IBDLabels Package

Fiona Grimson <fgrimson@uw.edu>

February 14, 2015

This package is intended for reference in converting between different IBD labelling schemes. All conversions are to or from the “label” as used by MORGAN programs. Conversions with the vector representation of the state, lexicographic indexing, and jacquard indexing are included.

The installation of the package and the usage of its main functions is described here.

1 Installation

To install this package, open R and type

```
> install.packages( "IBDLabels", repos = "http://R-Forge.R-project.org")
```

To open the package, type

```
> library("IBDLabels")
```

To open this tutorial within R, type

```
> vignette("IBDLabels_tutorial")
```

List all the functions available with the command

```
> ## list everything in the package
> ls( "package:IBDLabels", all=TRUE)

 [1] "allJaq"    "allLex"    "allStates" "allVec"    "fgl2vec"   "jaq2label"
 [7] "label2jaq" "label2lex" "label2vec" "lex2label" "maxlabel"  "vec2label"

> ## list all functions and their arguments
> lsf.str("package:IBDLabels")
```

```
allJaq : function ()
allLex : function (ngam)
allStates : function (ngam)
allVec : function (ngam)
fgl2vec : function (vec)
jaq2label : function (jaq, phased = TRUE)
label2jaq : function (label, phased = TRUE)
label2lex : function (label, ngam)
label2vec : function (label, ngam)
lex2label : function (lex, ngam)
maxlabel : function (ngam)
vec2label : function (vec)
```

2 Label/Vector

The vector representation of the IBD state is a complete and unambiguous description of the IBD state. There is one vector element per gamete. The first gamete is always numbered one, with subsequent gametes being given the same number if they are IBD with a previous gamete, or a new number if they are not. For example in (1,2,2,3), the second and third gametes are IBD.

Functions are included for a lookup table for all labels and vectors, `allVec`, or to do individual conversions between vector and label and vice versa, `vec2label` and `label2vec`. The function `maxlabel` gives the maximum label for a certain number of gametes. *Note that the minimum label is always zero, and while all valid IBD states produce a unique label not all labels correspond to a valid IBD state.* In this package, invalid labels will be converted to NA.

When `vec2label` is called, the input vector is always converted to a valid IBD state vector by calling the function `fgl2vec`. This ensures that valid, invertible labels are always produced. The function can be called directly on fgl vectors to find the corresponding IBD label.

```
> ## Vectors for all labels, note the invalid IBD state produced from
> ## label 2 has NA.
> allVec( ngam = 4 )

   [,1] [,2] [,3] [,4]
0     1    1    1    1
1     1    1    1    2
2    NA   NA   NA   NA
3     1    1    2    1
4     1    1    2    2
5     1    1    2    3
```

```
6     1    2    1    1
7     1    2    1    2
8     1    2    1    3
9     1    2    2    1
10    1    2    2    2
11    1    2    2    3
12    1    2    3    1
13    1    2    3    2
14    1    2    3    3
15    1    2    3    4
```

```
> ## Convert vector to label, the vectors are always renumbered with fgl2vec.
> vec2label( c(1,1,1,3))
```

```
[1] 1
```

```
> vec2label( c(1,1,1,2))
```

```
[1] 1
```

```
> ## Convert label to vector
> label2vec( 1, ngam = 4 ) ## Coverts to valid vector
```

```
[1] 1 1 1 2
```

```
> label2vec( 2, ngam = 4 ) ## Converts to invalid vector
```

```
[1] NA NA NA NA
```

```
> ## renumbering
> fgl2vec( c( 1,1,1,3) )
```

```
[1] 1 1 1 2
```

```
> fgl2vec( c(5,1,5,6) )
```

```
[1] 1 2 1 3
```

```
> ## maximum label for given number of gametes.
> maxlabel( ngam = 4 )
```

```
[1] 15
```

```
> maxlabel( ngam = 5 )
```

```
[1] 64
```

```
> maxlabel( ngam = 6 )
```

```
[1] 325
```

3 Label/Lexicographic

Lexicographic ordering of states is the same as the label order, but with invalid labels removed. For example, in the four gamete case the label 2 would produce (1,1,1,3) which is not a valid IBD state. *Note that lexicographic indexing starts from 1, and labels start from 0.*

The functions `label2lex` and `lex2label` do conversions from label to lexicographic and vice versa. The functions `allLex` creates a reference vector of all lexicographic and label conversions and for doing individual conversions.

```
> ## Vector of all lexicograghic states with labels ( names of elements )
> ## labels are printed on the top line
> allLex( ngam = 4 )

 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 1  2 NA  3  4  5  6  7  8  9 10 11 12 13 14 15

> ## Convert lex to label
> lex2label( lex = 5 , ngam = 4 )

[1] 5

> ## Convert label to lex
> label2lex( label = 3, ngam = 4 )

3
3

> ## Invalid labels do not convert to lex
> label2lex( label = 2, ngam = 4)

 2
NA

>
```

4 Label/Jacquard

Jacquard indexing only applies to 4-gamete IBD states, and can be phased (15 states) or unphased (9 states). *Note that jacquard indexing starts from 1, and labels start from 0.*

```
> ## list all jacquard states
> allJaq()
```

```
      label jaq15 jaq9
 [1,]     0     1    1
 [2,]     1     3    3
 [3,]     2    NA   NA
 [4,]     3     4    3
 [5,]     4     2    2
 [6,]     5     5    4
 [7,]     6     6    5
 [8,]     7     9    7
 [9,]     8    11    8
[10,]     9    10    7
[11,]    10     7    5
[12,]    11    13    8
[13,]    12    12    8
[14,]    13    14    8
[15,]    14     8    6
[16,]    15    15    9
```

```
> ## Convert label to jacquard
> label2jaq( 3, phased = TRUE )

[1] 4

> label2jaq( 3, phased = FALSE )

[1] 3

> ## Convert jacquard to label
> jaq2label( 9, phased = TRUE )

[1] 7

> jaq2label( 9, phased = FALSE )

[1] 15
```

5 All conversions

We can create a lookup table of all the conversions with the allStates command.

```
> ## Lookup table of all states. Always puts NA for invalid.
> allStates( ngam = 4 )
```

```
   label vec.1 vec.2 vec.3 vec.4 lex jaq15 jaq9
1      0     1     1     1     1   1     1    1
2      1     1     1     1     2   2     3    3
3      2    NA    NA    NA    NA  NA    NA   NA
4      3     1     1     2     1   3     4    3
5      4     1     1     2     2   4     2    2
6      5     1     1     2     3   5     5    4
7      6     1     2     1     1   6     6    5
8      7     1     2     1     2   7     9    7
9      8     1     2     1     3   8    11    8
10     9     1     2     2     1   9    10    7
11    10     1     2     2     2  10     7    5
12    11     1     2     2     3  11    13    8
13    12     1     2     3     1  12    12    8
14    13     1     2     3     2  13    14    8
15    14     1     2     3     3  14     8    6
16    15     1     2     3     4  15    15    9

>
```

As a visual reference, I have included visual representations of the IBD states in Table 1 (jacquard ordering) and Table 2 (lexicographic/label ordering.

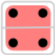
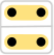
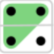
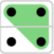
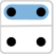
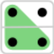
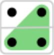
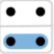
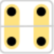
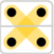
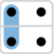
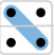
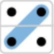
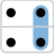
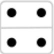
	Label	Lex	15-state Jacquard	9-state Jacquard	State Vector	No.IBD
	0	1	1	1	1111	2
	4	4	2	2	1122	0
	1	2	3	3	1112	1
	3	3	4	3	1121	1
	5	5	5	4	1123	0
	6	6	6	5	1211	1
	10	10	7	5	1222	1
	14	14	8	6	1233	0
	7	7	9	7	1212	2
	9	9	10	7	1221	2
	8	8	11	8	1213	1
	12	12	12	8	1231	1
	11	11	13	8	1223	1
	13	13	14	8	1232	1
	15	15	15	9	1234	0

Table 1: IBD state equivalences, ordered by Jacquard

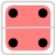
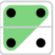
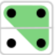
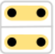
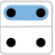
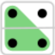
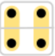
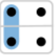
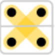
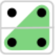
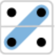
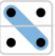
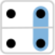
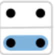
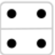
	Label	Lex	15-state Jacquard	9-state Jacquard	State Vector	No.IBD
	0	1	1	1	1111	2
	1	2	3	3	1112	1
	3	3	4	3	1121	1
	4	4	2	2	1122	0
	5	5	5	4	1123	0
	6	6	6	5	1211	1
	7	7	9	7	1212	2
	8	8	11	8	1213	1
	9	9	10	7	1221	2
	10	10	7	5	1222	1
	11	11	13	8	1223	1
	12	12	12	8	1231	1
	13	13	14	8	1232	1
	14	14	8	6	1233	0
	15	15	15	9	1234	0

Table 2: IBD state equivalences, ordered by Label