

Notas de Aula da Disciplina Física Computacional

Alexandre Souto Martinez
Universidade de São Paulo - USP
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto - FFCLRP
Departamento de Física - DF

tel.: 0xy16 3315-3720
email: asmartinez@ffclrp.usp.br

14 de Maio de 2018

Resumo

bla

Palavras-Chave:

Conteúdo

1	Simulações de Monte Carlo de Tempo Discreto	2
1.1	Lançamentos de Uma Moeda ou Lançamento de Moedas . . .	3
1.1.1	Lançamento de uma Moeda Uma Vez	3
1.1.2	Lançamentos de uma Moeda Muitas Vezes	3
1.2	Difusão de um Gás	5
1.3	Decaimento Radioativo	7
1.4	Exercícios	7
1.5	Genética de Populações	12
1.5.1	Modelo de Wright-Fisher	13
1.5.2	Modelo de Moran	17

Capítulo 1

Simulações de Monte Carlo de Tempo Discreto

A aleatoriedade aparece devido o nosso conhecimento incompleto da realidade, da falta de informação que proíbe uma predição do futuro. A aleatoriedade aparece da complexidade, do fato que causas são diversas e que pequenas perturbações podem resultar em grandes efeitos. Há mais de um século, a Ciência abandonou a visão determinista de Laplace e tem amplamente aceitado a tarefa de decifrar a aleatoriedade e inventado ferramentas adequadas para a sua descrição. A surpresa é que, afinal de contas, a aleatoriedade tem muitas facetas e que existem em muitos níveis de incerteza, mas, acima de tudo, uma nova forma de previsibilidade aparece, que não é mais determinista mas estatística.

Jean-Philippe Bouchaud e Marc Potters, em “Theory of financial risk and derivative pricing: from statistical physics to risk management. Cambridge University Press, Cambridge 2003.

A seguir, apresentamos alguns exemplos de cálculos computacionais, onde se utiliza o gerador de números pseudo-aleatórios. Em geral, quando se utiliza um gerador de números pseudo-aleatórios, para realizar um cálculo computacional, chamamos esse algoritmo de *simulação computacional de Monte Carlo*.

1.1 Lançamentos de Uma Moeda ou Lançamento de Moedas

1.1.1 Lançamento de uma Moeda Uma Vez

Vamos iniciar a modelagem numérica utilizando um sistema muito simples, o lançamento de uma moeda. Vamos supor, que ao lançar a moeda, perdemos as informações sobre as condições iniciais e que ela vira muito e bata e rebata diversas vezes no chão. Fazemos todas essas suposições para dizer que não sabemos dizer de ante mão qual vai ser o resultado, se cara ou coroa. Como o resultado deste experimento é aleatório, podemos ter cara com probabilidade p ou coroa, com probabilidade $1 - p$. Vamos chamar a face cara de *sucesso*, que ocorre com probabilidade p e é representado pela variável aleatória $x = 1$. Chamamos a face coroa de *fracasso*, que ocorre com probabilidade $1 - p$ e é representado por $x = 0$. Em estatística, um experimento onde podem ocorrer somente dois resultados, chamados de sucesso ou fracasso, onde o sucesso, que acontece com probabilidade p é representado por $x = 1$ e o fracasso, que acontece com probabilidade $1 - p$ e é representado por $x = 0$ é chamado de *experimento de Bernoulli*.

No computador podemos simular este experimento simplesmente definido $x = 0$ e se $\text{rnd}(\text{seed}) \leq p$ então $x = 1$. Observe que as moedas podem ser viciadas ($p \neq 1/2$).

1.1.2 Lançamentos de uma Moeda Muitas Vezes

Vamos supor agora que a moeda considerada acima seja lançada n vezes. Neste caso os lançamentos são independentes (o resultado de um lançamento não depende de resultados anteriores) e o valor de p fica inalterado durante a realização dos N experimentos de Bernoulli. Desejamos saber qual a probabilidade de se ter $k = \sum_{i=1}^n x_i$ sucessos em n experimentos de Bernoulli com p fixo.

O modelo teórico neste caso, pode ser facilmente desenvolvido. A probabilidade de se ter k sucessos e $n - k$ fracassos é: $p^k(1 - p)^{n-k}$, que podem ser combinados de $n!/[k!(n - k)!]$ formas distintas. Assim, obtemos a distribuição binomial:

$$P_{n,p}(k) = \frac{n!}{k!(n - k)!} p^k(1 - p)^{n-k}, \quad (1.1)$$

com $k = 0, 1, \dots, n$.

Na simulação computacional, devemos construir um histograma $h(k)$, onde contamos o número de realização que tiveram k sucessos, nos n expe-

rimentos de Bernoulli, com p constante. Um algoritmo para esta simulação é:

1. Definições:

- $seed$ (semente para o gerador de números pseudo-aleatórios)
- $rnd(seed)$ (gerador de números pseudo-aleatórios)
- $m > 0$ (número de realizações dos n experimentos de Bernoulli)
- $n \geq 0$ (número de experimentos de Bernoulli)
- $0 \leq p \leq 1$ (probabilidade de sucesso em cada experimento de Bernoulli)
- $0 \leq k \leq n$ (número de sucessos nas n provas independentes de Bernoulli com p fixo)
- $h(0 : n)$ (histograma de k sucessos)

2. Inicializações:

- $seed = ?$
- $m = ?$
- $n = ?$
- $p = ?$
- $h(0 : n) = 0$

3. Laço das realizações ($i = 1, m$)

- $k = 0$
- Laço dos lançamentos ($j = 1, n$)
 - if $rnd(seed) \leq p$ then $k = k + 1$
- $h(k) = h(k) + 1/m$ (aqui somamos $1/m$ para que o histograma seja normalizado).

Devemos agora comparar o valor obtido com a simulação numérica de Monte Carlo $h(k)$ com o valor teórico $P_{n,p}(k)$.

1.2 Difusão de um Gás

Considere uma caixa fechada que é dividida em duas partes de volumes iguais. O volume do lado esquerdo, que está inicialmente fechado, contém um gás de N partículas idênticas. O volume do lado direito está vazio. Vamos agora considerar que um pequeno furo (que permita passar somente uma partícula por vez) seja feito conectando os volumes do lado esquerdo e direito. Para um tempo longo, no equilíbrio, o número médio de partículas no lado direito é igual ao do lado esquerdo. Sabemos também que no equilíbrio, a distribuição espacial de partículas é uniforme e que a distribuição de velocidades é dada pela distribuição de Maxwell-Boltzmann.

Para simular numericamente a movimentação das partículas de uma caixa para outra até se encontrar o equilíbrio pode ser realizada usando as leis de Newton e admitindo que as partículas são pontuais, i.e. uma partícula não colide com a outra (gás ideal). Nessa simulação, devemos considerar todas as condições iniciais e evoluir o sistema. Esta técnica é conhecida como *dinâmica molecular* e por sua evolução devemos chegar ao número de partículas de cada lado da caixa e também na densidades de probabilidade de posição e velocidade.

Uma outra maneira de fazer, uma vez que conhecemos a distribuição de posição e de velocidades é somente considerar o número de partículas de cada lado da caixa. Para faremos uma simulação numérica do tipo Monte Carlo.

Uma maneira de implementar este modelo é definir no código numérico um vetor $\vec{L} = \{l_1, l_2, \dots, l_N\}$, de N componentes, onde a componente $l_i = \pm 1$, expressa o fato da partícula i estar no lado esquerdo $l_i = -1$ ou direito $l_i = 1$ da caixa. Inicialmente todas as partículas estão do lado esquerdo da caixa, assim $l_i = -1$, para $i = 1, 2, \dots, N$. A cada intervalo de tempo assumimos que uma partícula troca de lado, assim podemos escolher uma partícula ao acaso e inverter o seu sinal. Para escolher uma partícula ao acaso, usamos o gerador de números pseudo-aleatório uniformemente distribuídos no intervalo $\text{rnd}(\text{seed}) \in [0, 1[$, onde seed é um inteiro que inicializa o rnd . Para escolher uma das N partículas com a mesma probabilidade, o índice k da partícula escolhida é dado por $k = \text{floor}(N * \text{rnd}(\text{seed})) + 1$, onde a função $\text{floor}(x)$ retorna somente a parte inteira de x , com $0 \leq N * \text{rnd}(\text{seed}) < N - 1$, devemos adicionar uma unidade para termos $1 \leq k \leq N$. Em seguida invertamos o sinal de $l_k = -l_k$. Este processo deve ser repetido $N_t \gg N$ vezes, para se alcançar o equilíbrio. Percebemos que no final do processo, no número de partículas na caixa do lado esquerdo n_l varia indefinidamente em torno de $N/2$, representando a flutuação de número de partículas na caixa do lado esquerdo no equilíbrio.

Este processo representa uma história da evolução do sistema. Outras

histórias são igualmente prováveis e devem ser consideradas. Para se calcular o valor médio do número de partículas na caixa do lado esquerdo, devemos considerar muitas histórias, ou realizações.

Como o nosso interesse é somente no número de partículas em um volume da caixa, assim n é o número de partículas do lado esquerdo, enquanto que $n' = N - n$ é o número de partículas do lado direito. Como cada partícula tem a mesma chance de passar de um lado para outro pelo buraco, a probabilidade por unidade de tempo para que uma partícula mude da esquerda para a direita é igual ao número de partículas do lado esquerdo dividido pelo número total de partículas n/N . O algoritmo para simular a evolução do modelo pode ser resumido nos seguintes passos:

1. Gere um número aleatório $0 \leq r < 1$ uniformemente distribuídos,
2. Se $r \leq n/N$, mova uma partícula da esquerda para direita, caso contrário, mova uma partícula da direita para esquerda.
3. Incremente o intervalo de tempo.

Aqui também, para se calcular o valor médio do número de partículas na caixa do lado esquerdo, devemos considerar muitas histórias, ou realizações.

O valor médio do número de partículas cai em função do tempo de seu valor inicial $\bar{n}_l(0) = N$ a $\bar{n}_l(\infty) = N/2$. Para determinar analiticamente este decaimento, vamos supor que $N \gg 1$ e que no instante t , se tenha em média $\bar{n}_l(t)$ partículas no lado esquerdo da caixa. A variação de $\bar{n}_l(t)$ em Δt , onde assumimos que Δt é muito maior do que uma única mudança de partícula, é dada por:

$$\Delta \bar{n}_l(t) = \left[-\frac{\bar{n}_l(t)}{N} + \frac{N - \bar{n}_l(t)}{N} \right] \Delta t, \quad (1.2)$$

a primeira parcela do lado direito da igualdade representa o número médio de partículas que saem do volume do lado esquerdo em Δt e a segunda parcela o número de partículas que entram. Fazendo $\Delta t \rightarrow 0$, têm-se:

$$\frac{d\bar{n}_l(t)}{dt} = 1 - \frac{2\bar{n}_l(t)}{N}, \quad (1.3)$$

cujas solução é obtida fazendo a mudança de variável $y = 1 - 2\bar{n}_l/N$, de modo que $dy = -(2/N)d\bar{n}_l$ e tenhamos $dy/y = -2dt/N$, o que leva a: $y(t) = e^{2t/N}$ e consequentemente:

$$\bar{n}_l(t) = \frac{N}{2} (1 + e^{-t/t_0}), \quad (1.4)$$

onde $t_0 = N/2$ é o tempo característico de retorno ao equilíbrio.

1.3 Decaimento Radioativo

Considere o decaimento radioativo. Um conjunto de núcleos radioativos decaem em outros núcleos sem saber, a priori, quais são os núcleos que irão decair. Se todos os núcleos de um material são idênticos, pode-se assumir que em um intervalo de tempo Δt , cada núcleo tem uma probabilidade p de decair. O algoritmo funciona da seguinte maneira.

1. escolha um entre os N núcleos
2. gere um número aleatório $0 \leq r < 1$ uniformemente distribuído.
3. Se $r \leq p$ então o núcleo decai, caso contrário não.
4. Cada um dos N núcleos são testados em cada intervalo de tempo.

A lei fundamental do decaimento radioativo é que a taxa de decaimento é proporcional ao número de núcleos instáveis N , assim:

$$\frac{dN}{dt} = -\lambda N \quad (1.5)$$

onde λ é a constante de decaimento.

1.4 Exercícios

1. Considere uma caixa fechada que é dividida em duas partes com volumes iguais. Um dos volumes contém inicialmente N partículas idênticas e o outro volume está vazio. Fazemos então um pequeno buraco na partição entre os dois volumes. O que acontece? Sabemos que após um intervalo de tempo, o sistema alcança o equilíbrio e o número médio de partículas em cada caixa é $N/2$.

Um modo simples de simular este sistema é considerar processos aleatórios. Neste processo considera-se que a movimentação das partículas é tão complexo que pode-se assumir como tendo um comportamento aleatório. Considerando que as partículas não interagem umas com as outras que modo que a probabilidade por unidade de tempo de uma partícula passar pelo buraco é a mesma independentemente do número de partículas de cada metade. Considera-se também o buraco seja bem pequeno para que somente uma partícula passe por intervalo de tempo.

- (a) Um modo de implementar este modelo é escolhendo uma partícula ao acaso e mudar ela de volume. No entanto considera-se aqui que as partículas são distinguíveis.

```

clear;
close all;
seed = 19;                // semente para gerador de # aleatorios
N = 100.;                 // # de particula total
r_max = 100;              // # max de realizacoes
t_max = 300;              // # tempo maximo
clf();
rand('seed',2);           // semente do gerador de #'s aleatorios
rand('uniform');          // define gerador
for r = 1:r_max
    parti = zeros(1:N);   // inicializa vetor de particulas
    for t = 1:t_max
        n = int(N*rand() + 1); // escolhe uma particula
        if parti(n) == 0 then // muda a particula de lado
            parti(n) = 1;
        else
            parti(n) = 0;
        end
        nr_t(t) = sum(parti)/N;
    end
    plot2d(1: t_max,nr_t);
end

```

- (b) Como o nosso interesse é somente no número de partículas em um volume da caixa, assim n é o número de partículas do lado esquerdo enquanto que $n' = N - n$ é o número de partículas do lado direito. Como cada partícula tem a mesma chance de passar de um lado para outro através do buraco, a probabilidade por unidade de tempo para que uma partícula mude da esquerda para a direita é igual ao número de partículas do lado esquerdo dividido pelo número total de partículas n/N . O algoritmo para simular a evolução do modelo pode ser resumido nos seguintes passos:

- i. Gere um número aleatório $0 \leq r < 1$ uniformemente distribuídos,
- ii. Se $r \leq n/N$, mova uma partícula da esquerda para direita, caso contrário, mova uma partícula da direita para esquerda.
- iii. Incremente o intervalo de tempo.

```

clear;
close all;
seed = 19;                // semente para gerador de # aleatorios

```

```

N      = 100.;           // # de particula total
r_max = 100;            // # max de realizacoes
t_max = 300;            // # tempo maximo
clf();
rand('seed',2);          // semente do gerador de #'s aleatorios
rand('uniform');         // define gerador
t1 = 0:t_max;
for r = 1:r_max
    nl = N;              // # de particulas na caixa da esquerda
    nr = 0.;             // # de particulas na caixa da direita
    nr_t(1,r) = [nr];    // inicializa curva
    for t = 1:t_max      // laco no tempo
        if rand() < nl/N then // nl/N propocao de part. na caixa esquerda
            nl = nl - 1.;    // tira uma particula da caixa esquerda
            nr = nr + 1.;    // coloca uma particula na caixa direita
        else
            nl = nl + 1.;    // coloca uma particula na caixa esquerda
            nr = nr - 1.;    // tira uma particula da caixa direita
        end
        nr_t(t+1,r) = nr/N; // constroe a curva n_r/N versus tempo
    end
    scf(0);
    plot2d([t1],[nr_t(:,r)],style=[-9]); // grafica as curvas
end

```

- (c) Faça um estudo estatístico das curvas obtidas calculando média, desvio-padrão e erro-padrão para cada ponto da curva.

```

clear;
N      = 100; // # de particula total
r_max = 100; // # max de realizacoes
t_max = 300; // # tempo maximo
clf();
rand('uniform');
rand('seed',2)
t1 = 0:t_max;
for r = 1:r_max
    nl = N; // # de particulas na caixa da esquerda
    nr = 0.; // # de particulas na caixa da direita
    nr_t(1,r) = [nr]; // inicializa curva
    for t = 1:t_max // laco no tempo
        if rand() < (nl/N) then // nl/N propocao de part. na caixa esquerda

```

```

        nl = nl - 1; // tira uma partícula da caixa esquerda
        nr = nr + 1; // coloca uma partícula na caixa direita
    else
        nl = nl + 1; // coloca uma partícula na caixa esquerda
        nr = nr - 1; // tira uma partícula da caixa direita
    end;
    nr_t(t+1,r) = nr/N; // constroa a curva n_r/N versus tempo
end
scf(0);
plot2d([t1],[nr_t(:,r)],style=[-9]); // grafica as curvas
end
m_nr_t = zeros(1,t_max+1); // curva de valor medio
m_nr2_t = zeros(1,t_max+1); // curva de valor ao quadrado medio
m_sup = zeros(1,t_max+1); // curva de valor ao quadrado medio
m_inf = zeros(1,t_max+1); // curva de valor ao quadrado medio
for t = 1:t_max+1
    for r = 1:r_max
        m_nr_t(t) = m_nr_t(t) + nr_t(t,r);
        m_nr2_t(t) = m_nr2_t(t) + nr_t(t,r)*nr_t(t,r);
    end
    m_nr_t(t) = m_nr_t(t)/(r_max); // valor medio
    m_nr2_t(t) = m_nr2_t(t)/(r_max); // valor quadratico medio
    m_nr2_t(t) = sqrt(abs(m_nr2_t(t) - m_nr_t(t)*m_nr_t(t))); // desvio padrao
    m_inf(t) = m_nr_t(t) - m_nr2_t(t)/sqrt(r_max); // media - erro padrao
    m_sup(t) = m_nr_t(t) + m_nr2_t(t)/sqrt(r_max); // media + erro padrao
end // erro-padrao = desvio-padrao/
// raiz(realizacoes)
//clf();
scf(1);
//figure(2),
plot([t1],[m_nr_t],[t1],[m_inf],[t1],[m_sup]);
legend("valor medio","media - erro-padrao","media + erro-padrao");

```

(d) Mostre analiticamente que:

$$n(t) = \frac{N}{2} [1 + e^{-2t/N}] ?$$

2. Considere o decaimento radioativo. Um conjunto de núcleos radioativos decaem em outros núcleos sem saber, a priori, quais são os núcleos que irão decair. Se todos os núcleos de um material são idênticos, pode-se assumir que em um intervalo de tempo Δt , cada núcleo tem uma probabilidade p de decair. O algoritmo funciona da seguinte maneira.

- (a) escolha um entre os N núcleos
- (b) gere um número aleatório $0 \leq r < 1$ uniformemente distribuído.
- (c) Se $r \leq p$ então o núcleo decai, caso contrário não.
- (d) Cada um dos N núcleos são testados em cada intervalo de tempo.

Implemente este algoritmo e tente resolver o problema analiticamente para validar ou refutar os resultados encontrados pela simulação numérica.

```
clear;
lambda= input("Entre com o modulo do coeficiente de decaimento:") // 0.001
N =1/lambda; // # de partícula total
r_max = 100; // # max de realizacoes
t_max = 300; // # tempo maximo
clf();
rand('uniform');
rand('seed',2)
t1 = 0:t_max;
for r = 1:r_max
    n      = N;      // # de particulas
    nd      = 0.;     // # de particulas decaidas
    nd_t(1,r) = [nd]; // inicializa curva
    for t = 1:t_max // laco no tempo
        if rand() < (n/N) then // n/N propocao de part. nao decaidas
            n = n - 1;          // decaimento de uma partícula
            nd = nd + 1;        // partícula decaída
        end;
        nd_t(t+1,r) = -nd/N; // constroe a curva n_d/N versus tempo,para o decaim
    end
    scf(0);
    plot2d(t1,[nd_t(:,r)],style=[-9]); // grafica as curvas
end
m_nr_t  = zeros(1,t_max+1); // curva de valor medio
m_nr2_t = zeros(1,t_max+1); // curva de valor ao quadrado medio
m_sup   = zeros(1,t_max+1); // curva de valor ao quadrado medio
m_inf   = zeros(1,t_max+1); // curva de valor ao quadrado medio
for t    = 1:t_max+1
    for r    = 1:r_max
        m_nr_t(t)  = m_nr_t(t) + nd_t(t,r);
        m_nr2_t(t) = m_nr2_t(t) + nd_t(t,r)*nd_t(t,r);
    end
end
```

```

m_nr_t(t) = m_nr_t(t)/(r_max); // valor medio
m_nr2_t(t) = m_nr2_t(t)/(r_max); // valor quadrado
m_nr2_t(t) = sqrt(abs(m_nr2_t(t) - m_nr_t(t)*m_nr_t(t))); // desvio padrao
m_inf(t) = m_nr_t(t) - m_nr2_t(t)/sqrt(r_max); // media - erro-pad
m_sup(t) = m_nr_t(t) + m_nr2_t(t)/sqrt(r_max); // media + erro-pad
end // erro-pad

scf(1);
plot([t1],[m_nr_t],[t1],[m_inf],[t1],[m_sup]); //figure(2)
legend("valor medio","media - erro-pad","media + erro-pad"); //legenda

```

1.5 Genética de Populações

O texto em seguida foi adaptado do minicurso “Uma introdução à dinâmica estocástica de populações” ministrado por Leonardo Paulo Maia, no Encontro Regional de Matemática Aplicada e Computacional, 18-20/06/2008, Bauru, SP.

O filamento de um dos dois ácidos nucleicos, o DNA¹ ou RNA² (material genético) presente nas células de um organismo vivo contém toda a informação necessária para gerá-lo. Considera-se a seguir somente o DNA, que é uma sequência de nucleotídeos ou bases. Um *locus* é um trecho específico de uma sequência de DNA contém as informações necessárias para a realização de funções específicas (por exemplo, gerar proteínas). Um locus gênico é ocupado por um gene, que é um segmento específico de DNA que contém informação para desempenhar a função específica ao locus que ele ocupa. Um mesmo gene pode ser encontrado em diversas formas, semelhantes, mas não idênticas, que diferem umas das outras por alguns nucleotídeos. Cada uma dessas variantes é denominada de *alelo* e uma propriedade manifestada pela ação de um alelo em um certo ambiente é conhecido como *fenótipo*. Um *cromossomo* é uma sequência de DNA que contém diversos genes, entre outros segmentos. Cada organismo tem sua identidade genética registrada em uma família de cromossomos, que é o *genótipo* daquele organismo. Se concentrarmos nossa atenção em um particular locus gênico do genótipo de todos os organismos de uma população, portanto estudamos uma população de genes. A *genética de populações* é o estudo de genes ou conjunto de genes que formam uma população levando em conta princípios da genética. O objetivo é obter a distribuição de probabilidade de cada alelo.

¹ do inglês: ácido desoxirribonucleico.

² ácido ribonucleico.

1.5.1 Modelo de Wright-Fisher

O modelo de Wright-Fisher é o ponto de partida para o estudo de *genética de populações*. O modelo se propõe a dizer as proporções em que os *alelos* aparecem em uma população de tamanho finito. Através dos estados absorventes, ele explica como que as flutuações no número de alelos produzem a *deriva genética*.

Dois Alelos e a Extinção de um Deles

Vamos considerar uma população de N alelos que podem ter dois tipos. Esta população é representada por uma urna e os alelos são representados por bolas idênticas, com exceção da cor, que podem ser brancas ou pretas. A cor da bola representa o tipo de alelo. Vamos considerar intervalos de tempo discretos e a cada instante existem $n_1^{(t)}$ bolas brancas (alelos tipo 1), que podem ter valores no domínio dos inteiros entre 0 e N e $n_2^{(t)} = N - n_1^{(t)}$ bolas pretas (alelos tipo 2). O tamanho da população não depende $N = n_1^{(t)} + n_2^{(t)}$, do tempo. Considere uma nova urna vazia e N extrações independentes e com reposição de bolas da primeira urna. A cada extração, uma nova bola da mesma cor da bola retirada da primeira urna é colocada na segunda urna. Deste modo, no instante $t + 1$, na segunda urna, tem-se $n_1^{(t+1)}$ bolas brancas (alelos tipo 1) e $n_2^{(t+1)} = N - n_1^{(t+1)}$ bolas pretas (alelos tipo 2). A probabilidade de passar de $n_1^{(t)}$ bolas brancas para $n_1^{(t+1)}$ bolas brancas no instante $t + 1$, é dada pela distribuição binomial:

$$P[n_1^{(t)} \rightarrow n_1^{(t+1)}] = \binom{N}{n_1^{(t+1)}} p_1^{n_1^{(t+1)}}(t) p_2^{n_2^{(t+1)}}(t) \quad (1.6)$$

$$n_2^{(t+1)} = N - n_1^{(t+1)} \quad (1.7)$$

$$p_1(t) = \frac{n_1^{(t)}}{N} \quad (1.8)$$

$$p_2(t) = \frac{n_2^{(t)}}{N} = 1 - p_1(t) . \quad (1.9)$$

Observe que o valor médio de $n_1^{(t+1)}$ vale: $\langle n_1^{(t+1)} \rangle = N p_1(t) = n_1^{(t)}$ e sua variância $\sigma_{n_1^{(t)}}^2 = N p_1(t) p_2(t) = n_1^{(t)} p_2(t) = n_1^{(t)} n_2^{(t)} / N$.

Para representar este processo considere a evolução temporal da condição inicial $n_1^{(0)} \rightarrow n_1^{(1)} \rightarrow n_1^{(2)} \rightarrow \dots \rightarrow n_1^{(t)}$ até o instante t . Gostaríamos também de dizer o valor estacionário $n_1^{(\infty)}$, para quando $t \gg 1$.

Se a condição inicial for nula, ou seja $n_1^{(0)} = 0$, então a probabilidade de propagar uma bola branca é $p_1(0) = 0$, conseqüentemente $n_1^{(1)} = 0$ e $p_1(1) = 0$

assim por diante até $n_1^{(\infty)} = 0$, então o estado sem bolas brancas (alelos tipo 1) é chamado de *estado absorvente*. Uma vez que o sistema entrou em um estado absorvente, em algum instante t , $n_1^{(t)} = 0$, todos os valores posteriores também serão nulos. É possível ter $n_1^{(t)} = 0$, através de flutuações dos valores n_1 . O mesmo acontece para $n_1^{(0)} = N$, neste caso $p_2(0) = 0$, a probabilidade de propagar uma bola preta é nula, consequentemente todas as bolas serão brancas $n_1^{(1)} = N$ e $p_2(1) = 0$ assim por diante até $n_1^{(\infty)} = N$, então o estado com todas as bolas brancas (alelos tipo 1) é chamado de *estado absorvente*. Através das flutuações, uma vez o sistema entrou neste estado, em algum instante t , $n_1^{(t)} = N$, todos os valores posteriores também serão nulos. Para qualquer outra condição inicial $n_1^{(0)} \neq 0$ e $n_1^{(0)} \neq N$, o sistema varia até que algum instante t , $n_1^{(t)} = 0$ ou $n_1^{(t)} = N$, e o sistema entre em um estado absorvente e um dos tipo de alelos sejam extintos.

Seleção Natural. Para incluir a *seleção natural* no modelo de Wright-Fisher, vamos especificar as adaptabilidades (“fitness”) dos alelos 1 por w_1 e 2 por w_2 , que são pesos probabilísticos. Desde modo devemos considerar a probabilidade $p_1(t)$ e $p_2(t)$ das Eqs. 1.8 e 1.9 como

$$\begin{aligned} \tilde{p}_1(t) &= \frac{w_1 n_1^{(t)}}{w_1 n_1^{(t)} + w_2 n_2^{(t)}} = \frac{w_1 n_1^{(t)}}{w_1 n_1^{(t)} + w_2 (N - n_1^{(t)})} \\ &= \frac{n_1^{(t)}}{n_1^{(t)} + s(N - n_1^{(t)})} = \frac{1}{1 - s(1 - N/n_1^{(t)})} \\ &= \frac{1}{1 - s[1 - 1/p_1(t)]} \end{aligned} \quad (1.10)$$

$$s = \frac{w_2}{w_1} \quad (1.11)$$

$$\tilde{p}_2(t) = 1 - \tilde{p}_1(t) = , \quad (1.12)$$

observe que os pesos não são relevantes mas a razão entre eles dada por s (Eq. 1.11). A probabilidade de transição Eq. 1.6 se transforma em:

$$P[n_1^{(t)} \rightarrow n_1^{(t+1)}] = \binom{N}{n_1^{(t+1)}} \tilde{p}_1^{n_1^{(t+1)}}(t) \tilde{p}_2^{N-n_1^{(t+1)}}(t) , \quad (1.13)$$

onde $\tilde{p}_2(t)$ é dada pela Eq. 1.12 e $n_2^{(t+1)} = N - n_1^{(t+1)}$.

Mutação. Para levar a *mutação* em consideração no modelo, vamos considerar que um alelo extraído pode mutar para o alelo oposto com probabilidade $p^{(m)}$. Desde modo devemos considerar a probabilidade $\tilde{p}_1(t)$ e $\tilde{p}_2(2)$ das

Eqs. 1.10 e 1.12 como

$$\hat{p}_1(t) = [1 - p^{(m)}]\tilde{p}_1(t) + p^{(m)}[1 - \tilde{p}_1(t)] \quad (1.14)$$

$$\hat{p}_2(t) = 1 - \hat{p}_1(t) = , \quad (1.15)$$

e a probabilidade de transição Eq. 1.13 passa a ser escrita como:

$$P[n_1^{(t)} \rightarrow n_1^{(t+1)}] = \binom{N}{n_1^{(t+1)}} \hat{p}_1^{n_1^{(t+1)}}(t) \hat{p}_2^{n_2^{(t+1)}}(t) , \quad (1.16)$$

onde $\hat{p}_2(t)$ é dada pela Eq. 1.15 e $n_2^{(t+1)} = N - n_1^{(t+1)}$.

A Eq. 1.16 é a equação mais geral do modelo de Wright-Fisher de dois alelos, pois ela inclui a seleção através da razão dos pesos probabilísticos s (Eq. 1.11), referente à adaptabilidade de cada alelo e leva em consideração a mutação entre os alelos através da probabilidade $p^{(m)}$. Para desconsiderar a mutação devemos considerar $p_m = 0$ e para desconsiderar a seleção devemos considerar $s = 1$. Fazendo ambas considerações, reencontramos a Eq. 1.6.

Vários Alelos

Para representar este processo para M alelos considere o vetor:

$$\vec{x}^{(t)} = \begin{pmatrix} n_1^{(t)} \\ \vdots \\ n_M^{(t)} \end{pmatrix} , \quad (1.17)$$

onde $n_M^{(t)} = N - \sum_{i=1}^{M-1} n_i^{(t)}$, de modo que somente $M - 1$ coordenadas do vetor são independentes.

Dado a condição inicial \vec{x}^0 , estamos interessados em entender a evolução temporal desse sistema e poder dizer as componentes do vetor $\vec{x}^{(t)}$, no instante t . Gostaríamos também de dizer para onde o sistema converge quando $t \gg 1$, regime estacionário.

No instante $t + 1$, a probabilidade de passar de $n_1^{(t)}$ alelos do tipo 1 para $n_1^{(t+1)}$ alelos do tipo 1, $n_2^{(t)}$ alelos do tipo 2 para $n_2^{(t+1)}$ alelos do tipo 2 e assim

por diante, é dada pela distribuição Multinomial:

$$P[\vec{x}^{(t)} \rightarrow \vec{x}^{(t+1)}] = \left(\frac{N!}{\prod_{i=1}^M n_i^{(t)}} \right) \prod_{i=1}^M p_i^{n_i^{(t+1)}}(t) \quad (1.18)$$

$$\sum_{i=1}^M n_i^{(t+1)} = N \quad (1.19)$$

$$p_i(t) = \frac{n_i^{(t)}}{N} \quad (1.20)$$

$$\sum_{i=1}^M p_i(t) = 1 \dots \quad (1.21)$$

Para $M = 2$, reencontramos a equações para dois alelos. Observe que o valor médio de $n_1^{(t+1)}$ vale: $\langle n_1^{(t+1)} \rangle = N p_1(t) = n_1^{(t)}$ e sua variância $\sigma_{n_1^{(t)}}^2 = N p_1(t) p_2(t) = n_1^{(t)} n_2^{(t)} / N$.

Seleção Natural. Para incluir a *seleção natural* no modelo de M alelos, vamos especificar as adaptabilidades (“fitness”) dos alelos 1 por w_i , com $i = 1, 2, \dots, M$, que são pesos probabilísticos. Desde modo devemos considerar a probabilidade $p_i(t)$ das Eqs. 1.20 como:

$$\begin{aligned} \tilde{p}_i(t) &= \frac{w_i n_i^{(t)}}{\sum_{i=1}^M w_i n_i^{(t)}} \\ \sum_{i=1}^M \tilde{p}_i(t) &= 1 \dots \end{aligned} \quad (1.22)$$

A probabilidade de transição Eq. 1.18 se transforma em:

$$P[\vec{x}^{(t)} \rightarrow \vec{x}^{(t+1)}] = \left(\frac{N!}{\prod_{i=1}^M n_i^{(t)}} \right) \prod_{i=1}^M \tilde{p}_i^{n_i^{(t+1)}}(t) \quad (1.23)$$

onde $\sum_{i=1}^M p_i(t) = 1$ e $\sum_{i=1}^M n_i^{(t+1)} = N$.

Mutação. Para levar a *mutação* em consideração no modelo, vamos considerar que um alelo extraído i pode mutar para o alelo j com probabilidade $p_{i,j}^{(m)}$.

1.5.2 Modelo de Moran

O modelo de Wright-Fisher não permite um tratamento analítico exato, pois ele permite um dado alelo efetue passe a ter qualquer no intervalo de tempo seguinte. Além disso ele não permite a coexistência entre ascendentes e descendentes. Uma alternativa interessante ao modelo de Wright-Fisher é o modelo de Moran, pois ele permite soluções exatas em alguns casos, superposição de alelos entre gerações e tratamento tanto no caso discreto quanto no contínuo.

Dois Alelos e a Extinção de um Deles

Vamos considerar uma população de N alelos que podem ter dois tipos. Esta população é representada por uma urna e os alelos são representados por bolas idênticas, com exceção da cor, que podem ser brancas ou pretas. A cor da bola representa o alelo tipo 1. Vamos considerar intervalos de tempo discretos e a cada instante existem $n_1^{(t)}$ bolas brancas (alelos tipo 1), que podem ter valores no domínio dos inteiros entre 0 e N e $n_2^{(t)} = N - n_1^{(t)}$ bolas pretas (alelos tipo 2). O tamanho da população não depende $N = n_1^{(t)} + n_2^{(t)}$, do tempo. Considere duas extrações independentes e com reposição de bolas da urna. Na primeira extração, diz qual a cor da bola que vamos adicionar à urna (qual alelo será replicado) a segunda extração qual bola será retirada da urna para manter N constante. Observe que neste processo $n_1^{t+1} = n_1^t + n$, onde n é uma variável aleatória que pode assumir os seguintes valores: $n = -1, 0, 1$.

A probabilidade de extrair uma bola branca na primeira extração é $n_1^{(t)}/N$. Esta bola é replicada e uma delas volta à urna. A probabilidade de extrair uma bola preta é $1 - n_1^{(t)}/N$, assim temos $n_1^{(t+1)} = n_1^{(t)} + 1$. A probabilidade de extrair uma bola branca é $n_1^{(t)}/N$, assim temos $n_1^{(t+1)} = n_1^{(t)}$. A probabilidade de extrair uma bola preta na primeira extração é $1 - n_1^{(t)}/N$. Esta bola é replicada e uma delas volta à urna. A probabilidade de extrair uma bola preta é $1 - n_1^{(t)}/N$, assim temos $n_1^{(t+1)} = n_1^{(t)}$. A probabilidade de extrair uma bola branca é $n_1^{(t)}/N$, assim temos $n_1^{(t+1)} = n_1^{(t)} - 1$.

As probabilidades de transição são:

$$\begin{aligned} P[n_1^{(t)} \rightarrow n_1^{(t+1)} = n_1^{(t)} - 1] &= \left(1 - \frac{n_1^{(t)}}{N}\right) \frac{n_1^{(t)}}{N} \\ P[n_1^{(t)} \rightarrow n_1^{(t+1)} = n_1^{(t)}] &= 1 - 2\frac{n_1^{(t)}}{N} \left(1 - \frac{n_1^{(t)}}{N}\right) \\ P[n_1^{(t)} \rightarrow n_1^{(t+1)} = n_1^{(t)} + 1] &= \frac{n_1^{(t)}}{N} \left(1 - \frac{n_1^{(t)}}{N}\right). \end{aligned}$$

Observe que o valor médio de $n_1^{(t+1)}$ vale: $\langle n_1^{(t+1)} \rangle = \text{????}$ e sua variância $\sigma_{n_1^{(t)}}^2 = \text{????}$.

Para representar este processo considere a evolução temporal da condição inicial $n_1^{(0)} \rightarrow n_1^{(1)} \rightarrow n_1^{(2)} \rightarrow \dots \rightarrow n_1^{(t)}$ até o instante t . Gostaríamos também de dizer o valor estacionário $n_1^{(\infty)}$, para quando $t \gg 1$.

Se a condição inicial for nula, ou seja $n_1^{(0)} = 0$, então a probabilidade de propagar um bola branca é $p_1(0) = 0$, conseqüentemente $n_1^{(1)} = 0$ e $p_1(1) = 0$ assim por diante até $n_1^{(\infty)} = 0$, então o estado sem bolas brancas (alelos tipo 1) é chamado de *estado absorvente*. Uma vez que o sistema entrou em um estado absorvente, em algum instante t , $n_1^{(t)} = 0$, todos os valores posteriores também serão nulos. É possível ter $n_1^{(t)} = 0$, através de flutuações dos valores n_1 . O mesmo acontece para $n_1^{(0)} = N$, neste caso $p_2(0) = 0$, a probabilidade de propagar uma bola preta é nula, conseqüentemente todas as bolas serão brancas $n_1^{(1)} = N$ e $p_2(1) = 0$ assim por diante até $n_1^{(\infty)} = N$, então o estado com todas as bolas brancas (alelos tipo 1) é chamado de *estado absorvente*. Através das flutuações, uma vez o sistema entrou neste estado, em algum instante t , $n_1^{(t)} = N$, todos os valores posteriores também serão nulos. Para qualquer outra condição inicial $n_1^{(0)} \neq 0$ e $n_1^{(0)} \neq N$, o sistema varia até que algum instante t , $n_1^{(t)} = 0$ ou $n_1^{(t)} = N$, e o sistema entre em um estado absorvente e um dos tipo de alelos sejam extintos.

Chamando $i = n_1$ as seguintes probabilidade de transição:

$$p_{i,j} = \begin{cases} 0 & \text{se } |j - i| \geq 2 \\ \frac{i(N-i)}{N^2} & \text{se } |j - i| = 1 \\ 1 - \frac{2i(N-i)}{N^2} & \text{se } |j - i| = 0 \end{cases} \quad (1.24)$$

Considere que a probabilidade de uma realização da evolução de Moran atingir o estado absorvente N seja x_i , dado a condição inicial $n_1^{(0)} = i$, com $i = 0, 1, 2, \dots, N$. A probabilidade de se atingir o estado absorvente 0 é $1 - x_i$.

Para valores de $i = 0$ ou $i = N$, o processo já se inicia em um estado absorvente assim $x_0 = 0$ ou $x_N = 1$, respectivamente. Para $i = 1, 2, \dots, N - 1$, temos a recorrência

$$x_i = p_{i,i-1}x_{i-1} + (1 - p_{i,i-1} - p_{i,i+1})x_i + p_{i,i+1}x_{i+1}, \quad (1.25)$$

que pode ser reescrita como: $p_{i,i+1}(x_{i+1} - x_i) = p_{i,i-1}(x_i - x_{i-1})$. Chamando $y_i = x_i - x_{i-1}$, têm-se: $p_{i,i+1}y_{i+1} = p_{i,i-1}y_i$, chamando

$$\sigma_i = \frac{p_{i,i-1}}{p_{i,i+1}}, \quad (1.26)$$

resulta na relação:

$$y_{i+1} = \sigma_i y_i \quad (1.27)$$

com $y_1 = x_1$, a solução para com $i \geq 2$ é $y_i = (\prod_{k=1}^{i-1} \sigma_k) y_1 = (\prod_{k=1}^{i-1} \sigma_k) x_1$. Como $x_i = \sum_{j=1}^i y_j$ e $\sum_{j=1}^N y_j = 1$, temos que: $1 = x_1 + \sum_{j=2}^N (\prod_{k=1}^{j-1} \sigma_k) x_1 = x_1(1 + \sum_{j=2}^N \prod_{k=1}^{j-1} \sigma_k)$, o que resulta em:

$$x_1 = \frac{1}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^j \sigma_k}. \quad (1.28)$$

Como $y_i = (\prod_{k=1}^{i-1} \sigma_k) x_1$ e $x_i = \sum_{j=1}^i y_j$, temos $x_i = x_1 + \sum_{j=2}^i (\prod_{k=1}^{j-1} \sigma_k) x_1 = (1 + \sum_{j=1}^{i-1} \prod_{k=1}^j \sigma_k) x_1$, usando o valor de x_1 , têm-se:

$$x_i = \frac{1 + \sum_{j=1}^{i-1} \prod_{k=1}^j \sigma_k}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^j \sigma_k}. \quad (1.29)$$

Muitas vezes, é importante saber se uma população pura pode ser invadida por um organismo mutante ou migrante. No caso neutro (sem seleção), já vimos que a probabilidade de fixação de um particular indivíduo é $1/N$, onde N é o tamanho da população. Se a probabilidade de um único alelo alienígena se fixar for maior do que $1/N$, ele certamente é mais apto do que a variedade pré-existente, o que contudo, não garante sua fixação. De qualquer forma, é interessante estimar essa probabilidade de fixação. Imagine que em uma população de N alelos do tipo 2 ocorra uma mutação e surge um alelo do tipo 1, a probabilidade de fixação do alelo do tipo 1 é

$$\rho = x_1 = \frac{1}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^j \sigma_k} \quad (1.30)$$

É possível mostrar que a probabilidade de fixação de um alelo do tipo 1 é

$$\rho = \frac{1 - 1/s}{1 - 1/s^N} \quad (1.31)$$

se ele tem valor adaptativo $s \neq 1$ e os alelos do tipo 2 tem valor adaptativo unitário.