

数理生物学演習

第5回 突然変異固定までの待ち時間

第5回：突然変異固定までの待ち時間

本日の目標

- ハーディ-ワインベルグ平衡の導出
- ライト-フィッシャー モデルの解析
- 擬似乱数

ハーディー-ワインベルグ モデル

世代を越えて遺伝子型頻度が維持されるケース

仮定

二倍体の有性生殖する生物を考える。
次のような性質を持つと仮定する。

- ・ ランダム交配する
- ・ 世代は重ならない
- ・ 突然変異は起こらない

この生物の十分大きな集団において
中立な対立遺伝子Aとaに注目する。

ある世代における遺伝子型AA, Aa, aa
それぞれの頻度は、前の世代の配偶子中のA
とaの頻度 p と q (ただし $p + q = 1$) を用い
て、

$$AA : Aa : aa = p^2 : 2pq : q^2$$

となる。

この遺伝子型頻度は世代を越えて維持さ
れ、ハーディー-ワインベルグ平衡と呼ばれる。

次世代の遺伝子型頻度を計算してみよう

→ 遺伝子型頻度が維持されているか確認してみよう

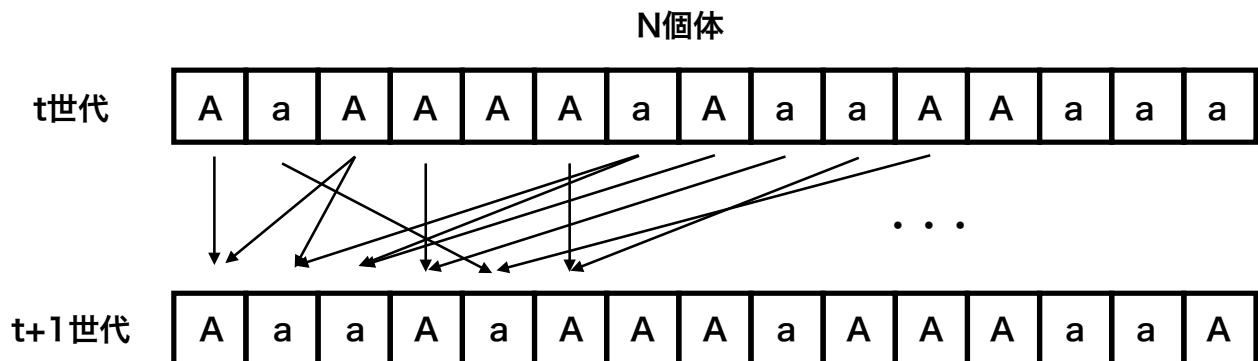
ライト-フィッシャー モデル

遺伝的浮動を考えてみる

半数体の生物N個体からなる集団について、ある中立な対立遺伝子A, aに注目する。
次のような性質を持つとする。

- ・ ランダム交配する
- ・ 世代は重ならない
- ・ (追加での) 突然変異は起こらない

もう少し詳しく知りたい人はHartl &
Clark, Principles of Population
Genetics 4th ed.の第3章を見てね



集団サイズはN個体で一定

親個体をランダムに選び、どちらの遺伝子を引き継ぐかはランダムに決まる

実際にプログラムを組んでみよう！

擬似乱数

- ある確率分布に従うランダムな数値の系列（乱数列）を生成したいが、“真に”ランダムな数値を得ることは難しい
- 決定論的なアルゴリズムによって本当の乱数列と（特定の目的上）区別がつかない数値（擬似乱数）列を生成し、これで代替することが一般的

- `srand(シード)`
乱数の種（シード）を設定する
- `rand()`
0以上RAND_MAX以下の範囲で整数型の疑似乱数を生成する
RAND_MAXは

注意：srand関数やrand関数を用いるときはstdlib.hをincludeする

```
5-1. 擬似乱数列の生成
#include <stdio.h>
#include <stdlib.h>

int main(void){
    int r,i;
    srand(1);

    for(i=0;i<100;i++){
        r=rand();
        printf("%d\n",r);
    }
    return 0;
}
```

乱数の種（シード）が同じなので毎回同じ結果が表示される

```
出力
16807
282475249
. . .
892053144
```

実際の解析では、もっと性質の良い（例えば、周期が長い）疑似乱数生成器を使うことが推奨されるが、本演習ではrand関数を用いる

擬似乱数の種（シード）の本演習における設定の仕方

本演習では（あまり良い方法ではないが）現在時刻をシードにする方法を用いる

- `time(NULL)`
グリニッチ標準時の1970年1月1日 00:00:00から現在までの経過時間（秒単位）を返す

注意1： `time`関数を用いるときは `time.h`をincludeする
注意2： 1秒以内に再度実行するとシードが同じになる

5-2. 現在時刻をシードにした擬似乱数生成

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void){
    int r,i, t;
    t = time(NULL);
    srand(t);

    for(i=0;i<100;i++){
        r=rand();
        printf("%d\n",r);
    }
    return 0;
}
```

出力
1548724357
1908466459
...
614321673

毎回異なる結果が表示される

特定の範囲内での乱数の生成

$M \leq r \leq N$ となる擬似乱数 r を生成させる場合は次式を用いる

$$r = \text{rand()} \% (N - M + 1) + M$$

M~Nの範囲に含まれる
整数の数

例：サイコロ ($M = 1, N = 6$)

- $N - M + 1 = 6$

`rand()`は0以上`RAND_MAX`以下の整数を返す

- $0 \leq \text{rand()} \% (N - M + 1) \leq 5$
- $1 \leq \text{rand()} \% (N - M + 1) + M \leq 6$

5-3. サイコロ

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void){
    int r,i,M,N;
    srand(time(NULL));
    M=1;
    N=6;

    for(i=0;i<100;i++){
        r=rand()%(N-M+1)+M;
        printf("%d\n",r);
    }
    return 0;
}
```

ループからの脱出：break文

- ・ループから強制的に抜け出したいときがある
- ・通常はforやwhileで条件判定をうまく設定すれば良いが、細かく継続判定を設定し分割して判定するよう処理したい場合などに用いる

- **break;**
一番内側のループを終了し、そのループの次の文へ処理が進む

ループから脱出する
(i>10でforループは実行されない)

for, whileのいずれでも利用できる

```
5-4. ループの中断
#include <stdio.h>

int main(void){
    int i;

    for(i=0;i<100;i++){
        printf("%d\n",i);
        if(i==10){
            break;
        }
    }
    return 0;
}
```

サイコロの目の総和が100を超えるまでの待ち時間

5-5. サイコロの目の総和が100を超えるまでの待ち時間

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

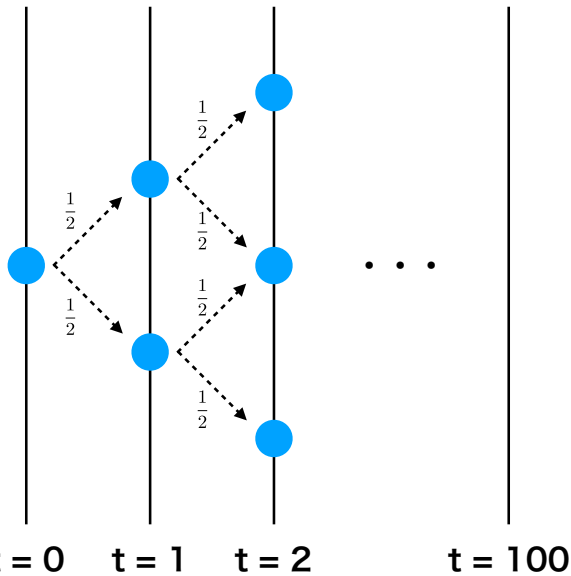
int main(void){
    int r,M,N,x,i;
    srand(time(NULL));
    M=1;
    N=6;
    x=0;
    x : サイコロの目の総和
```

```
for(i=0;i<100;i++){
    r=rand()%(N-M+1)+M;
    x=x+r;
    if(x>=100){
        printf("%d\n",i);
        break;
    }
}
return 0;
```

xが100以上ならば
ステップ数を出力してループを脱出

ランダムウォーク

以下のような1次元の単純ランダムウォークをシミュレーションしてみよう。



解釈例：

- コインをなげて表が出れば+1点，裏が出れば-1点を得るゲームの点数の推移
- 右と左にそれぞれ1/2の確率で移動する生物の移動軌跡

5-6. ランダムウォーク

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void){
    int r,x,i;
    srand(time(NULL));
    x=0;

    for(i=0;i<100;i++){
        r=rand()%2;
        if(r==0){
            x=x+1;
        }
        else{
            x=x-1;
        }

        printf("%d\n",i);
    }

    return 0;
}
```

ファイルへ出力するバージョンへ書き換えてみよう

マクロ

コンパイルの前処理で特定の文字列を置き換えることができる

- #define 文字列1 文字列2
文字列1を文字列2に置き換える

例えば，配列サイズを予め指定したい場合，変数では指定出来ない。



```
int main(void){
    int N;
    N = 50;
    int a[N],b[N],c[N];
    return 0;
}
```



```
#define N 50
int main(void){
    int a[N],b[N],c[N];
    return 0;
}
```

動的配列という方法があるが本演習では扱わない。興味ある人は調べてみて。

ライト-フィッシャー モデル

5-7. ライト-フィッシャー モデル

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 50

int main(void){
    int a[N],aa[N],i,t,r1,r2,r;
    srand(time(NULL));

    for(i=0;i<(N/2);i++){
        a[i]=0;
        a[i+N/2]=1;
    }

    for(i=0;i<N;i++){
        printf("%d ",a[i]);
    }
    printf("\n");
```

N個体の集団
個体番号は0~49

半分の個体が『0』
もう半分の個体が『1』を
持つとする

各個体の持つ
対立遺伝子を出力

```
for(t=0;t<100;t++){
    for(i=0;i<N;i++){
        r1=rand()%N;
        r2=rand()%N;
        r=rand()%2;
        if(r==0){
            aa[i]=a[r1];
        }
        if(r==1){
            aa[i]=a[r2];
        }
    }
    for(i=0;i<N;i++){
        a[i]=aa[i];
        printf("%d ",a[i]);
    }
    printf("\n");
}
return 0;
```

r1番目とr2番目
の個体が親とし
て選ばれる

どちらの親
から遺伝子
を引き継ぐ
か、確率1/2
でランダムに
決まる

各個体の持つ
対立遺伝子を出力

本日の課題

1. ソースコード5-5をループ（平均算出用）でネストして改良し、1回、5回、10回、100回、1000回の平均待ち時間を調べよ
2. 1について、手計算で平均待ち時間を求め、課題1の結果と比較、考察せよ
3. ランダムウォークを5系列重ねてプロットせよ
4. N個体の集団内に占める突然変異対立遺伝子を持つ個体の数をk、その頻度をp (=k/N) とする。このとき、集団に突然変異が固定する場合について、その100回分の平均待ち時間Tを求めよ

ハード 5. N=100, N=200の場合について、それぞれkを1~NまでN/10刻みで変化させ、突然変異の初期頻度pに対する平均待ち時間Tを10個ずつプロットしなさい。

ハード 6. 半数体生物に対して突然変異固定までの平均待ち時間Tの解析解が

$$T(p) = -\frac{1}{p} \{2N(1-p) \log_e(1-p)\}$$

で与えられるとき、このグラフをN=100, N=200について描き、同じグラフ上で上記のプロットと比較し、考察せよ。

7. 質問、意見、要望等をどうぞ。

課題をPDFファイルにまとめて、Google フォームにて提出すること

次回予告

第6回：疫学モデル

5月21日

復習推奨

- SIRモデル
- 基本再生産数
- 最小二乗法