doi: 10.17221/64/2017-CJAS

RE-NUM-OR: Python-based Renumbering and Reordering Software for Pedigree Files

*Kemal Yazgan**

Department of Animal Science, Faculty of Agriculture, Harran University, Sanliurfa, Turkey

**Corresponding author: kyazgan@harran.edu.tr, kemalyazgan@gmail.com*

ABSTRACT

Yazgan K. (2018): **RE-NUM-OR: Python-based renumbering and reordering software for pedigree files**. Czech J. Anim. Sci., 63, 70–77.

RE-NUM-OR is a new, flexible, and user-friendly renumbering and reordering data arrangement software for unprocessed pedigree files used in genetic evaluation systems for personal computers. RE-NUM-OR was written using Python (Ver. 2.7.13) 64 byte programming language and compiled with PyInstaller-3.2.1 software which is a set of utilities for freezing Python scripts into executable. RE-NUM-OR not only runs on 32 or 64 byte MS Windows but also runs on 64 byte GNU/Linux. The program has new, practical, and simple interface and the user does not need to create a parameter file for running the program and .txt, .xlsx or .et extension files can be used as input files directly. Output files (.txt, .xlsx, .et, .dat) can be obtained. For animal, sire and dam IDs, it can read all standard characters (ASCII codes from 32 to 126) in input files. The program supports both dot (.) and comma (,) for numerical data. Pedigree lines for parents do not need to be arranged prior to their progeny. If pedigree lines for parents follow their progeny, the program can detect this and it can reorder the animals as pedigree lines placing parents prior to their progeny. Another feature of this software is that it has a pop-up window with error notification. Also, it supports repeated observations. RE-NUM-OR executable, its user manual, and sample input files are available from www.kemalyazgan.com.tr.

Keywords: pedigree renumbering; pedigree reordering; data arrangement

A number of software programs for genetic evaluation and pedigree analysis of farm animals have already been developed. MTDFREML (Boldman et al. 1995), MTGSAM (Van Tassel and Van Vleck 1995), DFREML (Meyer 1997), WOMBAT (Meyer 2007), ASREML (Gilmour et al. 2015), BLUPF90 family (Misztal et al. 2016), and MIXBLUP (Mulder *et al.* 2012) are the most popular software packages to obtain (co)variance components and genetic parameters from animal populations using various algorithms such as REML, Gibbs sampling, MIVQUE, etc. In addition to this, a number of software packages measure genetic diversity, inbreeding etc. in the population using pedigree records: CFC (Sargolzaei et al. 2006), ENDOG (Gutierrez and Goyache 2005), and Pedig (Boichard 2002), INBUPGF90 (Aguilar and Misztal 2008), PyPedal (Cole 2007). However, there are some limitations for these programs. For example, MTDFREML, MTGSAM, DFREML, CFC, and ENDOG are limited to a single operating system (MS Windows). On the other hand, MTDFREML, MTGSAM, DFREML, WOMBAT, ASREML, BLUPF90 family and MIXBLUP software need processed pedigree, data and parameter files.

In an unprocessed pedigree file obtained from a farm directly, animals, sires and dams can be recorded with ear tag numbers in such a way that the numbers assigned for parents can follow their offspring in the listing of animal. Also, animal tag numbers can include some abbreviations such as

doi: 10.17221/64/2017-CJAS

DE, TR, USA, etc. representing the country or other animal characteristics.

Unprocessed data cannot be used directly with the above software and the unprocessed data and pedigree set must be rearranged and processed in various ways. For MTDFREML, MTGSAM, DFREML, and WOMBAT programs animal IDs must be only integers and all animals must have a numerically higher code than either of their parents. On the other hand, for some programs such as ASREML and MIXBLUP, animal IDs cannot only be numeric but also alphanumeric. There is a special software named Renumf90 in BLUPf90 family and it can renumber the animal IDs from 0 to *n* and similar to ASREML, Renumf90 supports the alphanumeric characters in animal IDs but Input files cannot contain the character "#". MIXBLUP supports alphanumerical and numerical data but animal IDs in the pedigree file must be of the same type as the animal ID in the data file (either numeric or text).

However, for the software mentioned above, pedigree lines for parents must precede their progeny. Otherwise, none of them runs. This situation can be a big problem for a large data set. Because in an unprocessed pedigree and data file thousands of animals can be recorded in this way and they must be sorted/reordered with other data such as fixed effect, random effect, genetic marker inheritance, single nucleotide polymorphism (SNP) information, and yield. So as to overcome this problem easily, the RE-NUM-OR software has been developed by the present author.

Python is a general-purpose, open-source, and powerful object-oriented programming language (www.python.org). It is optimized for software quality, developer productivity, program portability, and component integration. Python is used by hundreds of thousands of developers around the world in areas such as internet scripting, system programming etc. It is generally considered to be among the top four or five most widely-used programming languages in the world today (Lutz 2011).

The aim of this study was to introduce the pedigree-sorting software RE-NUM-OR for researchers focusing on genetic analysis of traits of farm animals by using software as e.g. MTDFREML, MTGSAM, WOMBAT, ASREML, and MIXBLUP.

MATERIAL AND METHODS

RE-NUM-OR is written using Python (Version 2.7.13) programming language (www.python.org) and compiled with PyInstaller-3.2.1 software, which is a set of utilities for freezing Python scripts into executable programs (http://www.pyinstaller.org/). As shown in Table 1, six different packages/modules were used in this program. RE-NUM-OR runs on 32 or 64 byte MS Windows and GNU Linux operating systems and was tested under MS Windows 7, 8.1, and 10, and Linux Mint 18.1. RE-NUM-OR has new, practical, and simple interface (Figure 1). There is no need for parameter or data files to be created to run the program. This software is available as an executable program by

Table 1. Packages and modules used in RE-NUM-OR

Package/Module	Explanation	Version	Reference
NumPy (Numerical Python)	NumPy: array processing for numbers, strings, records, and objects	1.11.2rc1	http://www.numpy.org
Pandas	data structures for data analysis, time series, and statistics	0.18.1	http://pandas.pydata.org
Xlrd	library for developers to extract data from MS Excel (tm) spreadsheet files	1.0.0	http://www.python-excel.org/
XlsxWriter	for writing files in MS Excel 2007 (or later) + .xlsx file format	0.9.3	https://github.com/jmcnamara/XlsxWriter
Tkinter	Tkinter module ("Tk interface") is the standard Python interface to Python Tk GUI toolkit (written in Python)	Python 2.7.12	https://www.python.org
Os	this module provides a portable way of using operating system dependent functionality (written in Python)	Python 2.7.12	https://www.python.org

doi: 10.17221/64/2017-CJAS

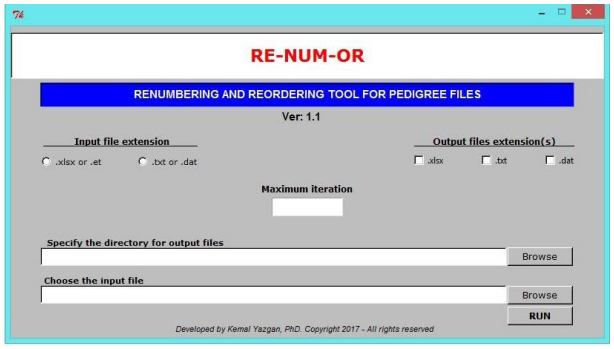

Figure 1. RE-NUM-OR main window

downloading from www.kemalyazgan.com.tr. Also, user manual and some examples of unprocessed input files are available from this web page.

MS Office 2007[1] or later versions must be installed to use an .xlsx extension file. Instead of this, WPS Office 2016, a free office package available from http://wps.com/office-free, can be used. Spreadsheet files with the extension ".et" can be an alternative to MS Excel. RE-NUM-OR can support both these file formats. Also, data files can be saved as .xlsx files.

RE-NUM-OR program can generate a new pedigree file (xlsx, txt or dat) from an unprocessed .xlsx or .txt extension file (mentioned in the introduction). In this file all animals are renumbered from 1 to *n* and, always a new animal number is bigger than its sire and dam numbers. Also, if necessary, pedigree lines can be sorted to make parents always precede their offspring in the list of animal IDs. Thanks to its iterative algorithm, this program can solve it easily in a number of iterations. Additionally, if the user wants to process the unprocessed pedigree lines by adding the individual animal data such as fixed effects, random effects, genetic marker inheritance, SNP information, and yields, RE-NUM-OR can handle this easily without the need of complex parameter files creating. The resulting file can be used directly for pedigree or genetic analyses (calculation of breeding values of animals, coefficient of relationship, etc.) with some software as mentioned above.

In Figure 2 various ID forms are shown. In practice, 2# # #, [2344488899\$_&*], @/t or s1## are meaningless IDs but, in order to show the flexibility and capability of RE-NUM-OR, these meaningless character groups were accepted as animal IDs and they were used here. Animals shown in coloured entries under Animal-ID column are misplaced, coming below their offspring (Figure 2). If this was the case, the associated rows including all entries (animal, sire, dam IDs, parity, season fat

	A	B	C	D	E	F	G
1	Animal_ID	Sire_ID	Dam_ID	lactation parity	season	fat yield	milk yield
2	TR012000065896822278045	B99444	[2344488899\$_&*]	1	3	0.06	4859.98
3	TR012000065896822278045	B99444	[2344488899\$_&*]	2	3	0.045	4751.9
4	TR012000065896822278045	B99444	[2344488899\$_&*]	3	4	0.04	3877
5	99644	85	0	1	2	0.025	4789
6	E_58/56800-069	BULL-128	<985ASDEU7891>	1	4	0.014	3987
7	E_58/56800-070	BULL-128	<985ASDEU7891>	2	2	0.054	3987
8	BULL-18	FR987855	C896669974444503	1	2	0.04	4178.88
9	s1##	0	0	1	3	0.0668	9888
10	98	99644	0	1	4	0.039	2987
11	Akm015	10360089	0	1	1	0.047	1299
12	B99444	103600	0	1	3	0.067	3687
13	COW-21	10360091	99644	1	4	0.047	9878
14	[2344488899\$_&*]	10360091	E_58/56800-070	1	2	0.025	4789
15	85	10360093	0	1	4	0.01	1478
16	BULL-128	10360094	0	1	3	0.019	5478
17	@/t	5899	0	1	1	0.089	6247
18	2# # #	BULL-18	0	1	1	0.4169	4880
19	<985ASDEU7891>	80	0	1	3	0.04	4899.8
20	DE52007963	6	0	1	2	0.0147	3258
21	DE15688978	898/987	TR012000065896822278045	1	4	0.047	4406

renumor Sheet2 Sheet3

Figure 2. An unprocessed .et input file for RE-NUM-OR. Coloured animals follow their progeny in "Animal ID" column

[1]In this study references or citations of all commercial or noncommercial software are made with the understanding that no advertising and discrimination is implied and no endorsement by the author is intended.

doi: 10.17221/64/2017-CJAS

yield, and milk yield) are moved down so that the lines are matched. For example, rows 2, 3, and 4 (the same animal and repeated observation) must move between the lines of the entries B9944 and [2344488899\$_&*], because the animal numbered as B99444 is progeny of the sire animal numbered as 103600 and an unknown dam. Thus, the entry B9944 does not follow its progeny (TR0120000065896822278045). All coloured lines must be reordered in the same way. This file can be a good example for an unprocessed data obtained from a ranch. It can be used as an input file for RE-NUM-OR directly.

To carry out reordering of animal entries as mentioned above, a recursive method was developed using Python codes. In this method, all animal entries, which are coming below their offspring, are moved one by one and placed above their offspring. The code may need a number of iterations during this reordering process. Iterating continues until all animals entries are reordered as placed above their offspring. The codes consist of a function (termed as "solution") and a "while" loop as shown below:

```
def solution(x1, x2, x3, x4):
  m1=[]
  m2=[]
  for i, j, k in zip(x1, x2, x3):
      if j > 0 and j in x3:
         window = Tk()
         window.wm_withdraw()
         window.geometry("1 x 1 + 200 + 200")
         tkMessageBox.showerror(title=" ERROR ",
         message=" ... ",parent=window)
         stop
      if k > 0 and k in x2:
         window = Tk()
         window.wm_withdraw()
         window.geometry("1 x 1 + 200 + 200")
         tkMessageBox.showerror(title=" ERROR ",
         message=" ... ", parent=window)
         stop
      if i in x3:
         s1=x1.index(i)
         s2=x3.index(i)
      if s1 > s2:
         m1.append(s1)
         m2.append(s2)
      if i in x2:
         s3=x1.index(i)
         s4=x2.index(i)
      if s3 > s4:
         m1.append(s3)
         m2.append(s4)
  for i, j in zip(m1, m2):
      old_index = i
      x1.insert(j, x1.pop(old_index))
  for i, j in zip(m1, m2):
      old_index = i
      x2.insert(j, x2.pop(old_index))
  for i, j in zip(m1, m2):
      old_index = i
      x3.insert(j, x3.pop(old_index))
  for i, j in zip(m1, m2):
      old_index = i
      x4.insert(j, x4.pop(old_index))
  return x4, x1, x2, x3
count = 0
n = 1
while count < maximum_ iteration:
  f=solution(anim, sire, dam, order)
  d1=f[1]
  d2=f[2]
  d3=f[3]
  g1=[]
  g2=[]
  for i, j, k in zip(d1, d2, d3):
      if i in d2 and d1.index(i) > d2.index(i):
         s1=1
         g1.append(s1)
      else:
         s=0
         g1.append(s1)
      if i in d3 and d1.index(i) > d3.index(i):
         s2=1
         g2.append(s2)
      else:
         s2=0
         g2.append(s2)
  t1=sum(g1)
  t2=sum(g2)
  h= t1+ t2
  n = n + 1
  count = count + 1
  if h > 0:
         print n-1,'Round: Program could not reorder
         the animal IDs. Retrying... '
  if h > 0 and count == maximum_ iteration:
         window = Tk()
         window.wm_withdraw()
         window.geometry("1 x 1 + 200 + 200")
         tkMessageBox.showerror(title=" ERROR ",
         message="Please enter bigger value to maxi-
         mum iteration box!!! ", parent=window)
         stop
  if h==0:
         print ''
         print'===ANIMAL IDS WERE REORDERED
         SUCCESSFULLY ! ! ! ==='
         break
```

doi: 10.17221/64/2017-CJAS

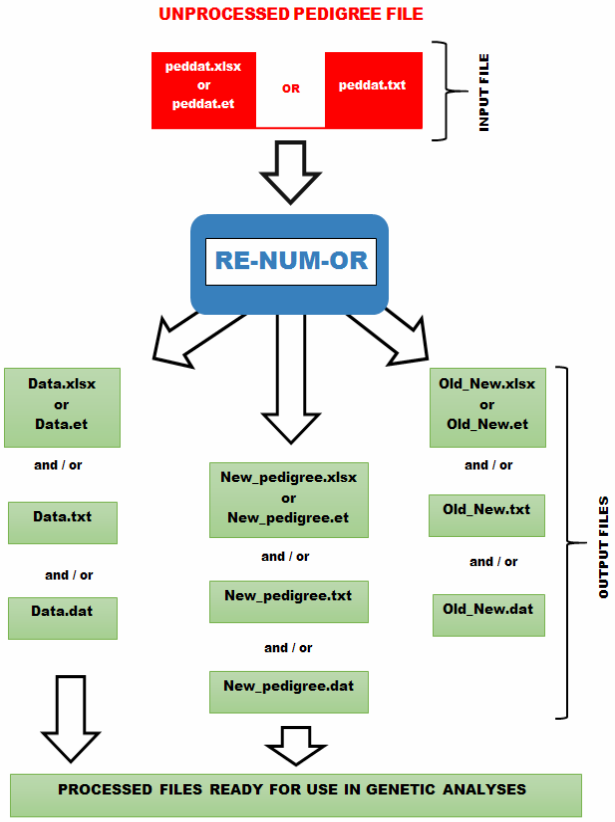

Figure 3. Flow diagram of RE-NUM-OR

RE-NUM-OR supports both dot (.) and comma (,) in input files. So, for all input file types the user does not need to convert a comma (,) to a dot (.) or a dot to a comma, even when both separator types are used in the same input file. In addition to this, all characters supported by RE-NUM-OR, which can be used in "joint pedigree and data" file, are given in Table 4.

For saving all output files created by the software, the user must specify the working directory (Figure 1). When the program terminates, all output files are saved into this user-selected folder. As shown in Figure 3, when RE-NUM-OR terminates, three files (Old_new, New_pedigree, and Data) are generated for the selected file format; their contents are described below.

Old_new.xlsx, Old_new.txt or Old_new.dat. These files are produced by RE-NUM-OR in order to list old (actual) and new (renumbered) IDs side by side. First and second columns (A and B) in Old_new.xlsx are actual IDs and renumbered IDs of animals, respectively. Similar to this file, in Old_new.txt and Old_new.dat the first and second columns are actual and renumbered IDs, respectively.

Table 2. RE-NUM-OR error notifications and their explanations

No.	Error notification text	Explanations
(1)	At least one animal recorded as both father and mother in pedigree file	Please open your input file and make the occurrences of that animal either as a sire or as a dam as unknown. If both parents of an animal are recorded as empty cells in an .xlsx extension input file, this error occurs, too. Because, for .xlsx extension files, empty cells are detected as ID by RE-NUM-OR. As mentioned earlier, missing observations must be coded as 0.
(2)	Please enter bigger maximum iteration value	If RE-NUM-OR cannot find the solution within the number of iterations given by the user, this message is shown. So, the user must try the program with a bigger maximum iteration number.
(3)	Out of order input file	RE-NUM-OR supports repeatable observations of animal data. However, if an animal ID is recorded with different sire or dam ID in its repeated observations or if an animal ID overlaps with its sire or dam ID, results with this error Various mistakes in .xlsx extension input file such as some characters unspecified in Table 3, results with this error, too.
(4)	Incompatible or out of order input file type	For example, if the user specifies an .xlsx extension input file using radio button and then selects a .txt or .dat extension input files using browse button, this error occurs. The specified file must be compatible with the selection of input file type. Similarly, choosing other files unsupported by RE-NUM-OR causes this error, too. Another cause of this error is some editing mistakes in .txt or .dat extension input files. Different from .xlsx files, space cannot be detected as ID by RE-NUM-OR. If the user prefers to use .txt or .dat files, animal/sire/dam ID must not be space. Otherwise, this error occurs. Similarly as in No. (3), except specified in Table 3, users must not use other unsupported characters.

doi: 10.17221/64/2017-CJAS

Table 3. RE-NUM-OR warning notifications and their explanations

No.	Warning notification text	Explanations
(1)	Please specify the input file type	Please select input file format (.xlsx, .et or .txt).
(2)	Please select the input file. It must be compatible with input file type	It must be compatible with input file type. Otherwise, program does not work.
(3)	Please specify the working directory for output files	Please select a folder. RE-NUM-OR saves the all output files chosen by user in this folder.
(4)	At least one file type (.xlsx, .txt or .dat) must be chosen	Please select at least one file type.
(5)	Iteration number must be integer and bigger than zero	If the user enters anything except a positive integer value, this error occurs.

New_pedigree.xlsx, New_pedigree.txt or New_pedigree.dat. This new pedigree file consists of renumbered and reordered animal records. In this new file, the new animal number is bigger than the new sire and dam numbers and pedigree lines for parents precede those of their progeny. Regardless of the file type, it contains three columns: animal number, sire number, and dam number. Apart from these three columns, this file contains nothing else.

Data.xlsx, Data.txt or Data.dat. These files include new sorted pedigree and data. Also, a work file named 01.xlsx is created. Software uses this file for some computing process.

How large can a pedigree or data set be? If the user prefers to use .txt or .dat type file for input, the number of pedigrees depends on their computer memory and disc space. In other words, there is no limit in the RE-NUM-OR for the number of pedigrees. However, if the user prefers to use .xlsx or .et file types, the limit is that of these file row number. For example, MS Excel 2007 supports the maximum of 1 048 576 rows in a sheet.

Error and warning messages. RE-NUM-OR has the error and warning notification pop-up windows (box). They contain the information about the occurrence of the error and about its nature. For example, if an animal is recorded as both dam and sire in the input file, the error message appears (Figure 4). Another example: if the user forgets to click the appropriate boxes concerning the extensions of file names (Figure 1), RE-NUM-OR can easily perceive this lack and notify the user of this with a warning message this time, so the user can identify the problem. The various error and warning messages that users may encounter while using the program and their explanations are given in Tables 2 and 3.

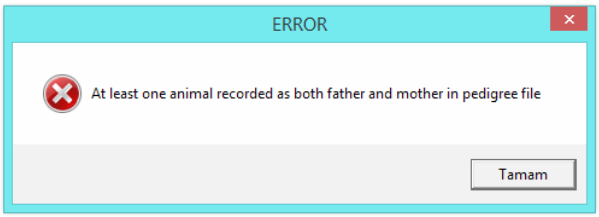

Figure 4. An example error notification pop-up window

RESULTS AND DISCUSSION

As previously mentioned, an unprocessed data set cannot be used in genetic analysis software. Using this software requires some adjustments in the data and pedigree set. RE-NUM-OR is a utility to be used with genetic analysis programs. It has new and flexible features and it can run without

Table 4. Characters supported by RE-NUM-OR (corresponding to ASCII codes from 32 to 126)

0	(space)	@	P	`	p
1	!	A	Q	a	q
2	“	B	R	b	r
3	#	C	S	c	s
4	$	D	T	d	t
5	%	E	U	e	u
6	&	F	V	f	v
7	‘	G	W	g	w
8	(	H	X	h	x
9	)	I	Y	i	y
:	*	J	Z	j	z
;	+	K	[	k	{
<	,	L	\	l	|
=	-	M	]	m	}
>	.	N	^	n	~
?	/	O	_	o	

doi: 10.17221/64/2017-CJAS

parameter files, it also supports using .xlsx, .et, .dat, and .txt for input and output files. The user can use both dot (.) and comma (,) for numerical data in input files. If pedigree lines for parents follow their progeny, it can reorder animals so that pedigree lines for parents are prior to their progeny and it can arrange this easily for not only single but also repeated observations per animal.

For now, RE-NUM-OR can read all standard characters (ASCII codes from 32 to 126) from input files. However, it is planned to extend the program by adding more supported ASCII codes in the software. Thus, some country-specific characters will be read by RE-NUM-OR.

With these features, the program can sort large data sets used in genetic analysis software such as MTDFREML, MTGSAM, WOMBAT, ASREML, and MIXBLUP without additional processing. It also runs under a number of operating systems, including 64 bit GNU/Linux and 32/64 bit MS Windows.

RE-NUM-OR is not a pedigree analysis software. It is a useful and flexible tool for renumbering and rearranging large and complex pedigree files. So, comparing this program to other software mentioned in this study (e.g. CFC (Sargolzaei et al. 2006), ENDOG (Gutierrez and Goyache 2005), Pedig (Boichard 2002), INBUPGF90 (Aguilar and Misztal 2008), and PyPedal (Cole 2007)) cannot be a valid approach. It can be used as a utility with genetic and pedigree analysis software. RE-NUM-OR may help overcome drawbacks of such programs. RE-NUM-OR may complete the deficiencies of some software which is running for pedigree and genetic analyses.

While CFC and ENDOG are limited to a single MS Windows operating system, RE-NUM-OR can be used under 32/64 bit MS Windows and 64 bit GNU Linux systems. This is an advantage similar to that of PyPedal. However, a user cannot use Excel files directly with PyPedal. While PyPedal animal IDs may be provided as either integers or character strings, both formats can be used in the same pedigree file in RE-NUM-OR (Figure 2). PyPedal software has some renumbering and reordering functionalities. But, for PyPedal the renumbering process requires that the pedigree file finally be reordered such that parents always precede their offspring. The actual ID assigned to an animal is of no particular importance, and it is even possible for parents to have larger IDs than their offspring. But, unless there is an error in it that would prevent unambiguously placing parents before their offspring, PyPedal software can reorder the pedigree files.

For INBUPGF90 program, a pedigree file does not need to be in a particular order and the alphanumeric identification of individuals is supported. However, the supported character set is not as rich as RE-NUM-OR. Also, MS Excel files cannot be used directly with this program. Also, if the user does not enter birth years as a separate column in the pedigree file, some problems may occur.

In comparison to the ability of RE-NUM-OR to accept input data from the software such as MS Excel or WPS Office, ENDOG accepts only .xls files (from MS Excel worksheets). Also, it does not support .xlsx files. If a data set is longer than 65 536 lines, it cannot be stored in an .xls file. Next, in input Excel worksheets for ENDOG the cells must contain only IDs with standard characters. Additionally, if records are not ordered, all the individuals must have a consistent birth date. This is not mandatory for RE-NUM-OR. Users can input any valid Excel contents beside the pedigree columns, and they do not have to enter birth dates of animals.

Similarly to the software mentioned above, Renumf90 supports alphanumeric characters in animal IDs but input files must not contain the character "#". MIXBLUP supports the alphanumerical and numerical data but animal IDs in the pedigree file must be of the same type as the animal ID in the data file (either numeric or text).

CONCLUSION

There are some limitations in pedigree and genetic analysis programs concerning the arrangement of the pedigree file. In most of the pedigree analysis software to arrange the pedigree files the users must either enter birth dates or reorder the animals themselves. Sometimes a user would prefer alphanumeric string or integer format (or both) for the animal ID. RE-NUM-OR uses iterative algorithm and is capable of using MS Excel or WPS Office worksheets, user-friendly interface and error message pop-up window, so it can be helpful for researchers.

Python is a general purpose, powerful, object-oriented programming language and this study

doi: 10.17221/64/2017-CJAS

also demonstrated and confirmed that it can be an alternative to Fortran in meeting programming needs of animal breeding disciplines.

REFERENCES

Aguilar I., Misztal I. (2008): Recursive algorithm for inbreeding coefficients assuming non-zero inbreeding of unknown parents. Journal of Dairy Science, 91, 1669–1672.

Boichard D. (2002): PEDIG: A Fortran package for pedigree analysis suited for large populations. In: Proc. 7th World Congress on Genetics Applied to Livestock Production, Montpellier, France, Comm. No. 28-13.

Boldman K.G., Kriese L.A., Van Vleck L.D., Van Tassel C.P., Kachman S.D. (1995): A Manual for Use of MTDFREML: A Set of Programs to Obtain Estimates of Variances and Covariances. United States Department of Agriculture, Agriculture Research Service, Washington, USA.

Cole J.B. (2007): PyPedal: A computer program for pedigree analysis. Computers and Electronics in Agriculture, 57, 107–113.

Gilmour A.R., Gogel B.J., Cullis B.R., Welham S.J., Thompson R. (2015): ASReml. User Guide Release 4.1, Structural Specification. VSN International Ltd, Hemel Hempstead, UK.

Gutierrez J.P., Goyache F. (2005): A note on ENDOG: a computer program for analysing pedigree information. Journal of Animal Breeding and Genetics, 122, 172–176.

Lutz M. (2011): Programming Python. O'Reilly Media, Inc., Sebastopol, USA.

Meyer K. (1997): DFREML Version 3. User Notes. Available at http://didgeridoo.une.edu.au/km/dfreml.php (accessed May 2, 2017).

Meyer K. (2007): WOMBAT. A tool for mixed model analyses in quantitative genetics by restricted maximum likelihood (REML). Journal of Zhejiang University – SCIENCE B, 8, 815–821.

Misztal I., Tsuruta S., Lourenco D., Masuda Y., Aguilar I., Legarra A., Vitezica Z. (2016): Manual for BLUPF90 Family of Programs. University of Georgia, Athens, USA.

Mulder H.A., Lidauer M., Stranden I., Mantysaari E.A., Pool M.H., Veerkamp R.F. (2012): MiXBLUP Manual. Animal Breeding and Genomics Centre, Wageningen UR, Livestock Research, Lelystad, the Netherlands.

Sargolzaei M., Iwaisaki H., Colleau J.J. (2006): CFC: A tool for monitoring genetic diversity. In: Proc. 8th World Congress on Genetics Applied to Livestock Production, Belo Horizonte, Brazil, 27–28.

Van Tassel C.P., Van Vleck L.D. (1995): A Manual for Use of MTGSAM. A Set of Fortran Programs to Apply Gibbs Sampling to Animal Models for Variance Component Estimation. United States Department of Agriculture, Agriculture Research Service, Washington, USA.

Received: 2017–06–10
Accepted after corrections: 2017–12–04