. This check can be used to detect sample swaps.

General information breed check by phylogenetic tree:

- The tree to which the new dogs are added contains many dog breeds (over 200) and is rooted with coyotes.
- In the breed_database folder are the bed bim fam files from this breed database.
    - This dataset contains only autosomal snps and the common SNPs between platforms from which the data comes.
    - Per breed 5 dogs are present, except for coyotes, of which 4 animals are present
    - The samples are selected based on lowest kinship, to minimalize the number of first or second degree relationships between samples.
    - A file with sample IDs in tree and their corresponding original sample IDs is present in the same folder
        - This was made, so the breeds in the tree are easier to recognize based on the sample name

    - Breeds_tree.txt is a file with breeds in the SNP dataset
- The produced tree can be visualized by programs such as ITOL (online tool), dendroscope, figtree etc.
- The produced tree annotations file can be loaded into the ITOL online tool
    - After loading the newick file, in the control panel shown on screen, in the datasets tab
    - By using the annotations file, the new dogs will be colored in the tree
- When a newly added dog is in the outer branch of a breed cluster, it can still be mixed. It can help to add more mixed or purebred animals of the breed to the tree, to see if this dog might be mixed breed.
- Tips for using the ITOL tool:
    - When option for branch lengths is turned off, the branches can be seen better
    - A picture of the tree can be saved
    - You can color animals in the tree

Steps performed by the quality control command line utility for the breed tree check with biopython:

1. GetInnerJoin.py python script to get the common SNPs between the breed database and the input file
    - Produces a list file with common SNPs
2. Plink to merge the breed database with the input files and only keep the common SNPs
    - plink --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --extract snp_list --bmerge database_files --allow-no-sex --chr-set 38 --out new_file
3. Plink using --distance triangle 1-ibs
    - To make a distance matrix in lower triangle format
    - plink --bim mergedfile.bim --fam mergedfile.fam --bed mergedfile.bed --distance triangle 1-ibs --allow-no-sex --chr-set 38 --out new_file
4. MakeTree.py to make a phylogenetic tree from the distance matrix, using biopython
    - To make a phylogenetic rooted tree by using the neighborjoin method
    - If the git bash .sh version is used, this script also outputs a tree .png image

Steps performed by the quality control command line utility for the breed tree check with phylip:

1. GetInnerJoin.py python script to get the common SNPs between the breed database and the input file
    - Produces a list file with common SNPs
2. Plink to merge the breed database with the input files and only keep the common SNPs
    - plink --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --extract snp_list --bmerge database_files --allow-no-sex --chr-set 38 --out new_file
3. Plink using --distance square 1-ibs
    - To make a distance matrix in square symmetrical format
    - plink --bim mergedfile.bim --fam mergedfile.fam --bed mergedfile.bed --distance square 1-ibs --allow-no-sex --chr-set 38 --out new_file
4. ReformatDist.py to reformat the distance matrix to phylip format
    - The distance matrix made in step 2, needs to be adjusted, so it can be used as input in the phylip program
    - The phylip format does not allow for sample IDs longer than 10 characters and wants a specific format for the IDs, so the sample ids are recoded to a temporary id. This is written to a file, so they can be reversed later.
5. Neighbor program from phylip to make a phylogenetic tree in newick format from the reformatted distance matrix
    - Used settings are:
        - O - Outgroup is used (is a coyote)
        - J - Input order of species is randomized
6. UpdateSampleIDs to revert the temporary sample IDs in the consensus newick file back to the original IDs

Credits

This project is part of the Expertise Centre Genetics of Companion Animals (Faculty veterinary medicine, Utrecht University).

License