

Python Biax Data Reader: `biaxread`

J.R. Leeman

June 20, 2013

1 Purpose

This script was designed to read data into an easy to use format from the text output of xlook. Data is read, the header parsed for column names, lengths, etc., and a rec array returned that is easy to access and call from within a script. Column names are used to call the data columns, so as long as consistent naming is used the column order in the file is irrelevant.

2 Description

biaxread first opens the text file with a standard open command. We know that each column in the data is written as 12 characters wide. The first line of the header is the number of records, the second is the column number, the third the column headings, the fourth the column units, and the fifth the number of records in each column. This information is parsed and stored. A rec array is created with the numpy package and the labels set of the column names that were parsed. Finally the array is populated with data and the final product returned to the calling script.

A new set of improvements would be to add a cutting tool or a way to interface the script with the Pandas data handling library.

3 Cautions

A few cautions should be observed when using the biaxread script:

1. The empty array is shaped by row number information from the header. If a datafile is cut, but the header is left unmodified there will be many extra zero data pairs at the end of the array.

4 Usage

First process the experimental data in xlook and output a text file with headers. To do this input *type 0 -1 1 12 p4074_data.txt* at the xlook command line or add it to the data reduction file. Be sure to correct any column or row number specifications here. Start IPython or open your own python script. Import biax read and pass the function ReadBiax the name of the text output file from xlook. The array is now returned and ready to use.

5 Acknowledgements

Thank you to Marco Scuderi for using the code and being patient as little issues were worked out. This code is free for all to use with proper acknowledgement. Please send any bug reports to kd5wxb@gmail.com.

```

Last login: Thu Jun 20 09:42:11 on ttys001
[dkdot1x-10:~] jleeman% cd /Users/jleeman/Dropbox/PennState/BiaxExperiments/p4074/
[dkdot1x-10:PennState/BiaxExperiments/p4074] jleeman% ipython --pylab
Enthought Python Distribution -- www.enthought.com

Python 2.7.3 |EPD 7.3-2 (32-bit)| (default, Apr 12 2012, 11:28:34)
Type "copyright", "credits" or "license" for more information.

IPython 0.12.1 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

Welcome to pylab, a matplotlib-based Python environment [backend: WXAgg].
For more information, type 'help(pylab)'.

In [1]: from biaxread import ReadBiax

In [2]: my_data = ReadBiax('p4074_data.txt')

Number of records: 118941
Number of columns: 10

-----
|          Name|          Unit|          Records|
-----
|      LP_Displ|          mic|          118941|
|      Shr_stress|         MPa|          118941|
|      nor_displ|        micron|          118941|
|      Nor_stress|         MPa|          118941|
|          Time|         sec|          118941|
|      ec_displ|          mic|          118941|
|      Samp_Freq|         Hz|          118941|
|          mu|          .|          118941|
| layer_thick|        micron|          118941|
-----

In [3]: plt.plot(my_data['LP_Displ'], my_data['Shr_stress'])
Out[3]: [matplotlib.lines.Line2D at 0x95c6ed0]

```

6 Code

```
1 import numpy as np
2 import sys
3
4 def ReadBiAx(f_name):
5     """
6     Takes a filename containing the text output (with headers) from xlook and
7     reads the columns into a rec array for easy data processing and access.
8     """
9
10    filename = f_name
11
12    datafile = open(filename, 'r')
13
14    col_width = 12 # Columns are 12 char wide in header
15
16    # First line of the file is the number of records
17    num_recs = datafile.readline()
18    num_recs = int(num_recs.strip('number of records = '))
19    print "\nNumber of records: %d" %num_recs
20
21    # Second line is column numbers, we don't care so just count them
22    num_cols = datafile.readline()
23    num_cols = num_cols.split('col')
24    num_cols = len(num_cols)
25    print "Number of columns: %d" %num_cols
26
27    # Third line is the column headings
28    col_headings_str = datafile.readline()
29    col_headings_str = col_headings_str[5:-1]
30    col_headings = []
31    for i in xrange(len(col_headings_str)/12):
32        heading = col_headings_str[12*i:12*i+12].strip()
33        col_headings.append(heading)
34
35    # Fourth line is column units
36    col_units_str = datafile.readline()
37    col_units_str = col_units_str[5:-1]
38    col_units = []
39    for i in xrange(len(col_units_str)/12):
40        heading = col_units_str[12*i:12*i+12].strip()
41        col_units.append(heading)
42    col_units = [x for x in col_units if x != '\n'] #Remove newlines
43
44    # Fifth line is number of records per column
45    col_recs = datafile.readline()
46    col_recs = col_recs.split('recs')
47    col_recs = [int(x) for x in col_recs if x != '\n']
48
49    # Show column units and headings
50    print "\n\n-----"
51    print "|%15s|%15s|%15s|" %('Name', 'Unit', 'Records')
52    print "-----"
53    for column in zip(col_headings, col_units, col_recs):
```

```

54     print "|%15s|%15s|%15s|" %(column[0],column[1],column[2])
55 print "-----"
56
57 # Read the data into a numpy recarray
58 dtype=[]
59 dtype.append(('row_num','float'))
60 for name in col_headings:
61     dtype.append((name,'float'))
62 dtype = np.dtype(dtype)
63
64 data = np.zeros([num_recs,num_cols])
65
66 i=0
67 for row in datafile:
68     row_data = row.split()
69     for j in xrange(num_cols):
70         data[i,j] = row_data[j]
71     i+=1
72 data_rec = np.rec.array(data,dtype=dtype)
73
74 datafile.close()
75
76 return data_rec

```
