Substitution models

Continuous-time Markov chains

I. Holmes

Department of Bioengineering
University of California, Berkeley

Spring semester

Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

Discrete State Spaces in Biology

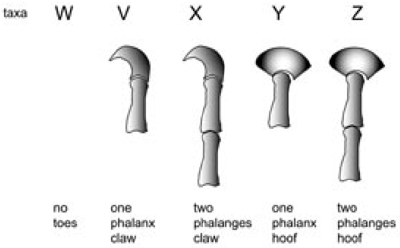
- Alphabets Ω: nucleotides, amino acids
  - Short strings Ω^N: codons, RNA basepairs, TF binding sites
  - Arbitrary-length strings Ω^*: proteins, viral genomes, chromosomes
    - Periodic strings: circular plasmids, bacterial genomes
  - Vectors of strings $[\Omega^*]^N$, e.g. genomes
- Integers: microsatellite repeat lengths, population size, allele frequency
  - Integer vectors: molecules, species populations
- Molecular states, e.g. gating of ion channels (“open” or “closed”; conductance measurable by patch clamp)
- Molecular ensembles, e.g. gene regulatory circuits

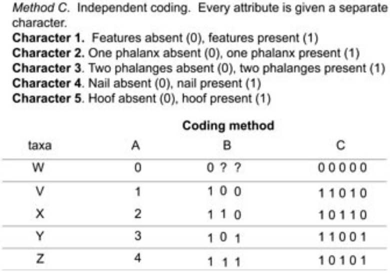
Discrete State Spaces in Biology

- Alphabets Ω: nucleotides, amino acids
  - Short strings Ω^N: codons, RNA basepairs, TF binding sites
  - Arbitrary-length strings Ω^*: proteins, viral genomes, chromosomes
    - Periodic strings: circular plasmids, bacterial genomes
  - Vectors of strings $[\Omega^*]^N$, e.g. genomes
- Integers: microsatellite repeat lengths, population size, allele frequency
  - Integer vectors: molecules, species populations
- Molecular states, e.g. gating of ion channels (“open” or “closed”; conductance measurable by patch clamp)
- Molecular ensembles, e.g. gene regulatory circuits

Discrete Non-Molecular Characters

We will mostly consider evolution of sequence residues; however, evolving “characters” need not be residues:





Method C. Independent coding. Every attribute is given a separate character.
Character 1. Features absent (0), features present (1)
Character 2. One phalanx absent (0), one phalanx present (1)
Character 3. Two phalanges absent (0), two phalanges present (1)
Character 4. Nail absent (0), nail present (1)
Character 5. Hoof absent (0), hoof present (1)

		Coding method	
taxa	A	B	C
W	0	0 ? ?	00000
V	1	1 0 0	11010
X	2	1 1 0	10110
Y	3	1 0 1	11001
Z	4	1 1 1	10101

Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

Discrete-Time Discrete-State Markov Chains

- Matrix notation for equation of state
  - Discrete-time: $p_j^{[n+1]} = \sum_i p_i^{[n]} Q_{ij}$ where $p_j^{[n]} = P(x_n = j)$ and $Q_{ij} = P(x_{n+1} = j | x_n = i)$
    - BTW, can write this as a graphical model $x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \ldots$
  - Matrix form: $\mathbf{p}^{[n+1]} = \mathbf{p}^{[n]}\mathbf{Q}$

Discrete-Time Discrete-State Markov Chains

- Matrix notation for equation of state
  - Discrete-time: $p_j^{[n+1]} = \sum_i p_i^{[n]} Q_{ij}$ where $p_j^{[n]} = P(x_n = j)$ and $Q_{ij} = P(x_{n+1} = j | x_n = i)$
    - BTW, can write this as a graphical model $x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \ldots$
  - Matrix form: $\mathbf{p}^{[n+1]} = \mathbf{p}^{[n]}\mathbf{Q}$

Discrete-Time Discrete-State Markov Chains

- Matrix notation for equation of state
  - Discrete-time: $p_j^{[n+1]} = \sum_i p_i^{[n]} Q_{ij}$ where $p_j^{[n]} = P(x_n = j)$ and $Q_{ij} = P(x_{n+1} = j | x_n = i)$
    - BTW, can write this as a graphical model $x_1 \to x_2 \to x_3 \to x_4 \ldots$
  - Matrix form: $\mathbf{p}^{[n+1]} = \mathbf{p}^{[n]}\mathbf{Q}$

Continuous-Time Discrete-State Markov Chains

- Discrete→continuous: suppose each step $n \to n+1$ represents a time interval Δt
  - We must now think in terms of the *rate* R_{ij} of mutation $i \to j$, i.e. the "probability per unit time" of the mutation.
  - Conventionally, define $R_{ii} = -\sum_{i \neq j} R_{ij}$, i.e. the negative "exit rate" from state i
  - Then $\mathbf{Q} = \mathbf{I} + \mathbf{R}\Delta t$ where $\sum_j R_{ij} = 0 \; \forall \; i$, so that $\Delta \mathbf{p} = \mathbf{p}\mathbf{R}\Delta t$
  - Continuous limit: $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$
- In general can have $\mathbf{R} \equiv \mathbf{R}(t)$ but will mainly consider homogeneous chains where $\mathbf{R}$ is constant

Continuous-Time Discrete-State Markov Chains

- Discrete→continuous: suppose each step $n \to n+1$ represents a time interval Δt
    - We must now think in terms of the *rate* R_{ij} of mutation $i \to j$, i.e. the "probability per unit time" of the mutation.
    - Conventionally, define $R_{ii} = -\sum_{i \neq j} R_{ij}$, i.e. the negative "exit rate" from state i
    - Then $\mathbf{Q} = \mathbf{I} + \mathbf{R}\Delta t$ where $\sum_j R_{ij} = 0 \; \forall \; i$, so that $\Delta \mathbf{p} = \mathbf{p}\mathbf{R}\Delta t$
    - Continuous limit: $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$
- In general can have $\mathbf{R} \equiv \mathbf{R}(t)$ but will mainly consider homogeneous chains where $\mathbf{R}$ is constant

Continuous-Time Discrete-State Markov Chains

- Discrete→continuous: suppose each step $n \to n+1$ represents a time interval Δt
  - We must now think in terms of the *rate* R_{ij} of mutation $i \to j$, i.e. the "probability per unit time" of the mutation.
  - Conventionally, define $R_{ii} = -\sum_{i \neq j} R_{ij}$, i.e. the negative "exit rate" from state i
  - Then $\mathbf{Q} = \mathbf{I} + \mathbf{R}\Delta t$ where $\sum_j R_{ij} = 0 \ \forall\ i$, so that $\Delta\mathbf{p} = \mathbf{p}\mathbf{R}\Delta t$
  - Continuous limit: $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$
- In general can have $\mathbf{R} \equiv \mathbf{R}(t)$ but will mainly consider homogeneous chains where $\mathbf{R}$ is constant

Continuous-Time Discrete-State Markov Chains

- Discrete→continuous: suppose each step $n \to n+1$ represents a time interval Δt
  - We must now think in terms of the *rate* R_{ij} of mutation $i \to j$, i.e. the “probability per unit time” of the mutation.
  - Conventionally, define $R_{ii} = -\sum_{i \neq j} R_{ij}$, i.e. the negative “exit rate” from state i
  - Then $\mathbf{Q} = \mathbf{I} + \mathbf{R}\Delta t$ where $\sum_j R_{ij} = 0 \; \forall \; i$, so that $\Delta \mathbf{p} = \mathbf{p}\mathbf{R}\Delta t$
  - Continuous limit: $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$
- In general can have $\mathbf{R} \equiv \mathbf{R}(t)$ but will mainly consider homogeneous chains where $\mathbf{R}$ is constant

Equilibrium and Ergodicity of a Markov Chain

- Equilibrium π: $\pi\mathbf{Q} = \pi$ (discrete), $\pi\mathbf{R} = \mathbf{0}$ (continuous)
    - Finite chains must have an equilibrium; infinite chains, not necessarily
    - Practical issue: how to find the equilibrium? QR decomposition works
    - A chain at equilibrium $\forall$ times t is called stationary
- Ergodicity implies a path between any two states; for continuous chains, stronger implications about rates

Reversibility of a Markov Chain

- Reversibility: $\pi_i R_{ij} = \pi_j R_{ji}$
  - Flow of probability from $i \rightarrow j$ exactly balances $j \rightarrow i$ at equilibrium (detailed balance)
  - At instantaneous level, forward & backward mutations indistinguishable
  - Implies $\exists$ symmetric matrix: $S_{ij} = R_{ij}\sqrt{\pi_i/\pi_j}$, so $\mathbf{S} = \mathbf{P}^{1/2}\ \mathbf{R}\ \mathbf{P}^{-1/2}$ where $\mathbf{P}$ is diagonal and $\mathbf{P}_{kk} = \pi_k$
  - Equilibrium doesn't imply reversibility; e.g. (contrived) unidirectional cycler $A \rightarrow C \rightarrow G \rightarrow T \rightarrow A$
- Origins of irreversibility; energy consumption (e.g. ATP)

Reversibility of a Markov Chain

- Reversibility: $\pi_i R_{ij} = \pi_j R_{ji}$
  - Flow of probability from $i \to j$ exactly balances $j \to i$ at equilibrium (detailed balance)
  - At instantaneous level, forward & backward mutations indistinguishable
  - Implies $\exists$ symmetric matrix: $S_{ij} = R_{ij}\sqrt{\pi_i/\pi_j}$, so $\mathbf{S} = \mathbf{P}^{1/2}\ \mathbf{R}\ \mathbf{P}^{-1/2}$ where $\mathbf{P}$ is diagonal and $\mathbf{P}_{kk} = \pi_k$
  - Equilibrium doesn't imply reversibility; e.g. (contrived) unidirectional cycler $A \to C \to G \to T \to A$
- Origins of irreversibility; energy consumption (e.g. ATP)

Reversibility of a Markov Chain

- Reversibility: $\pi_i R_{ij} = \pi_j R_{ji}$
    - Flow of probability from $i \to j$ exactly balances $j \to i$ at equilibrium (detailed balance)
    - At instantaneous level, forward & backward mutations indistinguishable
    - Implies $\exists$ symmetric matrix: $S_{ij} = R_{ij}\sqrt{\pi_i/\pi_j}$, so $\mathbf{S} = \mathbf{P}^{1/2}\,\mathbf{R}\,\mathbf{P}^{-1/2}$ where $\mathbf{P}$ is diagonal and $\mathbf{P}_{kk} = \pi_k$
    - Equilibrium doesn't imply reversibility; e.g. (contrived) unidirectional cycler $A \to C \to G \to T \to A$
- Origins of irreversibility; energy consumption (e.g. ATP)

Reversibility of a Markov Chain

- Reversibility: $\pi_i R_{ij} = \pi_j R_{ji}$
  - Flow of probability from $i \rightarrow j$ exactly balances $j \rightarrow i$ at equilibrium (detailed balance)
  - At instantaneous level, forward & backward mutations indistinguishable
  - Implies $\exists$ symmetric matrix: $S_{ij} = R_{ij}\sqrt{\pi_i/\pi_j}$, so $\mathbf{S} = \mathbf{P}^{1/2}\ \mathbf{R}\ \mathbf{P}^{-1/2}$ where $\mathbf{P}$ is diagonal and $\mathbf{P}_{kk} = \pi_k$
  - Equilibrium doesn't imply reversibility; e.g. (contrived) unidirectional cycler $A \rightarrow C \rightarrow G \rightarrow T \rightarrow A$
- Origins of irreversibility; energy consumption (e.g. ATP)

Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

Point Substitution Processes

- Examples: nucleotides, amino acids, RNA basepairs, codons, discrete characters
- Intuitive analogy to scalar difference processes & ODEs
  - Solution to $\mathbf{p}^{[t+1]} = \mathbf{p}^{[t]}\mathbf{Q}$ is clearly $\mathbf{p}^{[t]} = \mathbf{p}^{[0]}\mathbf{Q}^t$
    - Analogy to corresponding one-dimensional difference equation: $p^{[t+1]} = p^{[t]}Q \Rightarrow p^{[t]} = p^{[0]}Q^t$
  - Consider one-dimensional ODE, $\frac{dp}{dt} = pR$. Solution is $p(t) = p(0)\exp(Rt)$
  - By analogy, would like solution of $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$ to be $\mathbf{p}(t) = \mathbf{p}(0)\mathbf{M}(t)$ where $\mathbf{M}(t) = \exp(\mathbf{R}t)$

Point Substitution Processes

- Examples: nucleotides, amino acids, RNA basepairs, codons, discrete characters
- Intuitive analogy to scalar difference processes & ODEs
  - Solution to $\mathbf{p}^{[t+1]} = \mathbf{p}^{[t]}\mathbf{Q}$ is clearly $\mathbf{p}^{[t]} = \mathbf{p}^{[0]}\mathbf{Q}^t$
    - Analogy to corresponding one-dimensional difference equation: $p^{[t+1]} = p^{[t]}Q \Rightarrow p^{[t]} = p^{[0]}Q^t$
  - Consider one-dimensional ODE, $\frac{dp}{dt} = pR$. Solution is $p(t) = p(0)\exp(Rt)$
  - By analogy, would like solution of $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$ to be $\mathbf{p}(t) = \mathbf{p}(0)\mathbf{M}(t)$ where $\mathbf{M}(t) = \exp(\mathbf{R}t)$

Point Substitution Processes

- Examples: nucleotides, amino acids, RNA basepairs, codons, discrete characters
- Intuitive analogy to scalar difference processes & ODEs
  - Solution to $\mathbf{p}^{[t+1]} = \mathbf{p}^{[t]}\mathbf{Q}$ is clearly $\mathbf{p}^{[t]} = \mathbf{p}^{[0]}\mathbf{Q}^t$
    - Analogy to corresponding one-dimensional difference equation: $p^{[t+1]} = p^{[t]}Q \;\Rightarrow\; p^{[t]} = p^{[0]}Q^t$
  - Consider one-dimensional ODE, $\frac{dp}{dt} = pR$. Solution is $p(t) = p(0)\exp(Rt)$
  - By analogy, would like solution of $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$ to be $\mathbf{p}(t) = \mathbf{p}(0)\mathbf{M}(t)$ where $\mathbf{M}(t) = \exp(\mathbf{R}t)$

Point Substitution Processes

- Examples: nucleotides, amino acids, RNA basepairs, codons, discrete characters
- Intuitive analogy to scalar difference processes & ODEs
    - Solution to $\mathbf{p}^{[t+1]} = \mathbf{p}^{[t]}\mathbf{Q}$ is clearly $\mathbf{p}^{[t]} = \mathbf{p}^{[0]}\mathbf{Q}^t$
        - Analogy to corresponding one-dimensional difference equation: $p^{[t+1]} = p^{[t]}Q \;\Rightarrow\; p^{[t]} = p^{[0]}Q^t$
    - Consider one-dimensional ODE, $\frac{dp}{dt} = pR$. Solution is $p(t) = p(0)\exp(Rt)$
    - By analogy, would like solution of $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$ to be $\mathbf{p}(t) = \mathbf{p}(0)\mathbf{M}(t)$ where $\mathbf{M}(t) = \exp(\mathbf{R}t)$

Dreaming of a Matrix Exponential

- Would like solution of $\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{R}$ to be $\mathbf{p}(t) = \mathbf{p}(0)\mathbf{M}(t)$ where $\mathbf{M}(t) = \exp(\mathbf{R}t)$
- Here we've introduced (or wished for) the important *matrix exponential*, $\exp(\mathbf{A})$

Definition of the Matrix Exponential

- The matrix exponential
  - Can define e.g. by Taylor series, $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n/n!$
  - Alternative definition of matrix exponential uses limit of discrete-time approximation:

$$\exp(\mathbf{R}t) = \lim_{\Delta t \to 0} (\mathbf{I} + \mathbf{R}\Delta t)^{t/\Delta t}$$

  - While these are useful definitions, $\exp(\mathbf{A})$ is typically computed in other ways
    - See e.g. Moler & van Loan (2003). Nineteen Dubious Ways to Compute the Exponential of a Matrix, 25 years later

Definition of the Matrix Exponential

- The matrix exponential
  - Can define e.g. by Taylor series, $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n/n!$
  - Alternative definition of matrix exponential uses limit of discrete-time approximation:

$$\exp(\mathbf{R}t) = \lim_{\Delta t \to 0} (\mathbf{I} + \mathbf{R}\Delta t)^{t/\Delta t}$$

  - While these are useful definitions, $\exp(\mathbf{A})$ is typically computed in other ways
    - See e.g. Moler & van Loan (2003). Nineteen Dubious Ways to Compute the Exponential of a Matrix, 25 years later

Definition of the Matrix Exponential

- The matrix exponential
  - Can define e.g. by Taylor series, $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n / n!$
  - Alternative definition of matrix exponential uses limit of discrete-time approximation:

$$\exp(\mathbf{R}t) = \lim_{\Delta t \to 0} (\mathbf{I} + \mathbf{R}\Delta t)^{t/\Delta t}$$

  - While these are useful definitions, $\exp(\mathbf{A})$ is typically computed in other ways
    - See e.g. Moler & van Loan (2003). Nineteen Dubious Ways to Compute the Exponential of a Matrix, 25 years later

Chain Rule & the Matrix Exponential

- Taylor series definition: $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n / n!$
  - Note however that we have to be careful about transferring properties “blindly” over from scalar exponential
  - For example, if $\mathbf{A} \equiv \mathbf{A}(x, \ldots)$ then, in general, $\frac{\partial}{\partial x}[\exp(\mathbf{A})] = \frac{\partial \mathbf{A}}{\partial x} \exp(\mathbf{A})$ does **NOT** hold
    - this is because $\frac{\partial}{\partial x} \exp(\mathbf{A})$ contains terms of form $\sum_{k=0}^{n-1} [\mathbf{A}^k] \frac{\partial \mathbf{A}}{\partial x} [\mathbf{A}^{n-k-1}]$
    - such terms are *not* in general equal to $n \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{n-1}$, because $\mathbf{A}$ and $\frac{\partial \mathbf{A}}{\partial x}$ may not commute
    - exception is if these two matrices do, in fact, commute; then we recover the identity from the scalar case
    - obvious case is when $\mathbf{A} \equiv \mathbf{R}t$ and $x \equiv t$, so $\frac{d}{dt} \exp(\mathbf{R}t) = \exp(\mathbf{R}t)\mathbf{R}$. This means our Taylor definition works!

Chain Rule & the Matrix Exponential

- Taylor series definition: $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n/n!$
  - Note however that we have to be careful about transferring properties “blindly” over from scalar exponential
  - For example, if $\mathbf{A} \equiv \mathbf{A}(x,\ldots)$ then, in general, $\frac{\partial}{\partial x}[\exp(\mathbf{A})] = \frac{\partial \mathbf{A}}{\partial x}\exp(\mathbf{A})$ does **NOT** hold
    - this is because $\frac{\partial}{\partial x}\exp(\mathbf{A})$ contains terms of form $\sum_{k=0}^{n-1}[\mathbf{A}^k]\frac{\partial \mathbf{A}}{\partial x}[\mathbf{A}^{n-k-1}]$
    - such terms are *not* in general equal to $n\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{n-1}$, because $\mathbf{A}$ and $\frac{\partial \mathbf{A}}{\partial x}$ may not commute
    - exception is if these two matrices do, in fact, commute; then we recover the identity from the scalar case
    - obvious case is when $\mathbf{A} \equiv \mathbf{R}t$ and $x \equiv t$, so $\frac{d}{dt}\exp(\mathbf{R}t) = \exp(\mathbf{R}t)\mathbf{R}$. This means our Taylor definition works!

Chain Rule & the Matrix Exponential

- Taylor series definition: $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n / n!$
    - Note however that we have to be careful about transferring properties "blindly" over from scalar exponential
    - For example, if $\mathbf{A} \equiv \mathbf{A}(x, \ldots)$ then, in general, $\frac{\partial}{\partial x}[\exp(\mathbf{A})] = \frac{\partial \mathbf{A}}{\partial x} \exp(\mathbf{A})$ does **NOT** hold
        - this is because $\frac{\partial}{\partial x} \exp(\mathbf{A})$ contains terms of form $\sum_{k=0}^{n-1} [\mathbf{A}^k] \frac{\partial \mathbf{A}}{\partial x} [\mathbf{A}^{n-k-1}]$
        - such terms are *not* in general equal to $n \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{n-1}$, because $\mathbf{A}$ and $\frac{\partial \mathbf{A}}{\partial x}$ may not commute
        - exception is if these two matrices do, in fact, commute; then we recover the identity from the scalar case
        - obvious case is when $\mathbf{A} \equiv \mathbf{R}t$ and $x \equiv t$, so $\frac{d}{dt} \exp(\mathbf{R}t) = \exp(\mathbf{R}t)\mathbf{R}$. This means our Taylor definition works!

Chain Rule & the Matrix Exponential

- Taylor series definition: $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n / n!$
  - Note however that we have to be careful about transferring properties “blindly” over from scalar exponential
  - For example, if $\mathbf{A} \equiv \mathbf{A}(x, \ldots)$ then, in general, $\frac{\partial}{\partial x}[\exp(\mathbf{A})] = \frac{\partial \mathbf{A}}{\partial x} \exp(\mathbf{A})$ does **NOT** hold
    - this is because $\frac{\partial}{\partial x} \exp(\mathbf{A})$ contains terms of form $\sum_{k=0}^{n-1} [\mathbf{A}^k] \frac{\partial \mathbf{A}}{\partial x} [\mathbf{A}^{n-k-1}]$
    - such terms are *not* in general equal to $n \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{n-1}$, because $\mathbf{A}$ and $\frac{\partial \mathbf{A}}{\partial x}$ may not commute
    - exception is if these two matrices do, in fact, commute; then we recover the identity from the scalar case
    - obvious case is when $\mathbf{A} \equiv \mathbf{R}t$ and $x \equiv t$, so $\frac{d}{dt} \exp(\mathbf{R}t) = \exp(\mathbf{R}t)\mathbf{R}$. This means our Taylor definition works!

Chapman-Kolmogorov Equation & the Matrix Exponential

- The matrix exponential
  - Likewise, in general $\exp(\mathbf{A})\exp(\mathbf{B})$ does not equal $\exp(\mathbf{A}+\mathbf{B})$. However, $\mathbf{M}(t+t') = \mathbf{M}(t)\mathbf{M}(t')$.
  - This is a form of the Chapman-Kolmogorov equation, often used as the principal definition of a Markov chain.
    - Chapman-Kolmogorov is usually written $\sum_j P(x_t = j|x_0 = i)P(x_{t+t'} = k|x_t = j) = P(x_{t+t'} = k|x_0 = i)$
  - It can also be used as an implementation shortcut/approximation to computing $M(t)$

Chapman-Kolmogorov Equation & the Matrix Exponential

- The matrix exponential
  - Likewise, in general $\exp(\mathbf{A})\exp(\mathbf{B})$ does not equal $\exp(\mathbf{A}+\mathbf{B})$. However, $\mathbf{M}(t+t') = \mathbf{M}(t)\mathbf{M}(t')$.
  - This is a form of the Chapman-Kolmogorov equation, often used as the principal definition of a Markov chain.
    - Chapman-Kolmogorov is usually written $\sum_j P(x_t = j | x_0 = i) P(x_{t+t'} = k | x_t = j) = P(x_{t+t'} = k | x_0 = i)$
  - It can also be used as an implementation shortcut/approximation to computing $M(t)$

Implementation of the Matrix Exponential

- For Markov chains, $\mathbf{M}(t) = \sum_{n=0}^{\infty} \mathbf{R}^n t^n / n!$. If t is small, then $t^n/n! \to 0$ for larger n, i.e. can approximate $\mathbf{M}(t)$ with finite polynomial (e.g. Padé approximation)
- This is particularly time-efficient if $\mathbf{R}$ is sparse, in which case $\mathbf{R}^n$ will also be sparse for small n
- For finite chains, can do exact calculations using eigenvalues & eigenvectors

Implementation of the Matrix Exponential

- For Markov chains, $\mathbf{M}(t) = \sum_{n=0}^{\infty} \mathbf{R}^n t^n / n!$. If t is small, then $t^n/n! \to 0$ for larger n, i.e. can approximate $\mathbf{M}(t)$ with finite polynomial (e.g. Padé approximation)
- This is particularly time-efficient if $\mathbf{R}$ is sparse, in which case $\mathbf{R}^n$ will also be sparse for small n
- For finite chains, can do exact calculations using eigenvalues & eigenvectors

Implementation of the Matrix Exponential

- For Markov chains, $\mathbf{M}(t) = \sum_{n=0}^{\infty} \mathbf{R}^n t^n / n!$. If t is small, then $t^n/n! \to 0$ for larger n, i.e. can approximate $\mathbf{M}(t)$ with finite polynomial (e.g. Padé approximation)
- This is particularly time-efficient if $\mathbf{R}$ is sparse, in which case $\mathbf{R}^n$ will also be sparse for small n
- For finite chains, can do exact calculations using eigenvalues & eigenvectors

Matrix Diagonalisation

- Review of matrix diagonalisation
  - Matrix $\mathbf{A}$, eigenvalue λ: right (column) eigenvector $\mathbf{Au} = \lambda\mathbf{u}$, left (row) eigenvector $\mathbf{vA} = \lambda\mathbf{v}$
  - Eigenvalues are solutions of characteristic equation, $|\mathbf{A} - \lambda\mathbf{I}| = 0$
  - Assume matrix of right eigenvectors is invertible (prove it!)
  - Can write $\mathbf{A} = \mathbf{UDU}^{-1}$ where $\mathbf{D}$ is diagonal ($D_{kk} = \lambda_k$), $\mathbf{u}^{(k)}$ is k'th column of $\mathbf{U}$, $\mathbf{v}^{(k)}$ is k'th row of $\mathbf{U}^{-1}$
  - Thus $\mathbf{A}^n = \mathbf{UD}^n\mathbf{U}^{-1}$ and $\exp(\mathbf{A}) = \mathbf{U}\,\exp(\mathbf{D})\,\mathbf{U}^{-1}$
  - If $\mathbf{A}$ is symmetric then left and right eigenvectors can be chosen to be the same, $\mathbf{U}^{-1} = \mathbf{U}^T$, and all λ_k are real

Matrix Diagonalisation

- Review of matrix diagonalisation
  - Matrix $\mathbf{A}$, eigenvalue λ: right (column) eigenvector $\mathbf{Au} = \lambda \mathbf{u}$, left (row) eigenvector $\mathbf{vA} = \lambda \mathbf{v}$
  - Eigenvalues are solutions of characteristic equation, $|\mathbf{A} - \lambda \mathbf{I}| = 0$
  - Assume matrix of right eigenvectors is invertible (prove it!)
  - Can write $\mathbf{A} = \mathbf{UDU}^{-1}$ where $\mathbf{D}$ is diagonal ($D_{kk} = \lambda_k$), $\mathbf{u}^{(k)}$ is k'th column of $\mathbf{U}$, $\mathbf{v}^{(k)}$ is k'th row of $\mathbf{U}^{-1}$
  - Thus $\mathbf{A}^n = \mathbf{UD}^n\mathbf{U}^{-1}$ and $\exp(\mathbf{A}) = \mathbf{U}\ \exp(\mathbf{D})\ \mathbf{U}^{-1}$
  - If $\mathbf{A}$ is symmetric then left and right eigenvectors can be chosen to be the same, $\mathbf{U}^{-1} = \mathbf{U}^T$, and all λ_k are real

Matrix Diagonalisation

- Review of matrix diagonalisation
  - Matrix $\mathbf{A}$, eigenvalue λ: right (column) eigenvector $\mathbf{A}\mathbf{u} = \lambda \mathbf{u}$, left (row) eigenvector $\mathbf{v}\mathbf{A} = \lambda \mathbf{v}$
  - Eigenvalues are solutions of characteristic equation, $|\mathbf{A} - \lambda \mathbf{I}| = 0$
  - Assume matrix of right eigenvectors is invertible (prove it!)
  - Can write $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$ where $\mathbf{D}$ is diagonal ($D_{kk} = \lambda_k$), $\mathbf{u}^{(k)}$ is k'th column of $\mathbf{U}$, $\mathbf{v}^{(k)}$ is k'th row of $\mathbf{U}^{-1}$
  - Thus $\mathbf{A}^n = \mathbf{U}\mathbf{D}^n\mathbf{U}^{-1}$ and $\exp(\mathbf{A}) = \mathbf{U} \, \exp(\mathbf{D}) \, \mathbf{U}^{-1}$
  - If $\mathbf{A}$ is symmetric then left and right eigenvectors can be chosen to be the same, $\mathbf{U}^{-1} = \mathbf{U}^T$, and all λ_k are real

Matrix Diagonalisation

- Review of matrix diagonalisation
  - Matrix $\mathbf{A}$, eigenvalue λ: right (column) eigenvector $\mathbf{Au} = \lambda\mathbf{u}$, left (row) eigenvector $\mathbf{vA} = \lambda\mathbf{v}$
  - Eigenvalues are solutions of characteristic equation, $|\mathbf{A} - \lambda\mathbf{I}| = 0$
  - Assume matrix of right eigenvectors is invertible (prove it!)
  - Can write $\mathbf{A} = \mathbf{UDU}^{-1}$ where $\mathbf{D}$ is diagonal ($D_{kk} = \lambda_k$), $\mathbf{u}^{(k)}$ is k'th column of $\mathbf{U}$, $\mathbf{v}^{(k)}$ is k'th row of $\mathbf{U}^{-1}$
  - Thus $\mathbf{A}^n = \mathbf{UD}^n\mathbf{U}^{-1}$ and $\exp(\mathbf{A}) = \mathbf{U}\ \exp(\mathbf{D})\ \mathbf{U}^{-1}$
  - If $\mathbf{A}$ is symmetric then left and right eigenvectors can be chosen to be the same, $\mathbf{U}^{-1} = \mathbf{U}^T$, and all λ_k are real

Matrix Exponentiation by Diagonalisation

- General properties of eigenvalues and eigenvectors
  - Zero eigenvalue(s) correspond to equilibrium distribution
    - More than one zero eval ⇒ orthogonal equilibria ⇒ noncommunicating state cliques
    - That at least one eval is zero can be seen by considering $\mathbf{Rx} = \mathbf{0}$ where $x_i = 1 \; \forall \; i$
  - All remaining eigenvalues of $\mathbf{R}$ are negative; eigenvectors correspond to "modes of information loss"
    - Exercise: Prove that evals of a rate matrix are negative
  - Eigenvalues are either real, or come in complex conjugate pairs (see later section)

Matrix Exponentiation by Diagonalisation

- General properties of eigenvalues and eigenvectors
  - Zero eigenvalue(s) correspond to equilibrium distribution
    - More than one zero eval ⇒ orthogonal equilibria ⇒ noncommunicating state cliques
    - That at least one eval is zero can be seen by considering $\mathbf{Rx} = \mathbf{0}$ where $x_i = 1 \ \forall \ i$
  - All remaining eigenvalues of $\mathbf{R}$ are negative; eigenvectors correspond to "modes of information loss"
    - Exercise: Prove that evals of a rate matrix are negative
  - Eigenvalues are either real, or come in complex conjugate pairs (see later section)

Matrix Exponentiation by Diagonalisation

- General properties of eigenvalues and eigenvectors
  - Zero eigenvalue(s) correspond to equilibrium distribution
    - More than one zero eval ⇒ orthogonal equilibria ⇒ noncommunicating state cliques
    - That at least one eval is zero can be seen by considering $\mathbf{Rx} = \mathbf{0}$ where $x_i = 1 \; \forall \; i$
  - All remaining eigenvalues of **R** are negative; eigenvectors correspond to "modes of information loss"
    - Exercise: Prove that evals of a rate matrix are negative
  - Eigenvalues are either real, or come in complex conjugate pairs (see later section)

Matrix Exponentiation by Diagonalisation

- Reversible processes: similarity transform converts **R** to symmetric matrix
  - Algorithms to diagonalise symmetric matrices are simpler & stabler than those for asymmetric matrices
  - Recall, for reversible models, $\mathbf{R} = \mathbf{P}^{-1/2}\ \mathbf{S}\ \mathbf{P}^{1/2}$ where **P** is diagonal and **S** is symmetric
    - Not difficult to show that evals of **S** are those of **R**, and evecs are trivially related by **P**
  - Given π ($\Rightarrow$ **P**), we can work with evals & evecs of **S** rather than **R**
  - In particular, $\mathbf{R}^N = \mathbf{P}^{-1/2}\ \mathbf{S}^N\ \mathbf{P}^{1/2}$ and $\exp(\mathbf{R}t) = \mathbf{P}^{-1/2}\ \exp(\mathbf{S}t)\ \mathbf{P}^{1/2}$
- Irreversible processes
  - Can group complex eigenvalues into conjugate pairs (corresponding to recurrent aperiodic behaviour)

Matrix Exponentiation by Diagonalisation

- Reversible processes: similarity transform converts $\mathbf{R}$ to symmetric matrix
  - Algorithms to diagonalise symmetric matrices are simpler & stabler than those for asymmetric matrices
  - Recall, for reversible models, $\mathbf{R} = \mathbf{P}^{-1/2}\ \mathbf{S}\ \mathbf{P}^{1/2}$ where $\mathbf{P}$ is diagonal and $\mathbf{S}$ is symmetric
    - Not difficult to show that evals of $\mathbf{S}$ are those of $\mathbf{R}$, and evecs are trivially related by $\mathbf{P}$
  - Given π ($\Rightarrow \mathbf{P}$), we can work with evals & evecs of $\mathbf{S}$ rather than $\mathbf{R}$
  - In particular, $\mathbf{R}^N = \mathbf{P}^{-1/2}\ \mathbf{S}^N\ \mathbf{P}^{1/2}$ and $\exp(\mathbf{R}t) = \mathbf{P}^{-1/2}\ \exp(\mathbf{S}t)\ \mathbf{P}^{1/2}$
- Irreversible processes
  - Can group complex eigenvalues into conjugate pairs (corresponding to recurrent aperiodic behaviour)

Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

Simulating Trajectories from Markov Chains

- Simulation by discrete-time approximation
  - Break time interval t into $N = t/\Delta t$ discrete steps; probability matrix for each step is $\mathbf{Q} \simeq \mathbf{I} + \mathbf{R}\Delta t$
  - At each step x_n, sample x_{n+1} using $P(x_{n+1} = j | x_n = i) = Q_{ij}$
  - Drawbacks: accurate only if $\Delta t \ll \max_i(-1/R_{ii})$
    - Exercise: derive this bound

Simulating Trajectories from Markov Chains

- Simulation by discrete-time approximation
  - Break time interval t into $N = t/\Delta t$ discrete steps; probability matrix for each step is $\mathbf{Q} \simeq \mathbf{I} + \mathbf{R}\Delta t$
  - At each step x_n, sample x_{n+1} using $P(x_{n+1} = j | x_n = i) = Q_{ij}$
  - Drawbacks: accurate only if $\Delta t \ll \max_i(-1/R_{ii})$
    - Exercise: derive this bound

Simulating Trajectories from Markov Chains

- Simulation using the "embedded random walk"
  - Let $r_i = -R_{ii}$ be *exit rate* from state i, and let $s_j^{(i)} = R_{ij}/r_i$ be *jump probability* for $i \to j$ (with $i \neq j$) and $s^(i)_i = 0$
  - Each row of $\mathbf{R}$ is specified by an exit rate r_i and a vector $\mathbf{s}^{(i)}$ of jump probabilities
  - Simulating the process can be conceptualised as follows:
    - Wait for time t before exiting current state
    - The pdf of t is $p(t) = r_i \exp(-r_i t) = r_i \times P(\text{no mutation at time } t)$
    - Randomly pick the next state using weights $\mathbf{s}^{(i)}$

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Let $r_i = -R_{ii}$ be *exit rate* from state i, and let $s_j^{(i)} = R_{ij}/r_i$ be *jump probability* for $i \to j$ (with $i \neq j$) and $s^(i)_i = 0$
  - Each row of **R** is specified by an exit rate r_i and a vector $\mathbf{s}^{(i)}$ of jump probabilities
  - Simulating the process can be conceptualised as follows:
    - Wait for time t before exiting current state
    - The pdf of t is $p(t) = r_i \exp(-r_i t) = r_i \times P(\text{no mutation at time } t)$
    - Randomly pick the next state using weights $\mathbf{s}^{(i)}$

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Let $r_i = -R_{ii}$ be *exit rate* from state i, and let $s_j^{(i)} = R_{ij}/r_i$ be *jump probability* for $i \to j$ (with $i \neq j$) and $s^(i)_i = 0$
  - Each row of $\mathbf{R}$ is specified by an exit rate r_i and a vector $\mathbf{s}^{(i)}$ of jump probabilities
  - Simulating the process can be conceptualised as follows:
    - Wait for time t before exiting current state
    - The pdf of t is $p(t) = r_i \exp(-r_i t) = r_i \times P(\text{no mutation at time } t)$
    - Randomly pick the next state using weights $\mathbf{s}^{(i)}$

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Sampling the wait time: use transformation of variables
    - Let v be an r.v. uniformly distributed from 0 to 1 (so pdf is $q(v) = 1$ for $0 \leq v \leq 1$); this is easy to sample
    - Let $t = -\frac{1}{r_i} \log[1 - v]$, so that $v = 1 - \exp(-r_i t)$
    - Then $p(t) = q(v) \frac{dv}{dt} = r_i \exp(-r_i t)$ as required.
    - More generally, we can use this method to sample from any pdf whose cumulative distribution can be inverted
  - This algorithm allows us to simulate efficiently while keeping track of precise trajectories, including event times, as well as various summary statistics such as time spent in each state (w_i), number of substitution events (u_{ij}), etc.

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Sampling the wait time: use transformation of variables
    - Let v be an r.v. uniformly distributed from 0 to 1 (so pdf is $q(v) = 1$ for $0 \leq v \leq 1$); this is easy to sample
    - Let $t = -\frac{1}{r_i} \log[1 - v]$, so that $v = 1 - \exp(-r_i t)$
    - Then $p(t) = q(v)\frac{dv}{dt} = r_i \exp(-r_i t)$ as required.
    - More generally, we can use this method to sample from any pdf whose cumulative distribution can be inverted
  - This algorithm allows us to simulate efficiently while keeping track of precise trajectories, including event times, as well as various summary statistics such as time spent in each state (w_i), number of substitution events (u_{ij}), etc.

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Sampling the wait time: use transformation of variables
    - Let v be an r.v. uniformly distributed from 0 to 1 (so pdf is $q(v) = 1$ for $0 \leq v \leq 1$); this is easy to sample
    - Let $t = -\frac{1}{r_i}\log[1 - v]$, so that $v = 1 - \exp(-r_i t)$
    - Then $p(t) = q(v)\frac{dv}{dt} = r_i \exp(-r_i t)$ as required.
    - More generally, we can use this method to sample from any pdf whose cumulative distribution can be inverted
  - This algorithm allows us to simulate efficiently while keeping track of precise trajectories, including event times, as well as various summary statistics such as time spent in each state (w_i), number of substitution events (u_{ij}), etc.

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Sampling the wait time: use transformation of variables
    - Let v be an r.v. uniformly distributed from 0 to 1 (so pdf is $q(v) = 1$ for $0 \leq v \leq 1$); this is easy to sample
    - Let $t = -\frac{1}{r_i} \log[1 - v]$, so that $v = 1 - \exp(-r_i t)$
    - Then $p(t) = q(v) \frac{dv}{dt} = r_i \exp(-r_i t)$ as required.
    - More generally, we can use this method to sample from any pdf whose cumulative distribution can be inverted
  - This algorithm allows us to simulate efficiently while keeping track of precise trajectories, including event times, as well as various summary statistics such as time spent in each state (w_i), number of substitution events (u_{ij}), etc.

Simulating Trajectories from Markov Chains

- Simulation using the “embedded random walk”
  - Sampling the wait time: use transformation of variables
    - Let v be an r.v. uniformly distributed from 0 to 1 (so pdf is $q(v) = 1$ for $0 \leq v \leq 1$); this is easy to sample
    - Let $t = -\frac{1}{r_i} \log[1 - v]$, so that $v = 1 - \exp(-r_i t)$
    - Then $p(t) = q(v) \frac{dv}{dt} = r_i \exp(-r_i t)$ as required.
    - More generally, we can use this method to sample from any pdf whose cumulative distribution can be inverted
  - This algorithm allows us to simulate efficiently while keeping track of precise trajectories, including event times, as well as various summary statistics such as time spent in each state (w_i), number of substitution events (u_{ij}), etc.

Gillespie's Algorithm

An example of this simulation is Gillespie's algorithm from computational chemistry (J. Phys. Chem. 1977) Let $x_i(t)$ represent the number of molecules of species i at time t. Suppose that reaction μ has instantaneous rate c_μ and requires $n_i(\mu)$ molecules of species i. The total rate of this reaction is

$$a_\mu = c_\mu \prod_i \binom{x_i}{n_i(\mu)}$$

Total reaction rate is $r = \sum_\mu a_\mu$. Time to next reaction $\sim \mathrm{Exp}(r)$. P(next reaction is type $\mu) = a_\mu / r$. Assuming all molecular collisions result in a reaction, the individual reaction rates c_μ are determined by the physical properties of the molecules (mass, volume) and the temperature of the system via Boltzmann distributions. (Of course these assumptions are probably false, e.g. only some proportion of collisions will result in a reaction.)

Simulated Generation of Tree Topologies

- A couple of ways to generate an ultrametric phylogenetic tree by simulation (NB in an ultrametric tree, all leafs are same distance from root):
    - Yule process (forwards in time). Speciation rate $N\lambda$ where N is number of species. Stop after fixed time.
    - Coalescent process (backwards in time). Coalescence rate $\frac{1}{2}N(N-1)\kappa$.
- Can also generate non-ultrametric trees, e.g. with a birth-death process.

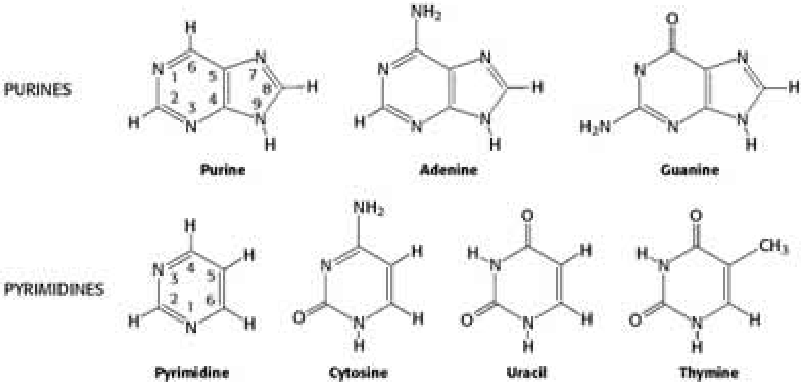
Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

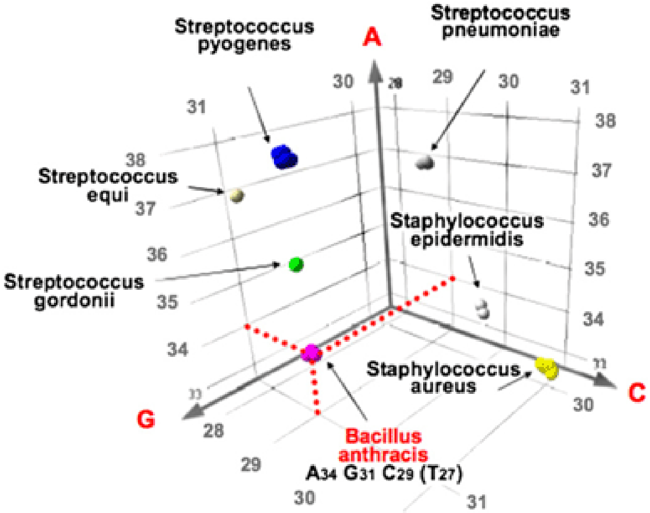
Nucleotide Substitution Models

1. Jukes and Cantor, 1969. Evolution of protein molecules.
2. Kimura M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J Mol Evol 1980 16(2)
3. Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 1981 17(6)
4. Hasegawa M, Kishino H, Yano T. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol 1985 22(2)
5. Yang Z. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J Mol Evol 1994 39(3)
6. Yang Z. Estimating the pattern of nucleotide substitution. J Mol Evol 1994 39(1)

Purines and pyrimidines



Basepair composition



The Jukes-Cantor Model (1969)

$$\mathbf{R} = \left(\begin{array}{ccccc} & A & C & G & T \\ A & -3\lambda & \lambda & \lambda & \lambda \\ C & \lambda & -3\lambda & \lambda & \lambda \\ G & \lambda & \lambda & -3\lambda & \lambda \\ T & \lambda & \lambda & \lambda & -3\lambda \end{array}\right)$$

Conventional to choose one substitution per unit time ($\lambda = 1/3$), however model is easier to solve if we rescale to one *replacement* per unit time, where a “replacement” can be an unobservable substitution-to-self ($\lambda = 1/4$).

Kimura's Two-Parameter Model (1980)

$$\mathbf{R} = \left(\begin{array}{ccccc} & A & C & G & T \\ A & -\alpha-2\beta & \beta & \alpha & \beta \\ C & \beta & -\alpha-2\beta & \beta & \alpha \\ G & \alpha & \beta & -\alpha-2\beta & \beta \\ T & \beta & \alpha & \beta & -\alpha-2\beta \end{array} \right)$$

Transition rate α, transversion rate β

Sometimes use transition/transversion ratio $\kappa = \alpha/\beta$

Eigenvectors represent "forgetting which purine", "forgetting which pyrimidine", "forgetting purine vs pyrimidine"

NB uniform equilibrium distribution (as with Jukes-Cantor)

Felsenstein’s Row-Independent Model (1981)

$$\mathbf{R} \;=\; \rho \times \begin{pmatrix} \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \\ \pi_A & \pi_C & \pi_G & \pi_T \end{pmatrix} - \rho \times \mathbf{1}$$

where ρ is the substitution rate, **1** the identity matrix and π a probability vector that (by construction) is the equilibrium probability

Hasegawa-Kishino-Yano (1985)

$$\mathbf{R} = \left(\begin{array}{ccccc} & A & C & G & T \\ A & * & \beta\pi_C & \alpha\pi_G & \beta\pi_T \\ C & \beta\pi_A & * & \beta\pi_G & \alpha\pi_T \\ G & \alpha\pi_A & \beta\pi_C & * & \beta\pi_T \\ T & \beta\pi_A & \alpha\pi_C & \beta\pi_G & * \end{array} \right)$$

Transition rate α, transversion rate β

Combines elements of K80 (transition/transversion) and F81 (non-uniform)

Diagonal elements omitted for clarity.

Can still be diagonalized algebraically!

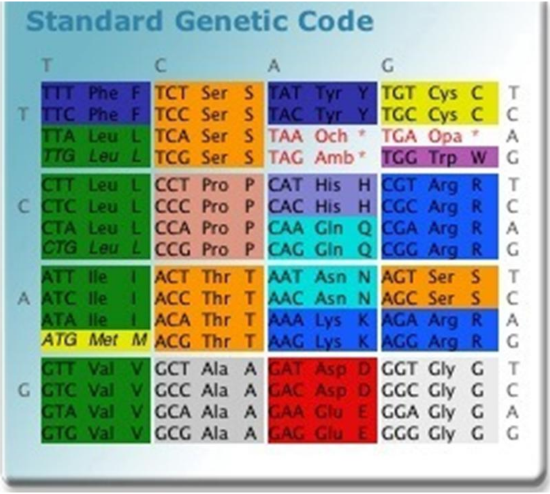
Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

Amino Acid and Codon Substitution Models

1. Dayhoff, Schwartz and Orcutt, 1978. A Model of Evolutionary Change in Proteins.
2. Goldman N, Whelan S. A novel use of equilibrium frequencies in models of sequence evolution. Mol Biol Evol 2002 Nov;19(11):1821-31.
3. Soyer O, Dimmic MW, Neubig RR, Goldstein RA. Using evolutionary methods to study G-protein coupled receptors. Pac Symp Biocomput 2002;:625-36. See also Koshi JM, Goldstein RA. Analyzing site heterogeneity during protein evolution. Pac Symp Biocomput 2001;:191-202.

Genetic code



Standard Genetic Code

	T	C	A	G	
T	TTT Phe F	TCT Ser S	TAT Tyr Y	TGT Cys C	T
	TTC Phe F	TCC Ser S	TAC Tyr Y	TGC Cys C	C
	TTA Leu L	TCA Ser S	TAA Och *	TGA Opa *	A
	TTG Leu L	TCG Ser S	TAG Amb *	TGG Trp W	G
C	CTT Leu L	CCT Pro P	CAT His H	CGT Arg R	T
	CTC Leu L	CCC Pro P	CAC His H	CGC Arg R	C
	CTA Leu L	CCA Pro P	CAA Gln Q	CGA Arg R	A
	CTG Leu L	CCG Pro P	CAG Gln Q	CGG Arg R	G
A	ATT Ile I	ACT Thr T	AAT Asn N	AGT Ser S	T
	ATC Ile I	ACC Thr T	AAC Asn N	AGC Ser S	C
	ATA Ile I	ACA Thr T	AAA Lys K	AGA Arg R	A
	ATG Met M	ACG Thr T	AAG Lys K	AGG Arg R	G
G	GTT Val V	GCT Ala A	GAT Asp D	GGT Gly G	T
	GTC Val V	GCC Ala A	GAC Asp D	GGC Gly G	C
	GTA Val V	GCA Ala A	GAA Glu E	GGA Gly G	A
	GTG Val V	GCG Ala A	GAG Glu E	GGG Gly G	G

Yang & Nielsen, 2000

$$R_{ij} = \begin{cases} 0 & \text{more than one difference} \\ \pi_j & \text{synonymous transversion} \\ \kappa\pi_j & \text{synonymous transition} \\ \omega\pi_j & \text{nonsynonymous transversion} \\ \omega\kappa\pi_j & \text{nonsynonymous transition} \end{cases}$$

Here i, j are codons.

Outline

1. Background
2. Markov Chain Theory
   - Basic Definitions
   - Matrix Exponentiation
   - Simulation
3. Substitution Models
   - DNA/RNA
   - Proteins
4. Mathematical Appendices
   - Linear Algebra
   - Probability

Symmetric Matrices have Real Eigenvalues

- Real-valued symmetric matrices have real eigenvalues
  - Proof: conjugate of $\mathbf{Au} = \lambda \mathbf{u}$ is $\mathbf{A}\bar{\mathbf{u}} = \bar{\lambda}\bar{\mathbf{u}}$
    - In general, $\overline{xy} = \bar{x}\bar{y}$, and $\mathbf{A}$ is real, so $\bar{\mathbf{A}} = \mathbf{A}$
  - Thus, $\mathbf{u}^T\mathbf{A}\bar{\mathbf{u}} = \bar{\lambda}\mathbf{u}^T\bar{\mathbf{u}} = \bar{\lambda}\bar{\mathbf{u}}^T\mathbf{u}$
  - Also, clearly, $\bar{\mathbf{u}}^T\mathbf{Au} = \lambda\bar{\mathbf{u}}^T\mathbf{u}$
  - Subtracting first from second gives $\bar{\mathbf{u}}^T\mathbf{Au} - \mathbf{u}^T\mathbf{A}\bar{\mathbf{u}} = (\lambda - \bar{\lambda})\bar{\mathbf{u}}^T\mathbf{u}$
  - Since $\mathbf{A}$ is symmetric, $\bar{\mathbf{u}}^T\mathbf{Au} = \mathbf{u}^T\mathbf{A}\bar{\mathbf{u}}$
  - Also, $\mathbf{u}^T\bar{\mathbf{u}} > 0$ unless $\mathbf{u} = 0$, because $\bar{x}x \geq 0$ unless $x = 0$.
  - Thus $\lambda - \bar{\lambda} = 0$.

Symmetric Matrices have Orthogonal Eigenvectors

- Eigenvectors can be chosen to be orthonormal
    - Proof that $\mathbf{u}_i$ and $\mathbf{u}_j$ are orthogonal if $\lambda_i \neq \lambda_j$:
    - Left-multiply $\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i$ by $\mathbf{u}_j^T$ to obtain $\mathbf{u}_j^T \mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_j^T \mathbf{u}_i$
    - Likewise $\mathbf{u}_i^T \mathbf{A}\mathbf{u}_j = \lambda_j \mathbf{u}_i^T \mathbf{u}_j$
    - Subtracting gives $\mathbf{u}_j^T \mathbf{A}\mathbf{u}_i - \mathbf{u}_i^T \mathbf{A}\mathbf{u}_j = (\lambda_i - \lambda_j)\mathbf{u}_i^T \mathbf{u}_j$
    - LHS is zero by symmetry of $\mathbf{A}$. Hence $\mathbf{u}_i^T \mathbf{u}_j = 0$.
    - Proof that one can find *m* orthonormal eigenvectors for an eigenvalue repeated *m* times is more involved
        - See e.g. `www.quandt.com/papers/basicmatrixtheorems.pdf`
        - Mirrored at `http://biowiki.org/BioE241`

Real Matrices' Eigenvalues are in Conjugate Pairs

- Eigenvalues of real-valued square matrices are either real, or occur in complex conjugate pairs
  - Sketch of proof: complex conjugativity is distributive over multiplication and addition, therefore if λ is a zero of the characteristic equation then $\bar{\lambda}$ must also be a zero.
  - If λ is an eigenvalue then $|\mathbf{R} - \lambda \mathbf{I}| = f(\lambda) = \sum_{n=0}^{N} a_n \lambda^n = 0$ where the a_n are all real
  - Therefore $f(\bar{\lambda}) = \sum_{n=0}^{N} a_n (\bar{\lambda})^n = \sum_{n=0}^{N} a_n \bar{\lambda}^n = f(\bar{\lambda}) = 0$, so $\bar{\lambda}$ is also an eigenvalue.

Common Matrix Algorithms and Decompositions

- QR decomposition: any real matrix $A = QR$ where Q is orthogonal ($Q^T Q = I$) and R is upper-triangular.
- Various algorithms used to find QR and other decompositions:
  - Gram-Schmidt process
  - Householder transformation
  - Givens rotations
- Jacobi algorithm: iterative method for computing eigensystems
- Most of these algorithms subject to instabilities of one form or another, especially for asymmetric matrices (also sparse & extreme matrices)

Outline

1. Background
2. Markov Chain Theory
    - Basic Definitions
    - Matrix Exponentiation
    - Simulation
3. Substitution Models
    - DNA/RNA
    - Proteins
4. Mathematical Appendices
    - Linear Algebra
    - Probability

Change of Variables

- If $p(x)$ is pdf of x, and $y = f(x)$, what is pdf $q(y)$ of y?
- Derivation: $dy = f'(x)dx$ and $p(x)dx = q(y)dy$
- Thus $q(y) = p(x)/f'(x)$ where $x = f^{-1}(y)$

$$q(y) = \frac{p(f^{-1}(y))}{f'(f^{-1}(y))}$$

- Application: transforming a uniform r.v. to sample from a distribution of choice. Suppose that $p(x) = 1$ for $0 \leq x \leq 1$ (and 0 elsewhere), $q(y)$ is given, and we seek $f(x)$.

Sampling from a distribution

- Application of change of variables: transforming a uniform r.v. to sample from a distribution of choice.
- Suppose that $p(x) = 1$ for $0 \leq x \leq 1$ (and 0 elsewhere), $q(y)$ is given, and we seek $f(x)$.
- Again, start with $dy = f'(x)dx$ and $p(x)dx = q(y)dy$
- Since $p(x)$ is constant, we have $\frac{dx}{dy} = q(y)$ so $x = f^{-1}(y) = C(y) = \int_{-\infty}^{y} q(y')dy'$
- Thus $f(x) \equiv C^{-1}(x)$ where $C(y)$ is the cumulative distribution of y

Summary

- Discrete-state Markov chains can model evolution of many biological characters, including nucleotides and amino acids.
- The infinitesimal and finite-time transitions are represented using rate and probability matrices.
- Matrix exponentiation relates the infinitesimal and finite-time transition matrices.

- Outlook
  - Diagonalizing irreversible matrices is difficult/unstable.
  - Matrices of size > 5 cannot in general be diagonalized algebraically.