

1 Application note

2 PyPedal: a computer program for pedigree analysis

3 John B. Cole

4 Animal Improvement Programs Laboratory

5 Agricultural Research Service

6 United States Department of Agriculture

7 Room 306, Bldg. 005, BARC-West

8 10300 Baltimore Avenue

9 Beltsville, MD 20705-2350 USA

10 Tel.: +1 301 504 8665; fax: +1 301 504 8092

11 E-mail address: jcole@aipl.arsusda.gov

12

13 **Abstract**

14 PyPedal is a pedigree analysis package that provides tools for error checking,
15 mathematical analysis, report generation, pedigree simulation, and data visualization. A
16 number of measures of genetic variability are provided, including coefficients of
17 inbreeding and relationship, effective founder and ancestor numbers, and founder genome
18 equivalents. Routines are also included for identifying ancestors and descendants,
19 computing coefficients of inbreeding from potential matings, quantifying pedigree
20 completeness, visualizing pedigrees, and producing high-quality printed reports. In
21 addition, a module is provided for applying graph theoretic tools to pedigrees. Input and
22 output files utilize plain-text formats, and printed reports are rendered as Adobe PDF files.
23 Users can easily write programs for automating analyses as well as create new reports.
24 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

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.

Materials and Methods

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 XP¹ (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.

¹ Reference to any commercial product is made with the understanding that no

discrimination is intended and no endorsement by USDA is implied.

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

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

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

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

72 Input and Output Files

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

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

90 Pedigree Metrics

91 Routines for calculating a number of measures of genetic variation are included in
92 PyPedal, including effective founder numbers and founder genome equivalents (Lacy,
93 1989), effective ancestor numbers (Boichard et al., 1997), average coefficients of
94 inbreeding and relationship (Wright, 1922), theoretical and realized effective population
95 sizes (Falconer and MacKay, 1996), and pedigree completeness (Cassell et al., 2003).

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

104 Coefficients of relationship and inbreeding are calculated using the method of
105 VanRaden (1992) in which pedigrees for individual animals are extracted from the full
106 pedigree and relationship matrices calculated using the tabular method (Emik and Terrill,
107 1949). Diagonals may be adjusted for the inbreeding of the parents. Inverse NRM ignoring
108 or accounting for inbreeding are formed directly using the methods of Henderson (1976)
109 and Quaas (1976).

110 Pedigree and Data Visualization

111 Pedigree drawing is a challenging problem for all but trivial populations. PyPedal
112 uses Graphviz (Gansner and North, 1999; <http://www.graphviz.org/>), an application for
113 visualizing directed graphs, to draw pedigrees (Figure 1). Both display- (e.g. JPG, PNG)
114 and print-oriented (e.g. PS) formats are supported. The “draw_colored_pedigree” routine in
115 the pyp_jbc module produces a pedigree in which nodes are colored based on the number
116 of sons an animal has. Additional enhancements are possible, such as weighting edges
117 between animals based on their additive relationship. Basic routines are also provided for
118 plotting data over time (Figure 2), and for visualizing the values and sparsity of NRM as

119 image maps. Visualization is an area that has not been well-developed in animal breeding
120 (Huang and Shanks, 1995).

121 Report Generation

122 The pyp_db module uses SQLite (<http://sqlite.org/>) for creating and working with
123 relational databases. PyPedal pedigrees are stored in a database and can be accessed using
124 command line tools or bindings to a number of programming languages. This is of great
125 value to the user in that data are not bound to a particular application or proprietary data
126 storage format. In conjunction with the pyp_reports module, which allows users to create
127 reports in Adobe's Portable DocumentFormat, users have the tools to easily define custom
128 reports. Basic reports are provided in the pyp_reports module, and the
129 pyp_reports_template module provides a template for use in writing custom reports.

130 Network Analysis

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

138 For example, it can be shown that animal pedigrees are directed acyclic graphs,
139 with nodes representing animals and edges representing gene flow from parents to
140 offspring. Edges in directed graphs flow from a source to a sink, in this case from parents
141 to offspring, and edges should never flow from offspring to either parent. A dyad census is
142 constructed by examining all pairwise combinations of animals in a graph and enumerating

143 the number of null dyads (pairs with no connection between them), asymmetric arcs (pairs
144 with one connection), and mutual arcs (pairs with two connections). A pair of mutual arcs
145 indicates an error in the pedigree, such as coding an animal as its own parent. A
146 conceptually-similar triad census can be used to identify cases in which an animal is coded
147 as its own grandparent. Efficient algorithms exist for a number of other graph-related
148 problems (Cormen et al., 2003), such as identifying groups of nodes that are loosely- or
149 unconnected to the other groups of nodes, which may have value in quantifying the degree
150 of connectedness in pedigrees.

151 **Discussion**

152 Cole et al. (2004) used PyPedal to describe the population genetic structure of a
153 colony of dog guides and study changes in genetic diversity over time. PyPedal has also
154 been validated by comparison of results against examples in the literature (Boichard et al.,
155 1997; Cassell et al., 2003; Lacy, 1989). Inbreeding calculations have also been checked for
156 a pedigree of ~500,000 animals by comparing results from PyPedal to those calculated
157 using the method of VanRaden (1992). PyPedal is capable of performing calculations on
158 extremely large pedigrees, but memory requirements grow linearly with the size of the
159 pedigree and an instance of an animal object requires ~1,200 bytes of storage.

160 **Conclusion**

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

167 Acknowledgments

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

171 References

- 172 Boichard, D., 2002. PEDIG: a FORTRAN package for pedigree analysis suited for large
173 populations. Comm. 28-13 in Proc. 7th World Congr. Genet. Appl. Livest. Prod.,
174 Montpellier, France.
- 175 Boichard, D., Maignel, L., Verrier, E., 1997. The value of using probabilities of gene origin
176 to measure genetic variability in a population. Genet. Select. Evol. 29, 5-23.
- 177 Cassell, B.G., Adamec, V., Pearson, R.E., 2003. Effect of incomplete pedigrees on
178 estimates of inbreeding and inbreeding depression for days to first service and summit milk
179 yield in Holsteins and Jerseys. J. Dairy Sci. 86, 2967-2976.
- 180 Cole, J.B., Franke, D.E., 2002. Pedigree analysis using the Python programming language.
181 J. Anim. Sci. 80 (Supplement 1), 323 (abstract).
- 182 Cole, J.B., Franke, D.E., Leighton, E.A., 2004. Population structure of a colony of dog
183 guides. J. Anim. Sci. 82, 2906-2912.
- 184 Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C., 2003. Introduction to
185 Algorithms, 2nd ed. Prentice-Hall, New York.
- 186 Emik, L.O., Terrill, C.E., 1949. Systematic procedures for calculating inbreeding
187 coefficients. J. Heredity 40, 51-55.
- 188 Falconer, D.S., MacKay, T.F., 1996. Introduction to Quantitative Genetics., 4th ed. John
189 Wiley & Sons, Inc., New York.

- 190 Gansner, E.R., North, S.C., 1999. An open graph visualization system and its applications
191 to software engineering. *Softw.Pract. Exper.* 00(S1), 1–5.
- 192 Gutiérrez, J.P., Goyache, F. 2005. A note on ENDOG: a computer program for analysing
193 pedigree information. *J. Anim. Breed. Genet.* 122, 172-176.
- 194 Henderson, C.R., 1976. A simple method for computing the inverse of a numerator
195 relationship matrix used in prediction of breeding values. *Biometrics* 32, 69-83.
- 196 Huang, Y.C., Shanks, R.D., 1995. Visualization of inheritance patterns from graphic
197 representation of additive and dominance relationships between animals. *J. Dairy Sci.* 78,
198 2877-2883.
- 199 Lacy, R.C., 1989. Analysis of founder representation in pedigrees: founder equivalents and
200 founder genome equivalents. *Zoo Biol.* 8, 111-123.
- 201 MacCluer, J.W., VandeBerg, J. L. Read, B., Ryder, O. A., 1986. Pedigree analysis by
202 computer simulation. *Zoo Biol.* 5, 147-160.
- 203 Newman, M.E.J., 2003. The structure and function of complex networks. *SIAM Rev.* 45,
204 167-256.
- 205 Quaas, R.L., 1976. Computing the diagonal elements of a large numerator relationship
206 matrix. *Biometrics* 32, 949-953.
- 207 Sargolzaei, M., Iwaisaki, H., Colleau, J.J., 2006. CFC: a tool for monitoring genetic
208 diversity. Comm 27-28 in Proc. 8th World Congr. Genet. Appl. Livest. Prod., Belo
209 Horizonte, Brazil.
- 210 VanRaden, P.M., 1992. Accounting for inbreeding and crossbreeding in genetic evaluation
211 of large populations. *J. Dairy Sci.* 75, 3136-3144.
- 212 Wright, S., 1922. Coefficients of inbreeding and relationship. *Amer. Natural.* 56:330-338.

213 **Figure Captions**

214 **Figure 1.** A drawing of a horse pedigree.

215 **Figure 2.** Average inbreeding of the U.S. Ayrshire population by birth year.

216 **Figure 3.** Print-ready three-generation horse pedigree.

217 **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).	1 procedure
pyp_reports_template	Skeleton for use in writing custom reports.	
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