

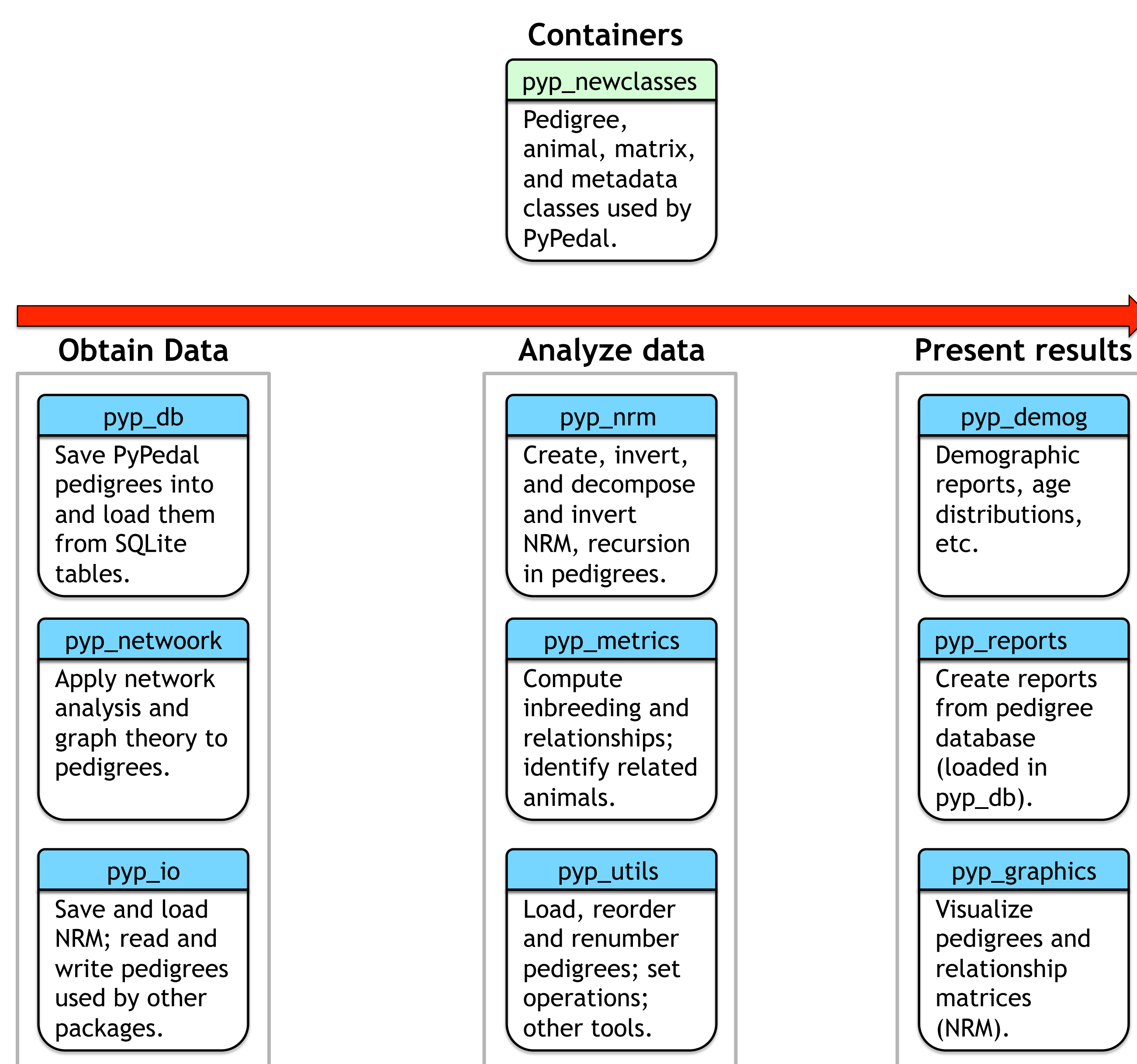
Introduction

PyPedal is an open source package written in the Python programming language that provides high-level tools for manipulating pedigrees. The goal is to provide expressive tools for exploratory data analysis.

Program Organization

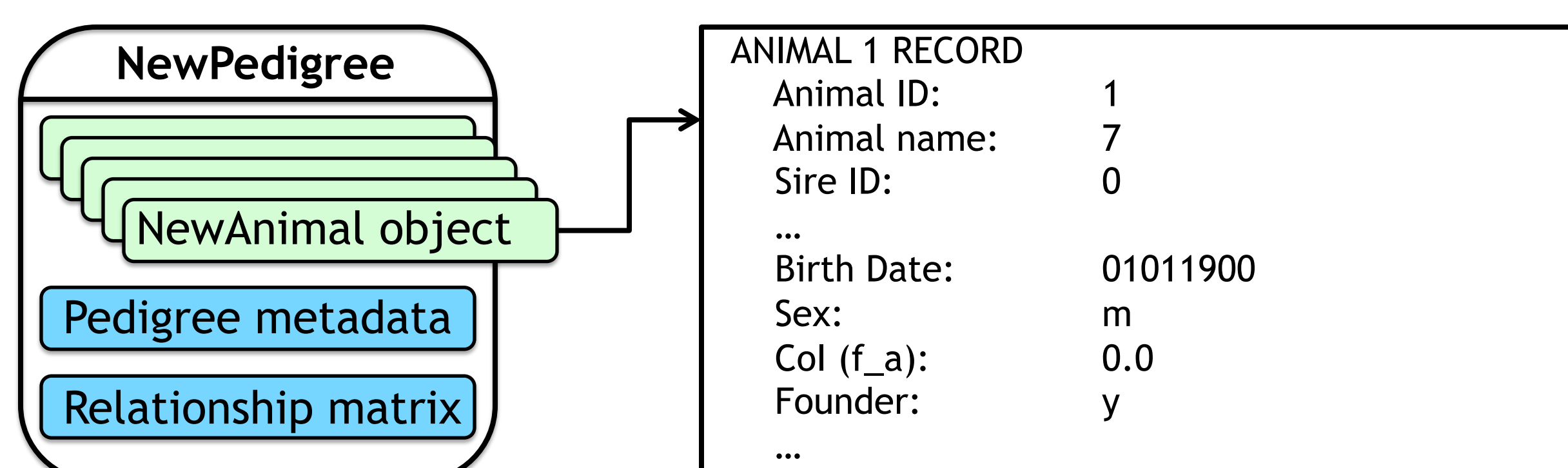
PyPedal is built as a series of modules (Figure 1), each of which groups related functions. Third-party modules are used for matrix manipulation, pedigree visualization and graph drawing, and report generation.

Figure 1. Important PyPedal modules.



What is a pedigree?

A PyPedal **pedigree** is a complex object that includes information about individual animals, data about the group of animals in a pedigree, and code for manipulating those data.



Input and Output

- The simplest way to get a pedigree is to read it from a text file:

```
>>> p = pyp_newclasses.loadPedigree(options)
>>> print p
<PyPedal.pyp_newclasses.NewPedigree instance
at 0x10604a560>
```

Table 1. Input and output options

Input	Output
Plain text files	Binary objects
Simulated pedigrees	Adjacency matrices
	SQLite databases
	GEDCOM 5.5
	GENES 1.20 (DBASE III)

Pedigree Load Process

- Data integrity checks:
 - Duplicate records eliminated
 - Parents without records added to the pedigree
 - Animals cannot appear as sires and dams
 - Numeric or character IDs
- Animals added to `NewPedigree` object
- ID cross-references established
- Metadata computed and attached to pedigree
- Pedigree is renumbered (optional)
- Numerator relationship matrix formed and attached to pedigree (optional)
- Missing information inferred (optional)

Inbreeding and Relationships

- One of the most common pedigree operations is calculation of inbreeding and relationships.
- PyPedal originally used the recursive tabular method of VanRaden (1992)
 - Tested on a pedigree of 600,000 Ayrshires
 - Relatively slow (function call overhead)
- PyPedal 2.0.4 has much faster inbreeding routines than previous versions
 - Meuwissen and Luo (1992)
 - Quaas's modified Meuwissen and Luo (1996)
 - Tested on simulated pedigrees (Table 2)

Table 2. Performance of inbreeding routines.

Size (n)	Method	Time (s)	Speedup
100	VanRaden	< 1	-
	Meuwissen & Luo	< 1	1x
	Modified M&L	< 1	1x
1,000	VanRaden	6	-
	Meuwissen & Luo	1	6x
	Modified M&L	< 1	6x
10,000	VanRaden	304	-
	Meuwissen & Luo	20	15x
	Modified M&L	22	14x
100,000	VanRaden	26,726	-
	Meuwissen & Luo	893	30x
	Modified M&L	978	27x

- It's a one-liner to compute coefficients of inbreeding:

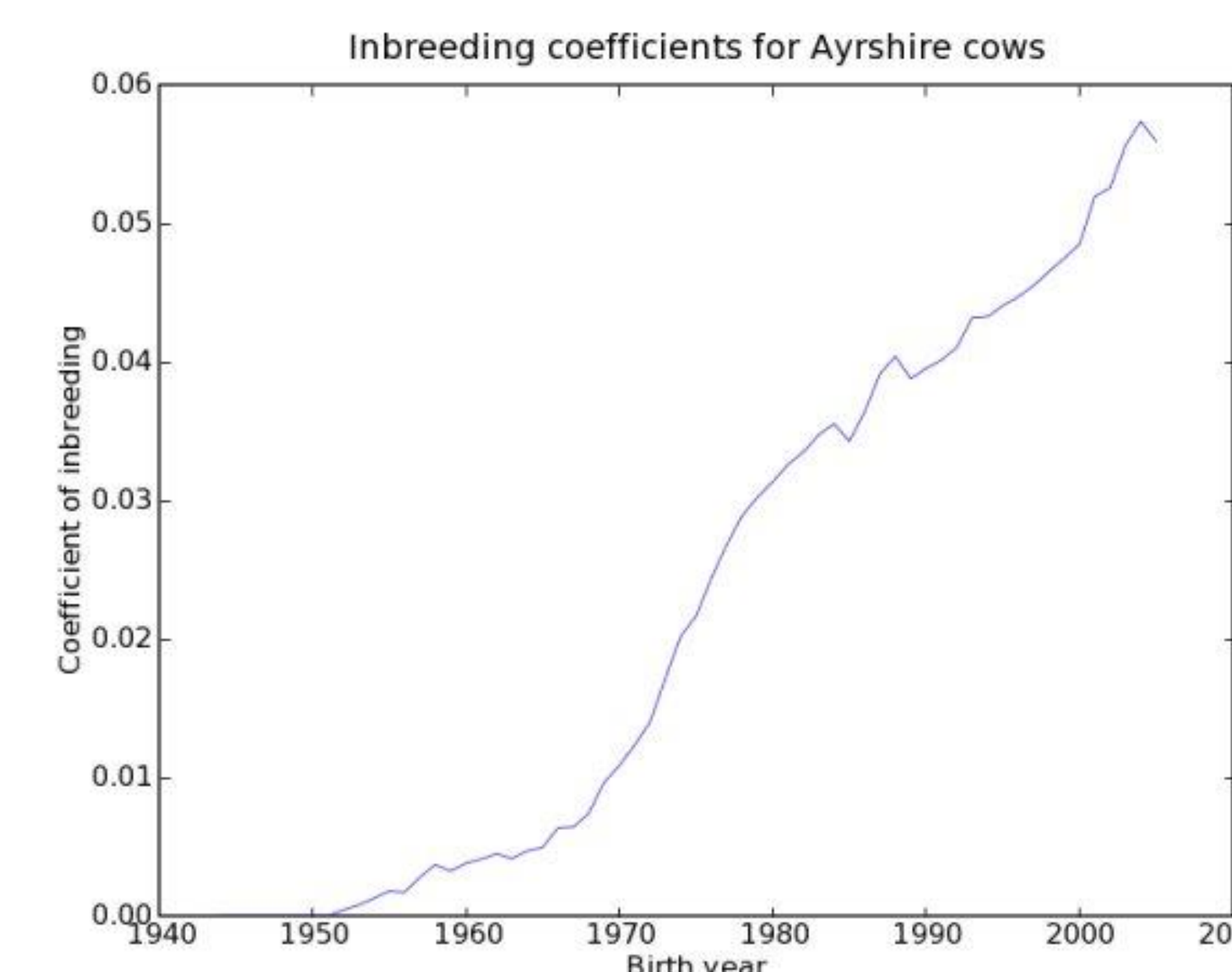

```
>>> fa = pyp_nrm.inbreeding(p, \ method='meu_luo')
```
- VanRaden's method also provides relationships:


```
>>> fa, reln = pyp_nrm.inbreeding(p, \
method='vanraden')
```

Other Measures of Diversity

- Many measures of pedigree diversity are implemented in PyPedal:
 - Ancestral and partial inbreeding
 - Effective founder and ancestor numbers
 - Founder genome equivalents
 - Pedigree completeness

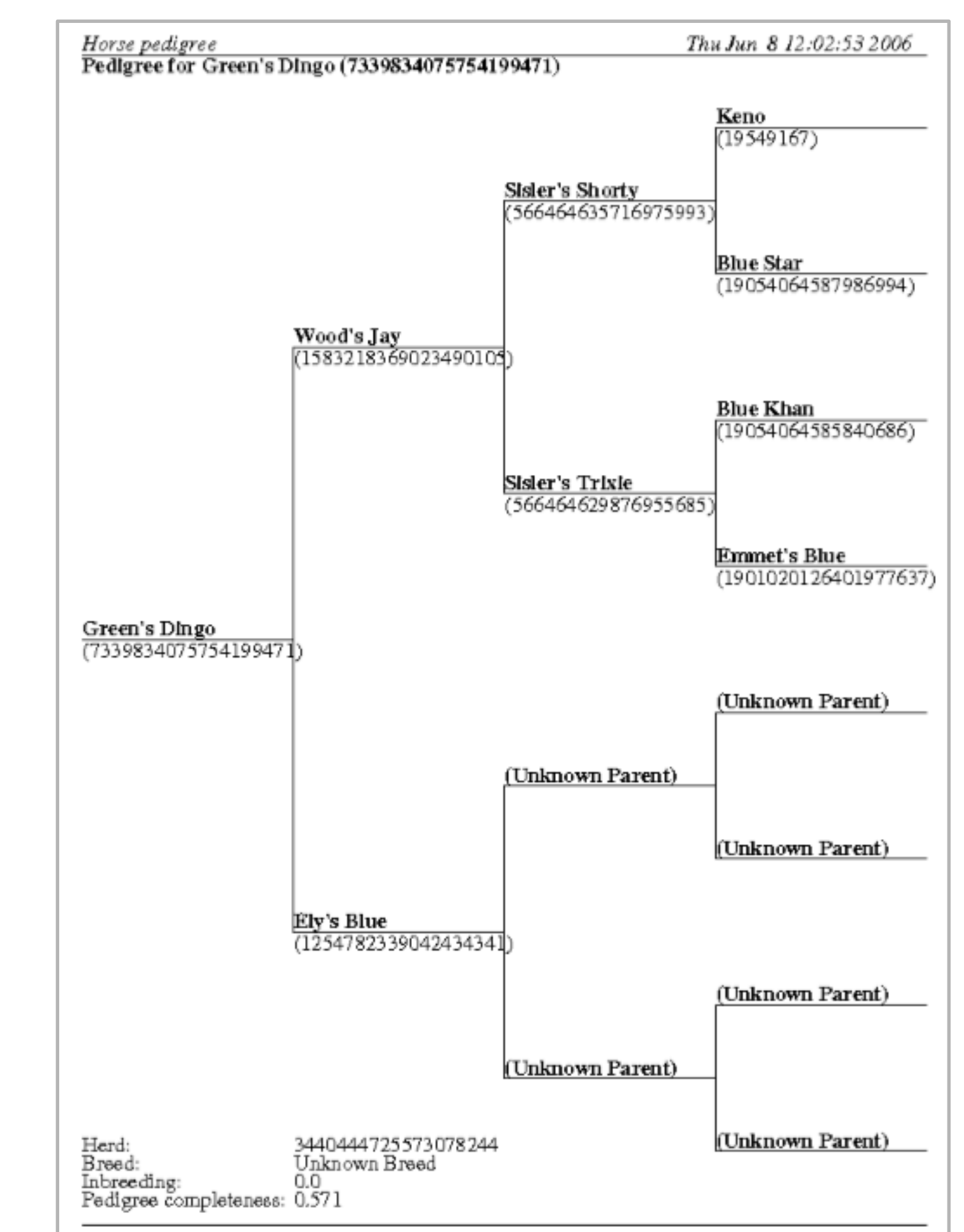
Figure 2. Average inbreeding of US Ayrshires by birth year.



Presentation of Results

- Quantitative data can be visualized (Figure 2)
- Printed reports can be prepared (Figure 3)
 - Templates are provided for user-created reports

Figure 3. Sample page from a three-generation pedigree book.



Other Features

- New methods to take **unions** and **intersections** of pedigrees
 - $A \cup B$ = unique animals in A or B or both
 - $A \cap B$ = unique animals common to A and B
- Other operations, such as subtraction, can be defined using these functions
 - $A - B = A - (A \cap B)$

```
>>> difference = pedigree1 - pedigree2
```
 - $A + B = A \cup B$

```
>>> sum = pedigree1 + pedigree2
```

Website and Documentation

- Website: <http://pypedal.sourceforge.net/>.
- Cole, J.B. 2007. PyPedal: A computer program for pedigree analysis. *Comp. Electron. Agric.* 57:107–113.