生態模擬： 以 C 語言為例

Class 10 2018/05/31

Input from & Output to files; Monte-Carlo Simulations

10.1 Review of rules in printf and scanf
10.2 Output to file
10.3 Input from file
10.4 Input from csv file (vector and matrix)
10.5 Input from files: specification from command line
10.6 Random number
10.7 Assignment (species richness vs functional richness)

Takeshi Miki

三木　健（海洋研究所）

10.1 Review of rules in printf and scanf

Basic rules for printf

轉換字元

```
printf(“%d is a integer.\n”,  123 );
```

```
printf(“%f is a float.\n”,  10.5 );
```

```
printf(“%.4lf is a double.\n”,  10.5 );
```

```
printf(“The value of the variable num is %d.\n”,  num );
```

10.1 Review of rules in printf and scanf

Input values from keyboard into variables

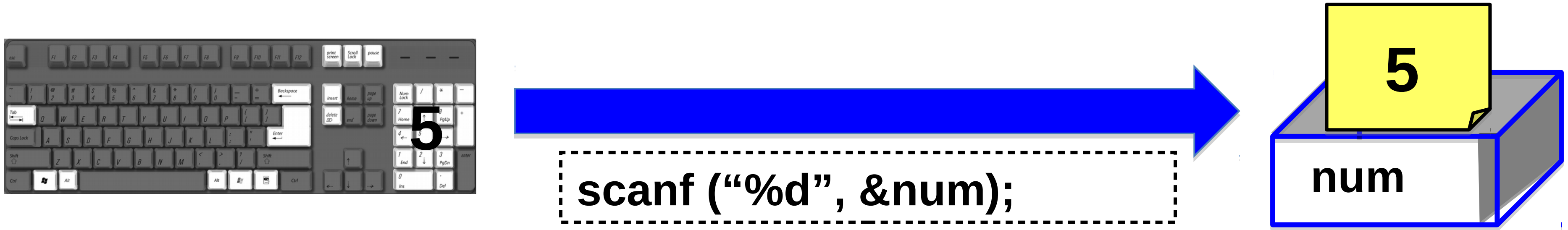
構文 (Syntax):

```
#include <stdio.h>

int main (void)
{
            宣告 ;

    scanf (“ 轉換字元” , & 變數 );
    ....
}
```

You can assign the value input from the keyboard to this variable.



```
scanf (“%d”, &num);
```

10.2 Output to file

Output data to a fileㅂ error management, .e.g “error.c”

```
void message_error(char error_text[])
/*Standard error handler*/
{
        printf("There are some errors...\n");
        printf("%s\n", error_text);
        printf("...now existing to system...\n");
        exit(1);
}
```

10.2 Output to file

Output data to a file

```
#include <stdio.h>
#include <stdlib.h>
#include “error.c”
#define NUM 5
int main (void)
{
        FILE *fp1;
        double *vector_test;
        int j;
        vector_test = d_vector(NUM);
        fp1 = fopen(“test1.txt”, “w”);
        if(fp1 == NULL) message_error(“File cannot be opened.\n”);

        for(j = 1; j <= NUM; j++) vector_test[j] = 10.0*j;
        for(j = 1; j <= NUM; j++) fprintf(fp1, “%d\t%lf\n”, j,  vector_test[j]);
        fclose(fp1);
        return 0;
}
```

Defining a pointer to file

Opening file “…” as writing mode “w”.

Error management

Closing the file

Outputting data to file

10.3 Input from file

Input data from a file

```
#include <stdio.h>
#include <stdlib.h>
#include “error.c”
#define NUM 5
int main (void)
{
        FILE *fp1;
        double *vector_test;
        int j;
        vector_test = d_vector(NUM);
        fp1 = fopen(“vector.dat”, “r”);
        if(fp1 == NULL) message_error(“File cannot be opened.\n”);

        for(j = 1; j <= NUM; j++) fscanf(fp1, “%lf”,  &vector_test[j]);
        for(j = 1; j <= NUM; j++) printf(“%lf\n”, vector_test[j]);
        fclose(fp1);
        return 0;
}
```

Defining a pointer to file

Opening file “…” as reading mode “r”.

Error management

Inputting data from file

Closing the file

vector.dat

```
80
68
23
45
33
```

11.4 Input from csv file (vector and matrix)

See “reading_matrix.c.pdf”

```
#include <stdio.h>
#include <stdlib.h>
#include “../new_util/nrutil_new.c”
#define NUM 5
…
int main (void)
{
       FILE *fp1, *fp2;
       …
#ifdef VECTOR
      for (i = 1; i <= SIZE_V; i++) fscanf(fp1, “%d,”, &vector_test[i]);
      for (i = 1; i <= SIZE_V; i++) printf(“%d\n”, &vector_test[i]);
#endif

#ifdef MATRIX
…
#endif
        return 0;
}
```

vector2.dat

```
80,68,23,0,3
```

matrix.dat

```
80;68;23
10;20;30
3;2;1
3;2;5
0;2;100
```

Conditional Compiling

```
bash-miki> gcc –DVECTOR ***.c –o vector.out
bash-miki> gcc –DMATRIX ***.c –o matrix.out
```

10.5 Input from files: specification from command line

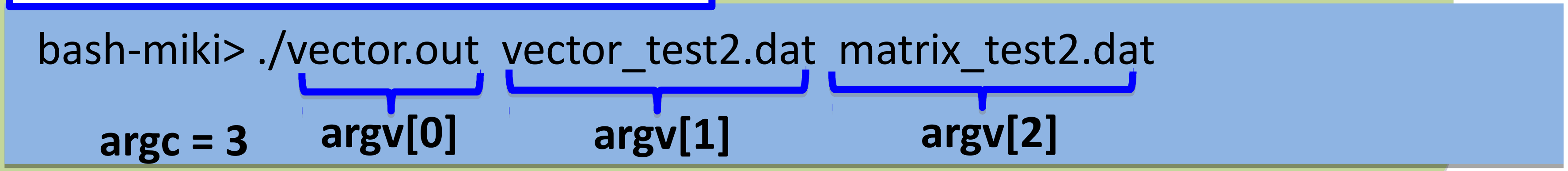
Modify “reading_matrix.c.pdf”

```
#include <stdio.h>
#include <stdlib.h>
#include “error.c”
…
int main(int argc, char *argv[])
{
        FILE *fp1, *fp2;
        …
        if(argc != 3) message_error(“Number of parameter is incompatible.”);
        fp1 = fopen( argv[1], “r”);
        fp2 = fopen( argv[2], “r”);

#ifdef VECTOR
…
#endif
…
return 0;
}
```

Error management

When you execute program



```
bash-miki> ./vector.out  vector_test2.dat  matrix_test2.dat
```

10.6 (Pseudo-) Random number (take time for practice)

SIMD-oriented Fast Mersenne Twister (SFMT) [BSD License]

```
#include <stdio.h>
#include <stdlib.h>
#include “../XXX/dSFMT.c”
…
int main(int argc, char* argv[]) {
    int i, seed;
    double x;
    dsfmt_t dsfmt;

    seed = time(NULL);        //return time (seconds  from1970/01/01/00:00:00 GMT
   if(seed == 0) seed = 1;

    dsfmt_init_gen_rand(&dsfmt, seed);
    x = dsfmt_genrand_close_open(&dsfmt);
…
}
```

You should first unzip dSFMT-src-2.1.zip and copy the all contents within the folder to somewhere (XXX). But you just need t o include ‘dSFMT.c’ in your code.

The period: 2^{19937}-1 (2^{521}-1 to 2^{216091}-1)

※default in C library rand(): only 2^{31} !!

generate 1 random number

Steps necessary for initialization

Initialization

Generating a double number in [0, 1)

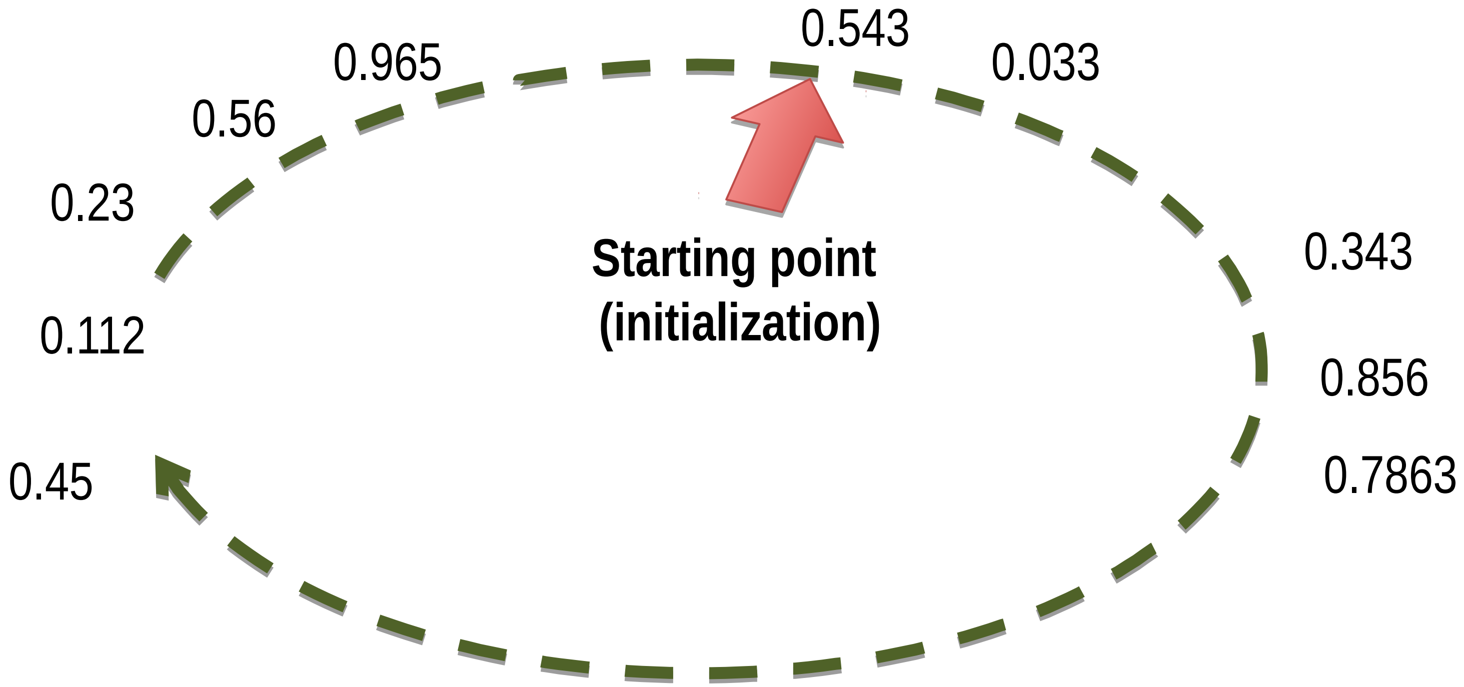
See “annotated.html” for detail manual

10.6 (Pseudo-)Random number (take time for practice)

For any pseudo-random number generator,

It generates the sequence of pseudo-random numbers from the defined '*periodic*' sequence.

The initialization process is necessary but only once to determine the starting point at the periodic sequence.



10.6 Random number

```
dasfmt_genrand_close_open(&dsfmt)
```

returns a pseudo random number [0, 1) with uniform distribution.

Question1:

How can you generate the random sequence of integer from 1 to N without overlap?

e.g. From 1 to 5, the sequence is (1, 3, 4, 2, 5)
e.g. From 1 to 5, non-overlapping three numbers (e.g. 1, 5, 3)

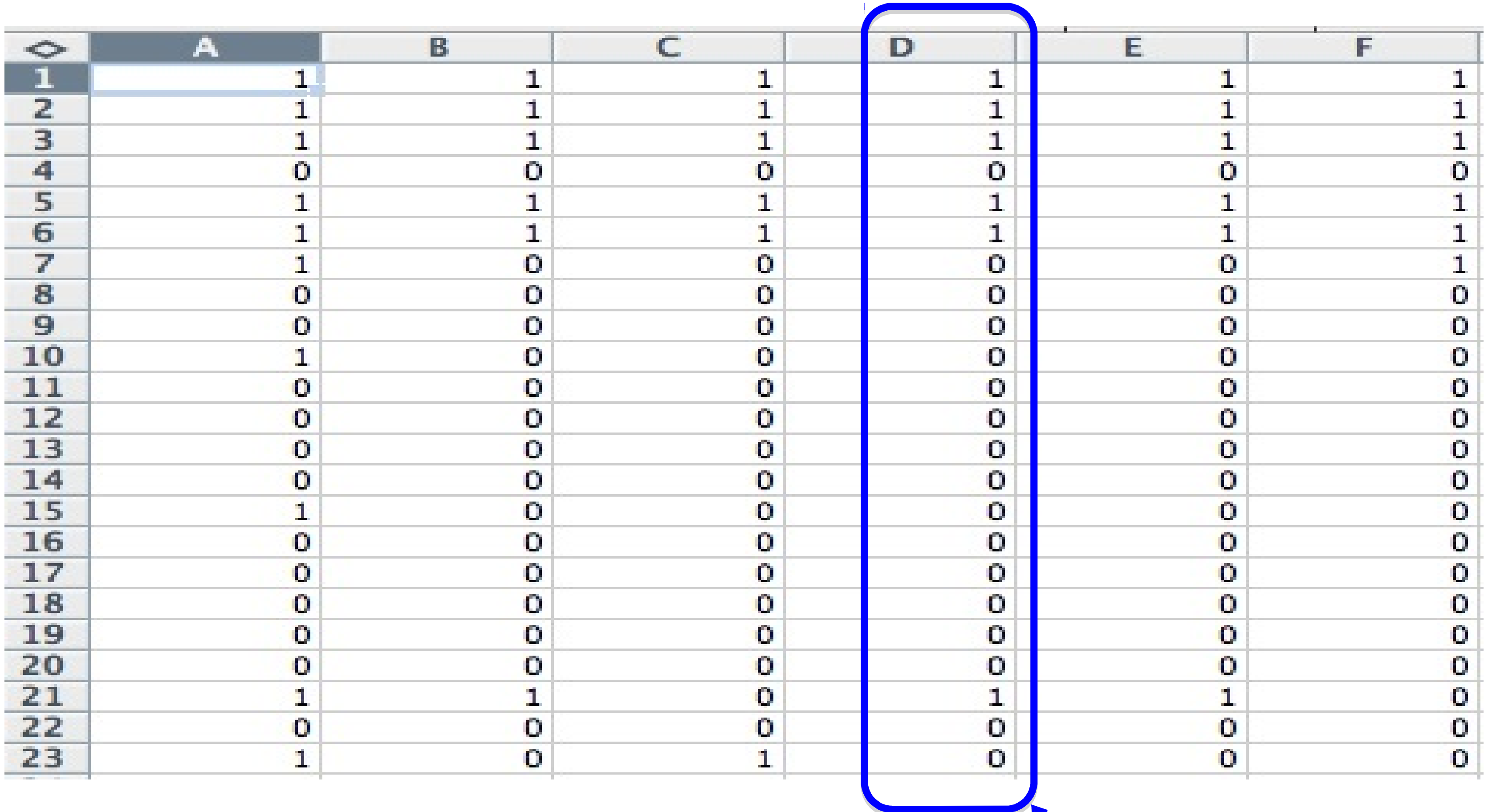
- First think by yourself and then, you can also google it.

10.7 Homework: species richness vs functional richness

STEP1: Load a matrix data “matrix_part.csv” into a program.

Species ID (# Species = 52)

Functional gene ID (# gene = 10,000)

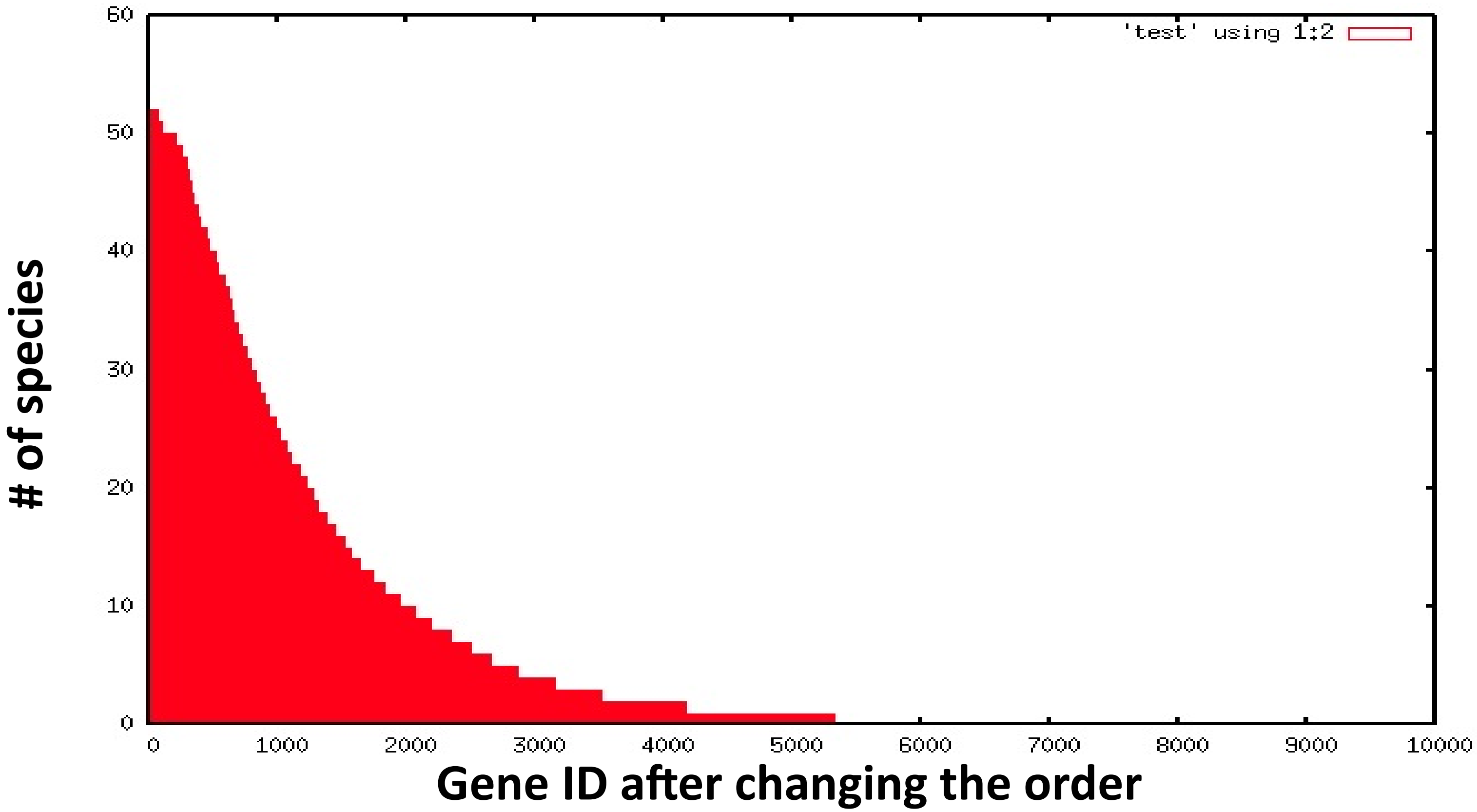


◇	A	B	C	D	E	F
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	0	0	0	0	0	0
5	1	1	1	1	1	1
6	1	1	1	1	1	1
7	1	0	0	0	0	1
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	1	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	1	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	1	1	0	1	1	0
22	0	0	0	0	0	0
23	1	0	1	0	0	0

List of presence(1)/absence(0) of each gene in sp. D

10.7 Homework: species richness vs functional richness

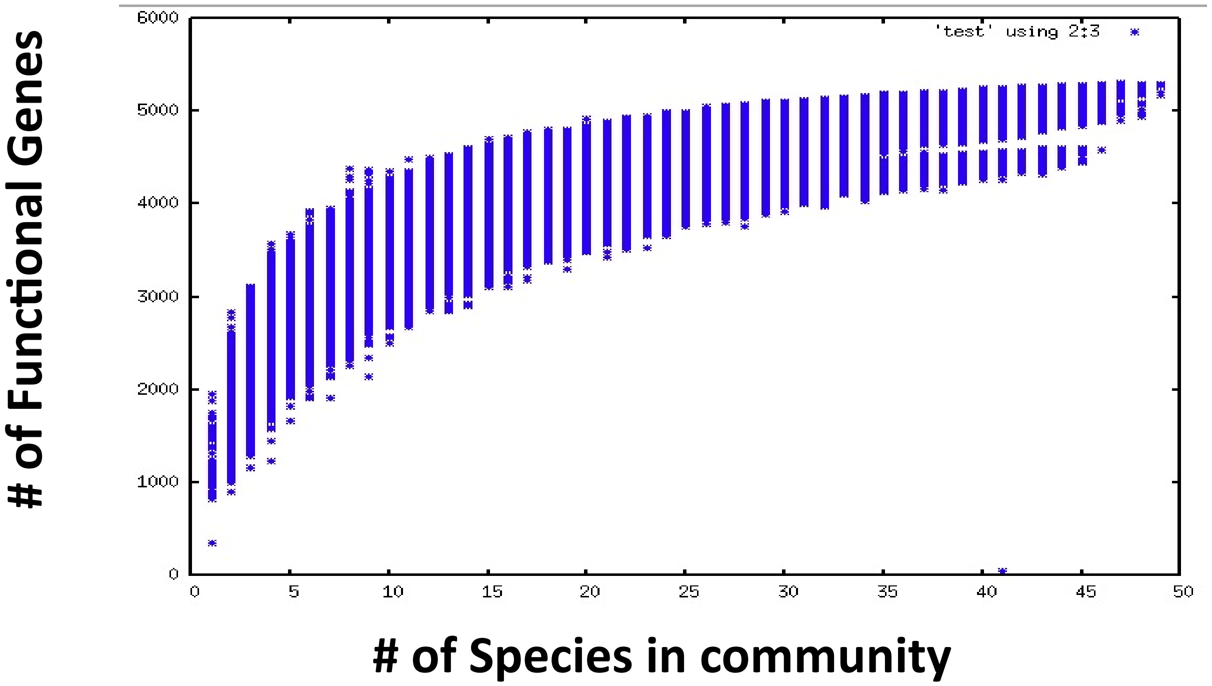
STEP2: For each gene j(j = 1,2,…,10000), **count** the number of species that have the gene j.
STEP3: Change the order of gene from 1st rank (with the largest number of species count) to 10000th rank, and **draw** the following graph.

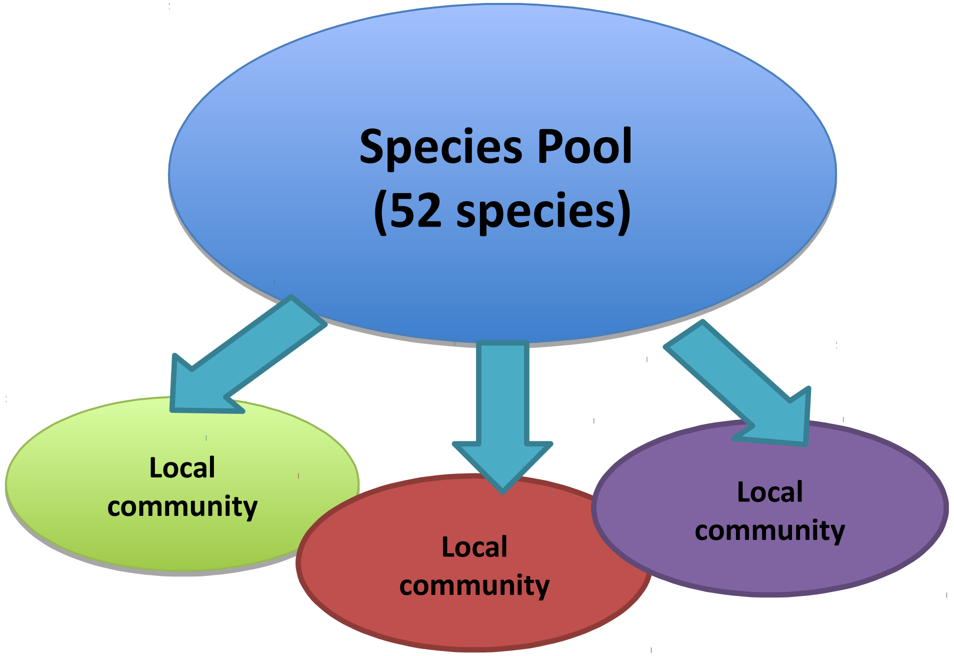


10.7 Homework: species richness vs functional richness

STEP4: Randomly choose k species (k = 1, 2, …, 52) without overlap as a member of local community from 52 species (acting as the species pool),and **count** the total number of genes that are found in the community. Of-course, **avoid** the double count the shared genes between species.

STEP5: Repeat this simulation for 100 times for each k (k = 2, 3, .., 50), i.e. **generate** 100 random communities for each k. For k = 1, 51, 52, you can count all combinations. Then, **draw** the diagram. You don't need to check if there is the identical communities within these 100 replicates.



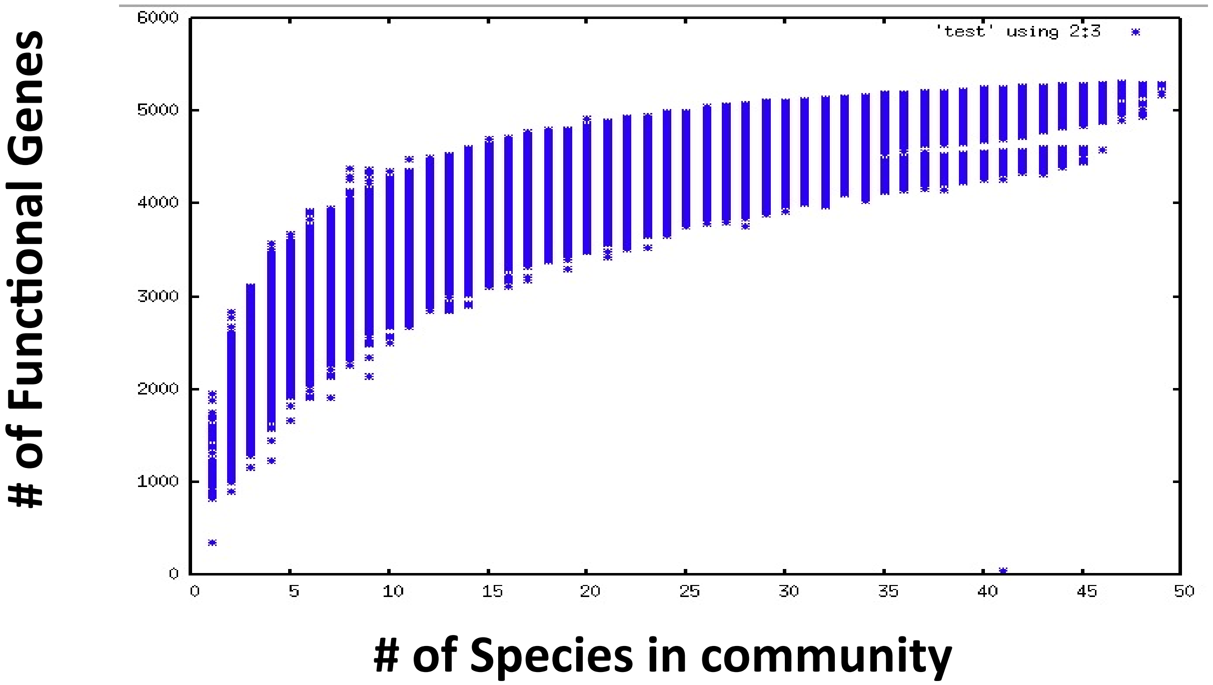

Species Pool (52 species)

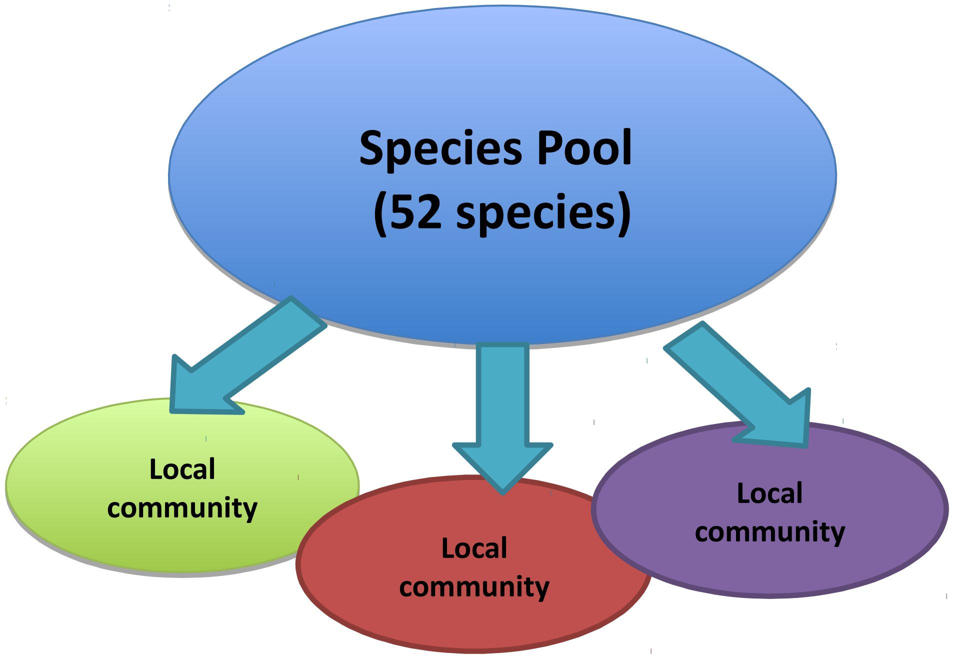
Local community

Local community

Local community

10.7 Homework: species richness vs functional richness





Species Pool (52 species)

Local community

Local community

Local community

This in fact is regarded as the simulation of random extinction processes of species from the largest community with 52 species.