Available online at www.sciencedirect.com

ScienceDirect

Computers and Electronics in Agriculture 57 (2007) 107–113

Computers and electronics in agriculture

www.elsevier.com/locate/compag

Application note

PyPedal: A computer program for pedigree analysis

John B. Cole [*]

Animal Improvement Programs Laboratory, Agricultural Research Service,
United States Department of Agriculture, Room 306, Bldg. 005, BARC-West,
10300 Baltimore Avenue, Beltsville, MD 20705-2350, USA

Received 26 July 2006; received in revised form 8 February 2007; accepted 8 February 2007

Abstract

PyPedal is a pedigree analysis package that provides tools for error checking, mathematical analysis, report generation, pedigree simulation, and data visualization. A number of measures of genetic variability are provided, including coefficients of inbreeding and relationship, effective founder and ancestor numbers, and founder genome equivalents. Routines are also included for identifying ancestors and descendants, computing coefficients of inbreeding from potential matings, quantifying pedigree completeness, visualizing pedigrees, and producing high-quality printed reports. In addition, a module is provided for applying graph theoretic tools to pedigrees. Input and output files utilize plain-text formats, and printed reports are rendered as Adobe PDF files. Users can easily write programs for automating analyses as well as create new reports. PyPedal has been validated using dairy cattle and working dog pedigrees. It is written in the Python programming language and operates on a number of operating systems, including GNU/Linux and Microsoft Windows. The program is free of charge; code, documentation, and examples of usage are available at http://pypedal.sourceforge.net/.

Keywords: Inbreeding; Pedigree analysis; Pedigree simulation; Visualization

1. Introduction

Pedigree analysis is a valuable tool for describing genetic diversity in animal populations (Cole et al., 2004). Although conceptually simple, measures of genetic diversity must be computed using specially written software for all but trivial pedigrees. A number of software packages provide this functionality, such as CFC (Sargolzaei et al., 2006), ENDOG (Gutiérrez and Goyache, 2005), and Pedig (Boichard, 2002). However, CFC and ENDOG are limited to a single operating system (MS Windows), and Pedig lacks report generation and visualization tools. PyPedal is portable across operating systems and provides tools for error checking, mathematical analysis, report generation, pedigree simulation, and data visualization.

[*] Tel.: +1 301 504 8665; fax: +1 301 504 8092.
E-mail address: jcole@aipl.arsusda.gov.

0168-1699/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2007.02.002

Table 1
PyPedal modules

Module name	Module description	Routines
pyp_db	Working with SQLite relational databases: create databases, add/drop tables, load PyPedal pedigrees into tables	8 procedures 4 classes
pyp_demog	Generate demographic reports, age distributions, for the pedigreed population	4 procedures
pyp_graphics	Visualize pedigrees and numerator relationship matrices (NRM)	9 procedures
pyp_io	Save and load NRM and inverses of NRM; write pedigrees to formats used by other packages	13 procedures
pyp_jbc	User-written custom procedures for coloring pedigrees	3 procedures
pyp_metrics	Compute metrics on pedigrees: effective founder and ancestor numbers, effective number of founder genomes, pedigree completeness. Tools for identifying related animals, calculating coefficients of inbreeding and relationship, and computing expected offspring inbreeding from matings	22 procedures
pyp_network	Convert pedigrees to directed graphs; apply network analysis and graph theory to pedigrees	19 procedures
pyp_newclasses	Pedigree, animal, and metadata classes used by PyPedal	4 classes
pyp_nrm	Creating, decompose, and inverting NRM, and recurse through pedigrees	15 procedures
pyp_reports	Create reports from pedigree database (loaded in pyp_db)	7 procedures
pyp_reports_template	Skeleton for use in writing custom reports	3 procedures
pyp_template	Skeleton for use in writing custom modules	1 procedure
pyp_utils	Load, reorder and renumber pedigrees; set flags in individual animal records; string and date-time tools	19 procedures

2. Materials and methods

2.1. Program design

PyPedal (Cole and Franke, 2002) is written in the Python programming language (v2.4; http://www.python.org/) and has been tested on the Microsoft Windows XP[1] (32-bit) and GNU/Linux (Fedora Core 5, 64-bit) operating systems. It may be used interactively or programs can be run in batch mode. PyPedal is built as a series of modules (Table 1), each of which collects related functions, and incorporates both object-oriented and procedural paradigms. Extensive use is made of third-party modules for matrix manipulation, pedigree visualization and graph drawing, report generation, and network analysis.

Python was chosen over other programming languages such as FORTRAN (Pedig), Visual Basic (ENDOG), and Visual C++ (CFC) because of its support for procedural and object-oriented programming paradigms, its rich data structures, the availability of third-party libraries, and speed of development. Compiled languages (e.g. FORTRAN) sometimes out-perform interpreted languages (e.g. Python), but that is typically due to poor algorithm design rather than innate limitations of interpreted languages. PyPedal performs well on pedigrees of hundreds to thousands of animals and is capable of processing pedigrees of hundreds-of-thousands of records.

Input pedigrees are described by simple format strings and read from ASCII flatfiles into pedigree objects. Pedigrees may also be simulated or read from directed graphs. Numerator relationship matrices (NRM) may be stored in pedigree objects, reducing the need to repeat time-consuming calculations. Heuristics are used to improve data completeness when minimal information is provided; for example, PyPedal can infer sexes if they are not provided.

Pedigree objects contain a list of instances of animal objects and a pedigree metadata object. Metadata are collected when a pedigree is loaded and are used by other routines to avoid unnecessary pedigree traversal. Pedigree objects are passed by reference to procedures in PyPedal modules; NRM are instances of NumPy (http://www.numpy.org/) matrix objects, which are dense-stored.

Pedigrees may also be produced by simulation, and a number of options are provided to produce pedigrees with structures of interest. Populations may be closed or open, the sex ratio defined, the number of parents of each sex and number of generations specified, and parent–offspring and full sib matings can be allowed. Simulated pedigrees are useful for studying the network structure induced by genetic relationships.

[1] Reference to any commercial product is made with the understanding that no discrimination is intended and no endorsement by USDA is implied.

2.2. Input and output files

Most input and output files utilize plain ASCII text formats, although some graphics and matrix routines write binary files. Animal IDs may be provided as either integers or strings; strings are hashed to integers internally. Comments and user-specified column delimiters may be included in pedigree files. Pedigree errors including duplicate animal IDs, animals appearing as both sires and dams, animals older than their parents, and animals with the same ID as a parent are detected and the user notified. Pedigree records are automatically generated for animals that appear only as parents.

PyPedal is also able to write pedigrees and NRM to disc as persistent Python objects; these objects are stored plain-text files, but they are not human-readable in the usual sense. Log files are generated automatically when a PyPedal program is run. Program options, such as the pedigree format code, may be set in the program or read from a file. A pedigree format code and pedigree file name must be provided; additional parameters may be provided to over ride defaults. The "set_sexes" option enables the sex-inference heuristic, "renumber" requests that the pedigree be reordered and renumbered, and "pedcomp" indicates that pedigree completeness (Cassell et al., 2003) should be calculated for each animal in the pedigree. A complete list of options and their functions is provided in the manual.

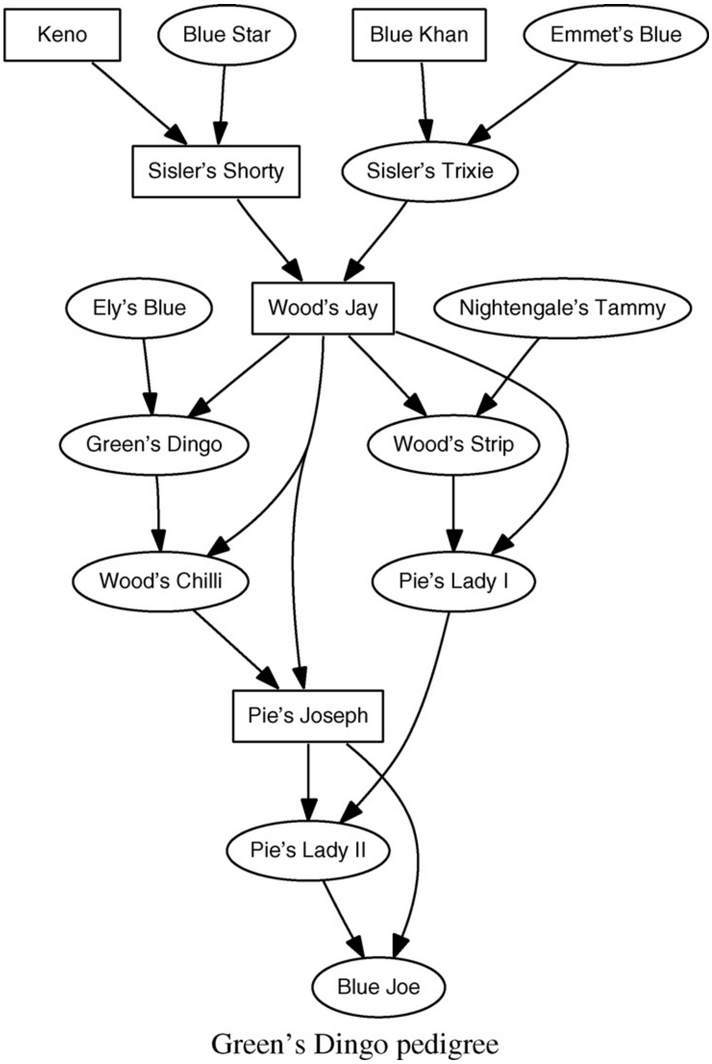


Fig. 1. A drawing of a horse pedigree.

2.3. Pedigree metrics

Routines for calculating a number of measures of genetic variation are included in PyPedal, including effective founder numbers and founder genome equivalents (Lacy, 1989), effective ancestor numbers (Boichard et al., 1997), average coefficients of inbreeding and relationship (Wright, 1922), theoretical and realized effective population sizes (Falconer and MacKay, 1996), and pedigree completeness (Cassell et al., 2003).

Founder alleles are simulated and segregated through the pedigree to calculate the effective number of founder genomes (MacCluer et al., 1986), but molecular data are not otherwise utilized. Routines that return values for each animal in the pedigree also return summary statistics such as means, minima, and maxima. Tools are also provided for calculating the additive relationship between two individuals, calculating the inbreeding of a given mating, identifying common ancestors, and calculating generation lengths and generation intervals. Results are returned in dictionaries that are easily passed to other routines for additional computation, plotting, or reporting. Most routines also write results to a file automatically.

Coefficients of relationship and inbreeding are calculated using the method of VanRaden (1992) in which pedigrees for individual animals are extracted from the full pedigree and relationship matrices calculated using the tabular method (Emik and Terrill, 1949). Diagonals may be adjusted for the inbreeding of the parents. Inverse NRM ignoring or accounting for inbreeding are formed directly using the methods of Henderson (1976) and Quaas (1976).

2.4. Pedigree and data visualization

Pedigree drawing is a challenging problem for all but trivial populations. PyPedal uses Graphviz (Gansner and North, 1999; http://www.graphviz.org/), an application for visualizing directed graphs, to draw pedigrees (Fig. 1). Both display- (e.g. JPG and PNG) and print-oriented (e.g. PS) formats are supported. The "draw_colored_pedigree" routine in the pyp_jbc module produces a pedigree in which nodes are colored based on the number of sons an animal has. Additional enhancements are possible, such as weighting edges between animals based on their additive relationship. Basic routines are also provided for plotting data over time (Fig. 2), and for visualizing the values and sparsity of NRM as image maps. Visualization is an area that has not been well-developed in animal breeding (Huang and Shanks, 1995).

2.5. Report generation

The pyp_db module uses SQLite (http://sqlite.org/) for creating and working with relational databases. PyPedal pedigrees are stored in a database and can be accessed using command line tools or bindings to a number of programming

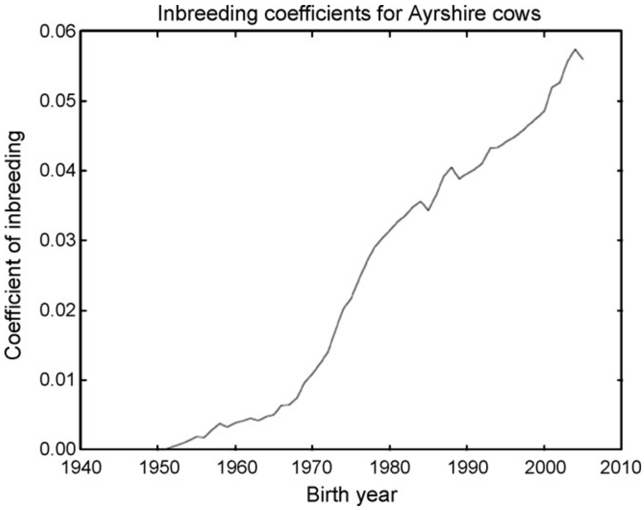
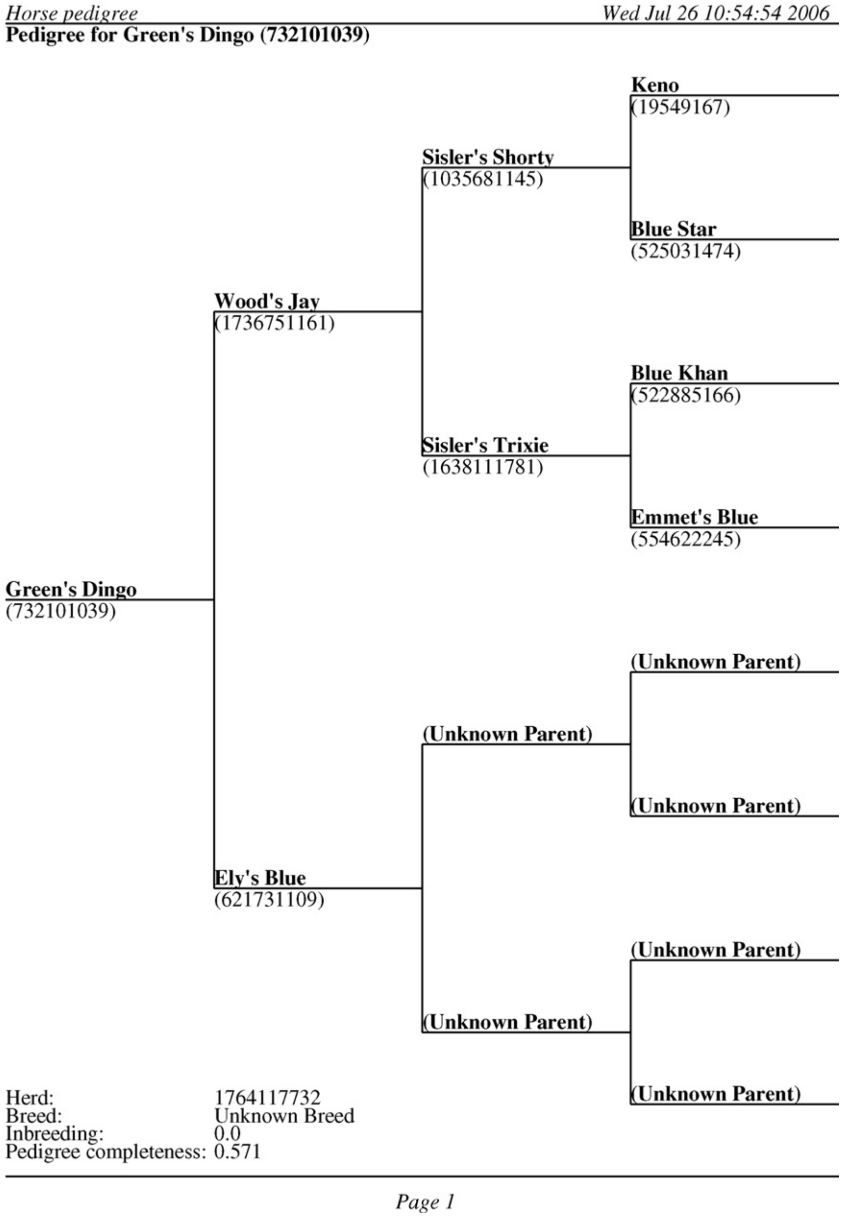


Fig. 2. Average inbreeding of the U.S. Ayrshire population by birth year.



Horse pedigree *Wed Jul 26 10:54:54 2006*

Pedigree for Green's Dingo (732101039)

Herd: 1764117732
Breed: Unknown Breed
Inbreeding: 0.0
Pedigree completeness: 0.571

Page 1

Fig. 3. Print-ready three-generation horse pedigree.

languages. This is of great value to the user in that data are not bound to a particular application or proprietary data storage format. In conjunction with the pyp_reports module, which allows users to create reports in Adobe's Portable Document Format (Fig. 3), users have the tools to easily define custom reports. Basic reports are provided in the pyp_reports module, and the pyp_reports template module provides a template for use in writing custom reports.

2.6. Network analysis

The features discussed in the preceding sections are applicable to both routine pedigree analysis and population management as well as research. Graph theoretic approaches to the study of networks (Newman, 2003) have proven insightful in a number of fields, including sociology and biology. The pyp_network module provides a number of network analysis tools for research into their application to pedigrees. Although the interpretation of many parameters is unclear in the context of pedigree analysis, some do show promise for error-checking and assessment of pedigree connectedness.

For example, it can be shown that animal pedigrees are directed acyclic graphs, with nodes representing animals and edges representing gene flow from parents to offspring. Edges in directed graphs flow from a source to a sink, in this case from parents to offspring, and edges should never flow from offspring to either parent. A dyad census is constructed by examining all pairwise combinations of animals in a graph and enumerating the number of null dyads (pairs with no connection between them), asymmetric arcs (pairs with one connection), and mutual arcs (pairs with two connections). A pair of mutual arcs indicates an error in the pedigree, such as coding an animal as its own parent. A conceptually similar triad census can be used to identify cases in which an animal is coded as its own grandparent. Efficient algorithms exist for a number of other graph-related problems (Cormen et al., 2003), such as identifying groups of nodes that are loosely or unconnected to the other groups of nodes, which may have value in quantifying the degree of connectedness in pedigrees.

3. Discussion

Cole et al. (2004) used PyPedal to describe the population genetic structure of a colony of dog guides and study changes in genetic diversity over time. PyPedal has also been validated by comparison of results against examples in the literature (Boichard et al., 1997; Cassell et al., 2003; Lacy, 1989). Inbreeding calculations have also been checked for a pedigree of ∼500,000 animals by comparing results from PyPedal to those calculated using the method of VanRaden (1992). PyPedal is capable of performing calculations on extremely large pedigrees, but memory requirements grow linearly with the size of the pedigree and an instance of an animal object requires ∼1200 bytes of storage.

4. Conclusion

PyPedal provides a rich set of tools for working with animal pedigrees and is easily extensible using the Python programming language. Users can assess the health of a population using various measures of genetic diversity, explore alternative management scenarios, prepare electronic and printed reports, and easily visualize pedigrees. The PyPedal website is http://pypedal.sourceforge.net/; downloads are available at http://sourceforge.net/projecfe/pypedal/.

Acknowledgments

Work on PyPedal was partially supported by a grant from The Seeing Eye Inc., Morristown, NJ. The author is particularly grateful to E. Hagen, B. Heins, T. von Hassel, and M. Kelly for their feedback, bug reports, and sample datasets.

References

Boichard, D., 2002. PEDIG: a FORTRAN package for pedigree analysis suited for large populations. In: Comm. 28-13 in Proceedings of the 7th World Congr. Genet. Appl. Livest. Prod, Montpellier, France.

Boichard, D., Maignel, L., Verrier, E., 1997. The value of using probabilities of gene origin to measure genetic variability in a population. Genet. Select. Evol. 29, 5–23.

Cassell, B.G., Adamec, V., Pearson, R.E., 2003. Effect of incomplete pedigrees on estimates of inbreeding and inbreeding depression for days to first service and summit milk yield in Holsteins and Jerseys. J. Dairy Sci. 86, 2967–2976.

Cole, J.B., Franke, D.E., 2002. Pedigree analysis using the Python programming language. J. Anim. Sci. 80 (Suppl. 1), 323 (abstract).

Cole, J.B., Franke, D.E., Leighton, E.A., 2004. Population structure of a colony of dog guides. J. Anim. Sci. 82, 2906–2912.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2003. Introduction to Algorithms, 2nd ed. Prentice-Hall, New York.

Emik, L.O., Terrill, C.E., 1949. Systematic procedures for calculating inbreeding coefficients. J. Heredity 40, 51–55.

Falconer, D.S., MacKay, T.F., 1996. Introduction to Quantitative Genetics, 4th ed. John Wiley & Sons Inc., New York.

Gansner, E.R., North, S.C., 1999. An open graph visualization system and its applications to software engineering. Softw. Pract. Exp. 00 (S1), 1–5.

Gutiérrez, J.P., Goyache, F., 2005. A note on ENDOG: a computer program for analyzing pedigree information. J. Anim. Breed. Genet. 122, 172–176.

Henderson, C.R., 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32, 69–83.

Huang, Y.C., Shanks, R.D., 1995. Visualization of inheritance patterns from graphic representation of additive and dominance relationships between animals. J. Dairy Sci. 78, 2877–2883.

Lacy, R.C., 1989. Analysis of founder representation in pedigrees: founder equivalents and founder genome equivalents. Zoo Biol. 8, 111–123.

MacCluer, J.W., VandeBerg, J.L., Read, B., Ryder, O.A., 1986. Pedigree analysis by computer simulation. Zoo Biol. 5, 147–160.

Newman, M.E.J., 2003. The structure and function of complex networks. SIAM Rev. 45, 167–256.

Quaas, R.L., 1976. Computing the diagonal elements of a large numerator relationship matrix. Biometrics 32, 949–953.

Sargolzaei, M., Iwaisaki, H., Colleau, J.J., 2006. CFC: a tool for monitoring genetic diversity. In: Comm 27-28 in Proceedings of the 8th World Congr. Genet. Appl. Livest. Prod, Belo Horizonte, Brazil.

VanRaden, P.M., 1992. Accounting for inbreeding and crossbreeding in genetic evaluation of large populations. J. Dairy Sci. 75, 3136–3144.

Wright, S., 1922. Coefficients of inbreeding and relationship. Am. Nat. 56, 330–338.