

R代码实现BOLD数据库数据批量下载及绘图

2021-7-3 首都师范大学 遗传多样性与进化实验室 王瑛 整理，有任何错误和不足，欢迎指出。

以下代码仅供大家参考，如有建议或意见，请邮件联系2200801021@cnu.edu.cn

R包的几点说明：

B包	作用
bold	基于BOLD数据的R操作，都得建立在这个R包的基础上
openxlsx	.xlsx格式文件的读取和写入都要依赖它
dplyr	一些特定的数据处理需要依赖它

如果library失败，请自行安装相应R包。{bold}R包的安装和详细使用请参考：<https://docs.ropensci.org/bold/>

问询序列存放在 species_list.xlsx 表格内的格式：下面的代码均使用这4个物种做测试。

	A	B
1	<i>Orthopygia glaucinalis</i>	
2	<i>Aglossa dimidiata</i>	
3	<i>Aglossa caprealis</i>	
4	<i>Papilio machaon</i>	
5		
6		

初次使用下面代码时，建议先使用和本教程一样的数据来测试代码在您设备上的可用性，确认没有问题后才测试您自己的数据。

测试物种有1个是BOLD数据库中没有收录的，还有一些是信息不全，缺少序列或者图片，所以代码运行过程中出现警告，不影响输出有结果物种的信息。

	A	B	C	D	E	F
1	测试物种	分类信息	坐标数据	图片url	GenBank ID	条码情况
2	<i>Orthopygia glaucinalis</i>	√	1个	1张	无	1条
3	<i>Aglossa dimidiata</i>	无	无	无	无	无条码
4	<i>Aglossa caprealis</i>	√	7个	10张	1个	10条
5	<i>Papilio machaon</i>	√	191个	40张	377个	624条

如果您只有一个物种需要查询，也是可以放在这个表格内的，如果没有相关信息，会提示NA。

代码结构

• 常规信息获取

- 1_除了序列外的数据.R
- 2_只要序列数据.R

- 获取分布数据绘制分布图

- 3_按科属种下载发布数据用于作图.R
- 4_物种批量出图.R

1. 常规信息获取

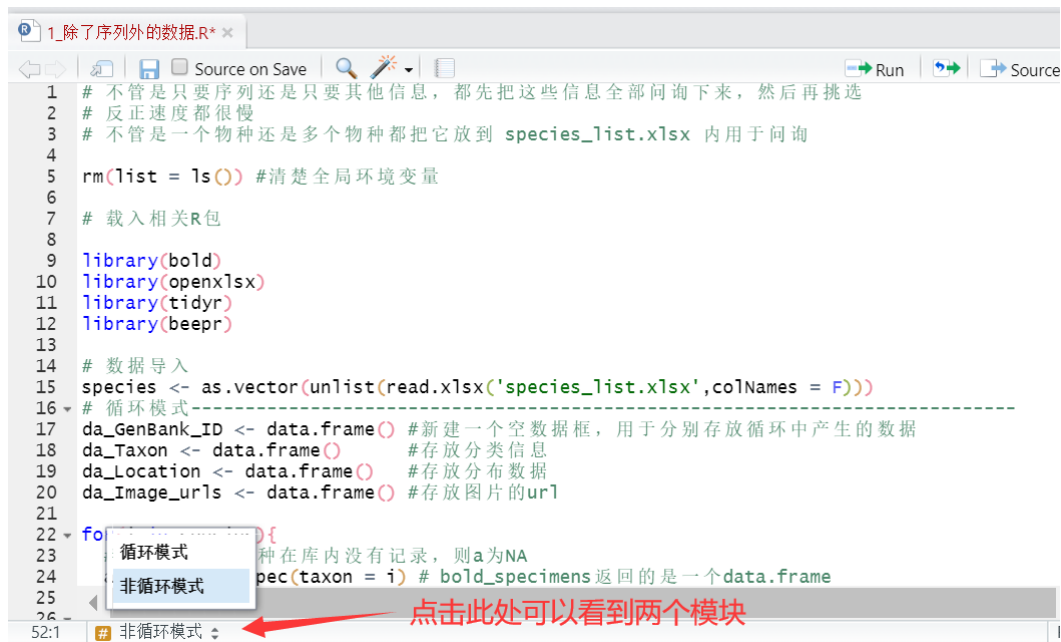
出于不同的数据下载需求，我将BOLD数据库常规信息获取的脚本分为两个部分：

1. 除了序列以外的数据，主要包括：分类信息、分布数据、图片的url
2. 只要序列数据

1_除了序列外的数据.R

以下代码主要分为两个部分：

1. 如果需要多个物种的分类、分布数据以及图片信息，可以直接使用循环模式；
2. 如果只需要某个物种以上三类信息、某些物种的某一类信息，则可以使用非循环模式。



```
1 # 不管是只要序列还是只要其他信息，都先把这些信息全部问询下来，然后再挑选
2 # 反正速度都很慢
3 # 不管是一个物种还是多个物种都把它放到 species_list.xlsx 内用于问询
4
5 rm(list = ls()) #清楚全局环境变量
6
7 # 载入相关R包
8
9 library(bold)
10 library(openxlsx)
11 library(tidyr)
12 library(beepr)
13
14 # 数据导入
15 species <- as.vector(unlist(read.xlsx('species_list.xlsx', colNames = F)))
16
17 # 循环模式-----
18 da_GenBank_ID <- data.frame() #新建一个空数据框，用于分别存放循环中产生的数据
19 da_Taxon <- data.frame() #存放分类信息
20 da_Location <- data.frame() #存放分布数据
21 da_Image_urls <- data.frame() #存放图片的url
22
23 for (i in 1:length(species)) {
24   # 循环模式
25   # 种在库内没有记录，则a为NA
26   spec(taxon = i) # bold_specimens返回的是一个data.frame
27 }
28
29 # 非循环模式
```

```
1 # 不管是只要序列还是只要其他信息，都先把这些信息全部问询下来，然后再挑选
2 # 反正速度都很慢
3 # 不管是一个物种还是多个物种都把它放到 species_list.xlsx 内用于问询
4
5 rm(list = ls()) #清楚全局环境变量
6
7 # 载入相关R包
8
9 library(bold)
10 library(openxlsx)
11 library(tidyr)
12 library(beepr)
13
14 # 数据导入
15 species <- as.vector(unlist(read.xlsx('species_list.xlsx', colNames = F)))
16
17 # 循环模式-----
18 -----
19 da_GenBank_ID <- data.frame() #新建一个空数据框，用于分别存放循环中产生的数据
```

```

19 da_Taxon <- data.frame()      #存放分类信息
20 da_Location <- data.frame()  #存放分布数据
21 da_Image_urls <- data.frame() #存放图片的url
22
23 for(i in species){
24   # 如果询问的物种在库内没有记录，则a为NA
25   a <- bold_seqspect(taxon = i) # bold_specimens返回的是一个data.frame
26   # 一般比较关心的是这个物种的GenBank编号、分类信息、坐标信息、图片网址信息。
27   if(is.na(a)){next}else{
28     # GenBank编号
29     GenBank_ID <- subset(a,institution_storing == 'Mined from GenBank,
NCBI',
30                           select =
c(species_name,sampleid,institution_storing))
31     da_GenBank_ID <- rbind(da_GenBank_ID,GenBank_ID)
32     # 分类信息
33     Taxon <- unique(a[c(16,18,20,22)]) #16,18,20,22对应的列是科,亚科,属,种,后面
还有亚种,看需要
34     da_Taxon <- rbind(da_Taxon,Taxon)
35     # 坐标信息
36     Location <- unique(a[c(22,47,48)]) # 47,48是纬度和经度，去重可能会有一行是空
值
37     good <- complete.cases(Location) #找到非空值
38     Location <- Location[good, ] #提取非空值所有行，就是所有非重复坐标
39     da_Location <- rbind(da_Location,Location)
40     # 图片信息
41     Image_urls <- subset(a,image_urls != "",select =
c(species_name,image_urls))
42     da_Image_urls <- rbind(da_Image_urls,Image_urls)
43     print(i) # 看看运行到第几个了（对于异常终止比较有用）
44   }
45 }
46 beep::beep(8) #因为BOLD比较慢（相比于GBIF），这边设置一个程序运行完的提示音
47
48 # 接下来就是保存你需要的数据内容
49 write.xlsx(da_GenBank_ID,'da_GenBank_ID.xlsx')
50 write.xlsx(da_Taxon,'da_Taxon.xlsx')
51 write.xlsx(da_Location,'da_Location.xlsx')
52 write.xlsx(da_Image_urls,'da_Image_urls.xlsx')
53
54 # 非循环模式-----
55
56 # 如果询问的物种在库内没有记录，则a为NA
57 a <- bold_seqspect(taxon = species) # bold_specimens返回的是一个data.frame
58 beep::beep(8) #因为BOLD比较慢（相比于GBIF），这边设置一个程序运行完的提示音
59
60 # 一般比较关心的是这个物种的GenBank编号、分类信息、坐标信息、图片网址信息。需要哪个就运行
哪个。
61
62 # GenBank编号
63 GenBank_ID <- subset(a,institution_storing == 'Mined from GenBank, NCBI',
64                       select = c(species_name,sampleid,institution_storing))
65
66 # 分类信息
67 Taxon <- unique(a[c(16,18,20,22)]) #16,18,20,22对应的列是科,亚科,属,种,后面还有
亚种,看需要
68
69 # 坐标信息

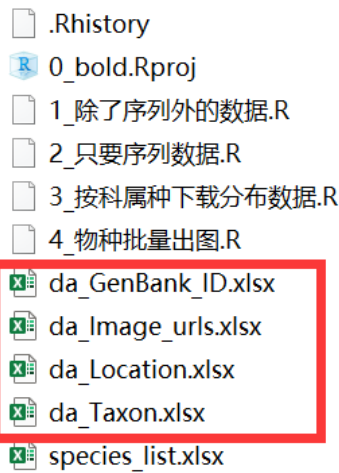
```

```

70 Location <- unique(a[c(22,47,48)]) # 47,48是纬度和经度，去重可能会有一行是空值
71 good <- complete.cases(Location) #找到非空值
72 Location <- Location[good, ] #提取非空值所有行，就是所有非重复坐标
73
74 # 图片信息
75 Image_urls <- subset(a,image_urls != "",select =
  c(species_name,image_urls))
76
77 # 接下来就是保存你需要的数据内容
78 write.xlsx(da_GenBank_ID, 'da_GenBank_ID.xlsx')
79 write.xlsx(da_Taxon, 'da_Taxon.xlsx')
80 write.xlsx(da_Location, 'da_Location.xlsx')
81 write.xlsx(da_Image_urls, 'da_Image_urls.xlsx')

```

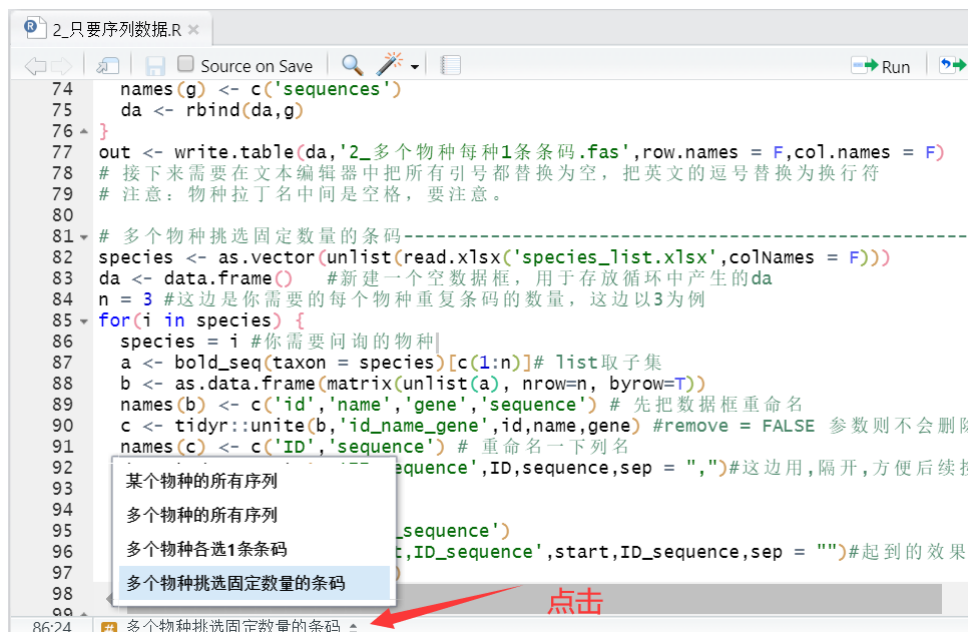
结果文件：



2_只要序列数据.R

以下代码主要分为四个部分：

- 某个物种的所有序列
- 多个物种的所有序列
- 多个物种各选1条条码
- 多个物种挑选固定数量的条码



```

1  # 只获取序列数据
2
3  rm(list = ls())
4
5  library(bold)
6  library(openxlsx)
7  library(tidyr)
8
9
10 # 某个物种的所有序列-----
11 -----
12 species = 'Papilio machaon' #你需要询问的物种,这边以金凤蝶为例
13 a <- bold_seq(taxon = species) # bold_seq返回的是一个包含4个list(id name gene
sequence)的list
14 length(a) # a包含的list的个数,下面一行命令会用到
15 b <- as.data.frame(matrix(unlist(a), nrow=length(a), byrow=T)) #nrow是根据a
中包含的list的个数定的
16 # 1和3列是重复的,把第一列删除
17 b <- b[,-1]
18 # 把每行合并起来,新生成一列放在b后面,方便后续的使用
19 names(b) <- c('name','gene','sequence') # 先把数据框重命名
20 c <- tidyr::unite(b,'name_gene',name,gene) #remove = FALSE 参数则不会删除原来
的数据列
21 names(c) <- c('ID','sequence') # 重命名一下列名
22 d <- tidyr::unite(c,'ID_sequence',ID,sequence,sep = ",")#这边用,隔开,方便后续
按照,替换为换行符做成.fasta格式
23 e <- data.frame(matrix(rep('>',length(a)), nrow=length(a), byrow=T))
24 f <- cbind(e,d)
25 names(f) <- c('start','ID_sequence')
26 g <- tidyr::unite(f,'start_ID_sequence',start,ID_sequence,sep = "")#起到的
效果就是在d的每一行开头添加一个>
27 names(g) <- c('sequences')
28 filename <- paste0('2_', '某个物种的所有序列_', species, ".fas")
29 out <- write.table(g,filename,row.names = F,col.names = F)
30
31 # 接下来需要在文本编辑器中把所有引号都替换为空,把英文的逗号替换为换行符
32 # 注意:物种拉丁名中间是空格,要注意。
33
34 # 多个物种的所有序列-----
35 -----
36 rm(list = ls())
37 species <- as.vector(unlist(read.xlsx('species_list.xlsx',colNames = F)))
38 da <- data.frame() #新建一个空数据框,用于存放循环中产生的da
39 for(i in species) {
40   species = i #你需要询问的物种
41   a <- bold_seq(taxon = species) # bold_seq返回的是一个包含4个list(id name
gene sequence)的list
42   length(a) # a包含的list的个数,下面一行命令会用到
43   if(length(a)==0){next}else{ #如果需要询问的物种没有序列,则length(a)=0,则跳过
该物种
44     b <- as.data.frame(matrix(unlist(a), nrow=length(a), byrow=T)) #nrow是
根据a中包含的list的个数定的
45     # 1和3列是重复的,把第一列删除

```

```

46     b <- b[,-1]
47     # 把每行合并起来，新生成一列放在b后面，方便后续的使用
48     names(b) <- c('name', 'gene', 'sequence') # 先把数据框重命名
49     c <- tidyr::unite(b, 'name_gene', name, gene) #remove = FALSE 参数则不会删除原来的数据列
50     names(c) <- c('ID', 'sequence') # 重命名一下列名
51     d <- tidyr::unite(c, 'ID_sequence', ID, sequence, sep = ",")#这边用，隔开，方便后续按照，替换为换行符做成.fasta格式
52     e <- data.frame(matrix(rep('>', length(a)), nrow=length(a), byrow=T))
53     f <- cbind(e, d)
54     names(f) <- c('start', 'ID_sequence')
55     g <- tidyr::unite(f, 'start_ID_sequence', start, ID_sequence, sep = "")#起到的效果就是在d的每一行开头添加一个>
56     names(g) <- c('sequences')
57     da <- rbind(da, g)
58   }
59 }
60 out <- write.table(da, '2_多个物种所有条码.fas', row.names = F, col.names = F)
61 # 接下来需要在文本编辑器中把所有引号都替换为空(就删除)，把英文的逗号替换为换行符
62 # 注意：物种拉丁名中间是空格。
63 # 如果需要特定属的，只需要把物种名换成属名就行，那每一个小的list就是一个物种。
64
65 # 多个物种各选1条条码-----
66 -----
67 # 某个物种的第一条序列（也就是选一条序列来代表这个物种），当然输入肯定不只是一个物种
68 # 那就在上面代码的基础上
69 rm(list = ls())
70 species <- as.vector(unlist(read.xlsx('species_list.xlsx', colNames = F)))
71 da <- data.frame() #新建一个空数据框，用于存放循环中产生的da
72 for(i in species) {
73   species = i #你需要询问的物种
74   s <- bold_seq(taxon = species)
75   if(length(s)==0){next}else{ #如果需要询问的物种没有序列，则length(s)=0，则跳过该物种
76     a <- bold_seq(taxon = species)[[1]] # bold_seq返回的是一个包含4个list(id name gene sequence)的list
77     b <- as.data.frame(matrix(unlist(a), nrow=1, byrow=T))
78     b <- b[,-1]
79     # 把每行合并起来，新生成一列放在b后面，方便后续的使用
80     names(b) <- c('name', 'gene', 'sequence') # 先把数据框重命名
81     c <- tidyr::unite(b, 'name_gene', name, gene) #remove = FALSE 参数则不会删除原来的数据列
82     names(c) <- c('ID', 'sequence') # 重命名一下列名
83     d <- tidyr::unite(c, 'ID_sequence', ID, sequence, sep = ",")#这边用，隔开，方便后续按照，替换为换行符做成.fasta格式
84     e <- data.frame(matrix(rep('>', length(a)), nrow=length(a), byrow=T))
85     f <- cbind(e, d)
86     names(f) <- c('start', 'ID_sequence')
87     g <- tidyr::unite(f, 'start_ID_sequence', start, ID_sequence, sep = "")#起到的效果就是在d的每一行开头添加一个>
88     names(g) <- c('sequences')
89     da <- rbind(da, g)
90   }
91 }
92 da <- unique(da)
93 out <- write.table(da, '2_多个物种每种1条条码.fas', row.names = F, col.names = F)

```

```

94 # 接下来需要在文本编辑器中把所有引号都替换为空，把英文的逗号替换为换行符
95 # 注意：物种拉丁名中间是空格，要注意。
96
97 # 多个物种挑选固定数量的条码-----
-----
98 rm(list = ls())
99 species <- as.vector(unlist(read.xlsx('species_list.xlsx', colNames = F)))
100 da <- data.frame() #新建一个空数据框，用于存放循环中产生的da
101 n = 3 #这边是你需要的每个物种重复条码的数量，这边以3为例
102 for(i in species) {
103   species = i #你需要询问的物种
104   s <- bold_seq(taxon = species) # bold_seq返回的是一个包含4个list(id name
gene sequence)的list
105   length(s) # a包含的list的个数，下面一行命令会用到
106   if(length(s)==0){next #如果需要询问的物种无条码，则跳过
107   }else if(length(s) < n){ #如果需要询问的物种一共的条码数都不到n，则全取
108   a <- bold_seq(taxon = species) # bold_seq返回的是一个包含4个list(id name
gene sequence)的list
109   length(a) # a包含的list的个数，下面一行命令会用到
110   b <- as.data.frame(matrix(unlist(a), nrow=length(a), byrow=T)) #nrow是
根据a中包含的list的个数定的
111   # 1和3列是重复的，把第一列删除
112   b <- b[,-1]
113   # 把每行合并起来，新生成一列放在b后面，方便后续的使用
114   names(b) <- c('name', 'gene', 'sequence') # 先把数据框重命名
115   c <- tidyr::unite(b, 'name_gene', name, gene) #remove = FALSE 参数则不会删
除原来的数据列
116   names(c) <- c('ID', 'sequence') # 重命名一下列名
117   d <- tidyr::unite(c, 'ID_sequence', ID, sequence, sep = ",")#这边用，隔开，方
便后续按照，替换为换行符做成.fasta格式
118   e <- data.frame(matrix(rep('>', length(a)), nrow=length(a), byrow=T))
119   f <- cbind(e, d)
120   names(f) <- c('start', 'ID_sequence')
121   g <- tidyr::unite(f, 'start_ID_sequence', start, ID_sequence, sep = "")#起
到的效果就是在d的每一行开头添加一个>
122   names(g) <- c('sequences')
123   da <- rbind(da, g)
124   }else{
125   a <- bold_seq(taxon = species)[c(1:n)]# list取子集
126   b <- as.data.frame(matrix(unlist(a), nrow=n, byrow=T))
127   # 1和3列是重复的，把第一列删除
128   b <- b[,-1]
129   # 把每行合并起来，新生成一列放在b后面，方便后续的使用
130   names(b) <- c('name', 'gene', 'sequence') # 先把数据框重命名
131   c <- tidyr::unite(b, 'name_gene', name, gene) #remove = FALSE 参数则不会删
除原来的数据列
132   names(c) <- c('ID', 'sequence') # 重命名一下列名
133   d <- tidyr::unite(c, 'ID_sequence', ID, sequence, sep = ",")#这边用，隔开，方
便后续按照，替换为换行符做成.fasta格式
134   e <- data.frame(matrix(rep('>', length(a)), nrow=length(a), byrow=T))
135   f <- cbind(e, d)
136   names(f) <- c('start', 'ID_sequence')
137   g <- tidyr::unite(f, 'start_ID_sequence', start, ID_sequence, sep = "")#起
到的效果就是在d的每一行开头添加一个>
138   names(g) <- c('sequences')
139   da <- rbind(da, g)
140   }
141 }

```



```
142 out <- write.table(da,'2_多个物种每种固定数量条码.fas',row.names = F,col.names
143 = F)
```

结果文件:

1. 2_某个物种的所有序列_Papilio machaon.fas

```
2_某个物种的所有序列_Papilio machaon.fas
1 ">Papilio.machaon.ABOLD047-16,AACATTATATTTTATTTTGGTATTTGAGCAAGTATATTAG
2 ">Papilio.machaon.CNCBF575-14,AACATTATATTTTATTTTGGTATTTGAGCAAGTATATTAG
3 ">Papilio.machaon.dodi_GBGL23550-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
4 ">Papilio.machaon.dodi_GBGL23551-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
5 ">Papilio.machaon.dodi_GBGL23552-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
6 ">Papilio.machaon.dodi_GBGL23553-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
7 ">Papilio.machaon.dodi_GBGL23554-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
8 ">Papilio.machaon.dodi_GBGL23555-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
9 ">Papilio.machaon.dodi_GBGL23556-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
10 ">Papilio.machaon.dodi_GBGL23557-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAG
```

2. 2_多个物种所有条码.fas

```
2_多个物种所有条码.fas
1 ">Orthopygia.glaucinalis_LTOLB153-08,AACTCTTATTTTATTTTGGGATTTGATCTGGGA
2 ">Aglossa.caprealis_ANICC567-10,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGTT
3 ">Aglossa.caprealis_ANICC569-10,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGTT
4 ">Aglossa.caprealis_ANICQ1777-11,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGT
5 ">Aglossa.caprealis_ANICQ1778-11,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGT
6 ">Aglossa.caprealis_ANICQ1779-11,-----
7 ">Aglossa.caprealis_GBMNB46900-20,AACTTTATATTTTATTTTCGGAATTTGATCAGGTATAG
8 ">Aglossa.caprealis_LNAUU2429-15,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGT
9 ">Aglossa.caprealis_LOCBF3722-14,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGTT
10 ">Aglossa.caprealis_NLLEA1257-14,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGT
11 ">Aglossa.caprealis_NLLEA1332-14,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGT
12 ">Papilio.machaon.ABOLD047-16,AACATTATATTTTATTTTGGTATTTGAGCAAGTATATTAG
```

3. 2_多个物种每种1条条码.fas

```
2_多个物种每种1条条码.fas
1 ">Orthopygia.glaucinalis_LTOLB153-08,AACTCTTATTTTATTTTGGGATTTGATCTGGG
2 ">Aglossa.caprealis_ANICC567-10,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGT
3 ">Papilio.machaon.ABOLD047-16,AACATTATATTTTATTTTGGTATTTGAGCAAGTATATTAG
```

4. 2_多个物种每种固定数量条码.fas (这边选了3条, 不足3条则全要)

```
2_多个物种每种固定数量条码.fas
1 ">Orthopygia.glaucinalis_LTOLB153-08,AACTCTTATTTTATTTTGGGATTTGATCTGGGATA
2 ">Aglossa.caprealis_ANICC567-10,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGTTGG
3 ">Aglossa.caprealis_ANICC569-10,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGTTGG
4 ">Aglossa.caprealis_ANICQ1777-11,AACTTTATATTTTATTTTGGGAATTTGATCAGGTATAGTTG
5 ">Papilio.machaon.ABOLD047-16,AACATTATATTTTATTTTGGTATTTGAGCAAGTATATTAGGAA
6 ">Papilio.machaon.CNCBF575-14,AACATTATATTTTATTTTGGTATTTGAGCAAGTATATTAGGAA
7 ">Papilio.machaon.dodi_GBGL23550-19,TGGATTTGGAATAATTTCTCATATTATTTCCCAAGAAA
8
```

2. 获取分布数据绘制分布图

有时候想要获取特定科或者属的分布数据用于绘图, 这边把数据下载的代码和绘图的代码分开, 便于后期调用。

3_按科属种下载分布数据.R

但是由于BOLD网页本身在数据输入和存储上可能会有些问题, R在利用API解译的时候可能会出现串行的问题, 如果遇到类似问题, 建议去网页版直接下载有问题的数据。

涉及高界元数据的下载时, 会比较慢, 对网络环境的要求比较高, 请耐性等待, 出现下载问题时, 可重复尝试2-3次, 结合beep的提示音, 会方便一些。


```

2  # 当然前面查询基本信息和序列的时候也可以按照科属水平去查找，可自行变通
3
4  rm(list = ls())
5
6  library(bold)
7  library(openxlsx)
8  library(tidyr)
9  library(beepr)
10
11
12  # 科-----
13  a <- bold_specimens(taxon='Nolidae') #瘤蛾科测试
14  a <- bold_specimens(taxon='Nolidae',geo='China') #限制样本采集地为中国，不限制默
    认为全世界
15  beepr::beep(sound = "mario")#因为BOLD比较慢（相比于GBIF），这边设置一个程序运行完的
    提示音
16  Location <- unique(a[c(22,47,48)]) # 47,48是纬度和经度，去重可能会有一行是空值
17  good <- complete.cases(Location) #找到非空值,这里都是逻辑值
18  Location_family <- Location[good, ] #提取非空值所有行，就是所有非重复坐标
19  # 保存数据
20  #如果物种名一栏是空的，表明只定到了属，没定到种，可根据分析目的合理筛选
21  write.xlsx(Location_family,'Location_family.xlsx')
22  # 作图
23  # 请使用【物种批量出图.R】脚本作图
24
25  # 属-----
26  b <- bold_specimens(taxon='Meganola') #瘤蛾科 Meganola属 测试
27  b <- bold_specimens(taxon='Meganola',geo='China')
28  beepr::beep(sound = "mario")
29  Location <- unique(b[c(22,47,48)]) # 47,48是纬度和经度，去重可能会有一行是空值
30  good <- complete.cases(Location) #找到非空值,这里都是逻辑值
31  Location_genus <- Location[good, ] #提取非空值所有行，就是所有非重复坐标
32  # 保存数据
33  #如果属名一栏是空的，表明只定到了科，没定到属，可根据分析目的合理筛选
34  write.xlsx(Location_genus,'Location_genus.xlsx')
35  # 作图
36  # 请使用【物种批量出图.R】脚本作图
37
38  # 种-----
39  # 如果是一个物种可以向上面科属类似，把“taxon=”后面的内容换成某个物种就行
40  # 但是一般我们需要批量查询多个物种，可以采用下面的循环
41  # species_list.xlsx 存入需要询问的物种，格式参考实例文件
42  species <- as.vector(unlist(read.xlsx('species_list.xlsx',colNames = F)))
43
44  Location_sp <- data.frame()#新建一个空数据框，用于分别存放循环中产生的坐标数据
45
46  for(i in species){
47    # 如果询问的物种在库内没有记录，循环结束之后会有报错，不用在意
48    a <- bold_specimens(taxon = i) # bold_specimens返回的是一个data.frame
49    if(is.na(a)){next}else{
50      # 坐标信息
51      Location <- unique(a[c(22,47,48)]) # 47,48是纬度和经度，去重可能会有一行是空
    值
52      good <- complete.cases(Location) #找到非空值
53      Location <- Location[good, ] #提取非空值所有行，就是所有非重复坐标
54      Location_sp <- rbind(Location_sp,Location)
55
56      print(i) # 看看运行到第几个了（对于异常终止比较有用）

```

```

57 }
58 }
59 beep::beep(sound = "mario")
60
61 # 保存数据
62 write.xlsx(Location_sp, 'Location_sp.xlsx')
63

```

结果文件:

1. Location_family.xlsx

species_name	lat	lon
Westermannia superba	21.616	101.579
Meganola scriptoides	21.591	101.548
Giaura robusta	21.591	101.549
Hylophilodes orientalis	21.62	101.572
Hylophilodes orientalis	21.621	101.571
Risoba prominens	21.591	101.547
Ptyonota formosa	21.621	101.574
Risoba prominens	21.62	101.574
Chloroplaga nygmia	21.613	101.576
Chloroplaga nygmia	21.613	101.58
物种名不确定 可能本身只鉴定 到了科或者属	24.278	101.265
	24.284	101.256
	24.287	101.251
	27.167	100.233
	24.286	101.251
	24.284	101.254

2. Location_genus.xlsx

species_name	lat	lon
只鉴定到属，没能到种	29.183	118.5
Meganola scriptoides	21.591	101.548

3. Location_sp.xlsx

	A	B	C
1	species_name	lat	lon
2	Orthopygia glaucinalis	25	121.31
3	Aglossa caprealis	-35.218	138.541
4	Aglossa caprealis	-34.191	150.982
5	Aglossa caprealis	-35.263	149.122
6	Aglossa caprealis	-35.22	138.541
7	Aglossa caprealis	-28.73	153.467
8	Aglossa caprealis	-35.926	149.703
9	Aglossa caprealis	52.16	4.483
10	Papilio machaon	47.833	16.167
11	Papilio machaon	49.6078	-119.677
12	Papilio machaon	40.667	16.614
13	Papilio machaon	41.955	14.966
14	Papilio machaon	46.208	13.526
15	Papilio machaon	40.817	8.934
16	Papilio machaon	39.256	9.367
17	Papilio machaon	42.133	14.657

4_物种批量出图.R

BOLD提供的坐标数据相对比较少，此处只是简单看看分布，如果需要做进一步精细的处理，建议去GBIF或其他更专业的网站下载数据用于绘图。

```

1 # 不同物种批量打点出地图
2
3 rm(list = ls()) #清除环境变量
4
5 library(openxlsx)
6 library(ggplot2)
7 library(tidyverse)
8 library(sf)
9 library(maptools)
10 library(ggspatial)
11 library(cowplot)
12 library(purrr)
13
14 # 世界循环作图:定义函数+purrr-----
15 input <- read.xlsx('去重后-4.xlsx')
16
17 world <- map_data("world")
18 # world <- world[world$region != "Antarctica",] # 剔除南极洲
19
20 # 定义作图函数(逐个保存)-----
21 plot_world <- function(x) {
22   df <- subset(input, species_name == x)
23   ggplot(world)+
24
25     geom_polygon(aes(x=long,y=lat,group=group),fill='white',colour='black')+
26     coord_quickmap()+ #为地图设置合适的纵横比
27     geom_point(data = df,aes(lon,lat),
28               shape=16,colour='red',size=1)+
29     ggtitle(paste('The worldwide distribution of',x))+
30     theme(plot.title = element_text(hjust = .5),panel.grid =
31     element_blank())
32   filename <- paste('world_',x,'.pdf')
33   ggsave(filename,width = 7, height = 5)
34 }
35 # 利用purrr的map函数循环出图, 利用cowplot::plot_grid()函数排列图
36 sp <- as.vector(unlist(unique(input[1])))
37 purrr::map(sp, plot_world) #批量绘图,默认的是7*7的尺寸

```

结果文件:

