Simulating Genome of *Dictyostelium discoideum* Using k^{th}-Order Markov Chain Genetic Models

Kairavi Chahal, Tony Yang, Julian Zhou

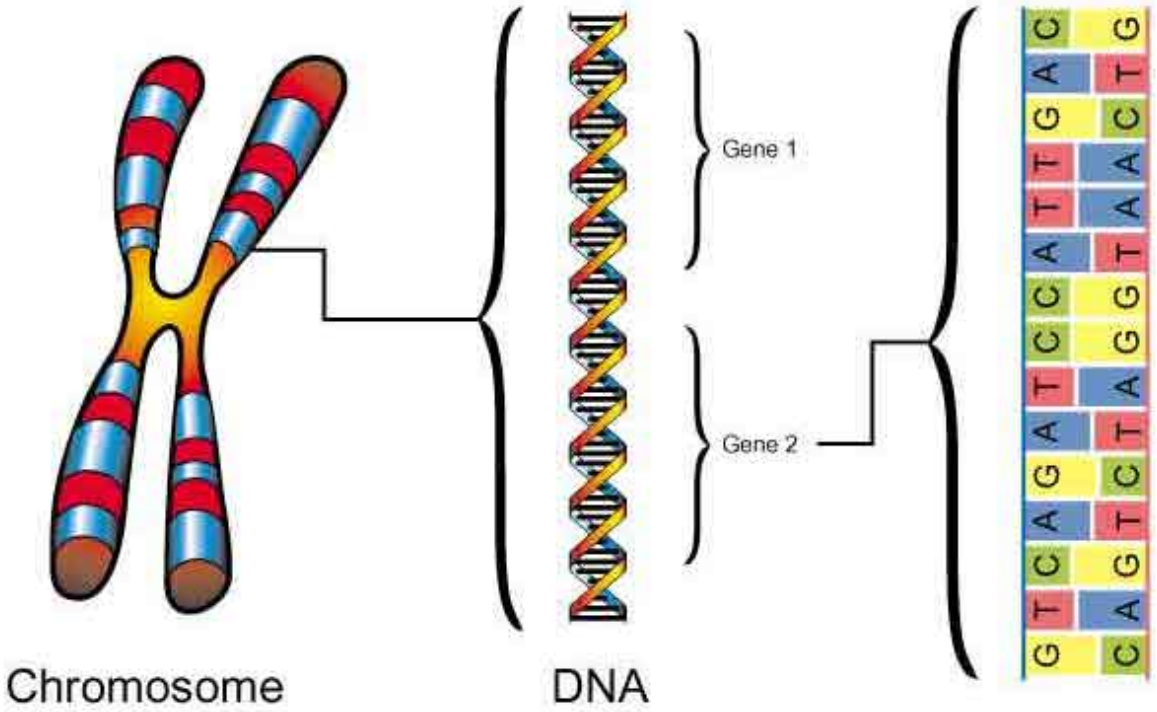
Department of Statistics
Carnegie Mellon University

December 8, 2013

Chromosomes, DNA sequences, nucleotide bases

- Soil-living amoeba, aka 'slime mold'
- Interested in simulating sequences of *ATGC*

Source: Alex Wild, Scientific American



Source: Plant & Soil Sciences eLibrary

k^{th}-th Order Markov Chain

- A single-stranded DNA sequence: $B_1,\ B_2,\ ...,\ B_i$; $b = \{A,\ T,\ G,\ C\}$
- $P(B_i = b|B_{i-1},\ ...,\ B_1) = P(B_i = b|B_{i-1},\ ...,\ B_{i-k})$
- i.e. $P(B_i = b)$ is *conditionally independent* of $\{B_{i-k-1},\ ...,\ B_1\}$, given $\{B_i,\ ...,\ B_{i-k}\}$

k^{th}-th Order Transition Matrix

- Row: $(i-k)^{th}$ through $(i-t)^{th}$ bases - 'prior sequence'
- Column: i^{th} base
- Dimension: 4^kx4
- Each row sums up to 1
- E.g. 2^{nd}-order transition matrix based on original sequence in Chromosome 2

	A	C	G	T
AA	0.505	0.093	0.074	0.328
AC	0.420	0.245	0.054	0.282
AG	0.424	0.107	0.131	0.337
AT	0.310	0.127	0.119	0.445
CA	0.468	0.138	0.093	0.300
CC	0.612	0.116	0.045	0.227
CG	0.436	0.106	0.131	0.327
CT	0.277	0.152	0.151	0.419
GA	0.420	0.071	0.114	0.394
GC	0.462	0.142	0.065	0.332
GG	0.307	0.078	0.114	0.501
GT	0.290	0.080	0.191	0.439
TA	0.436	0.103	0.082	0.380
TC	0.484	0.135	0.066	0.316
TG	0.387	0.091	0.219	0.303
TT	0.262	0.100	0.135	0.503

Reading in Data


```
>DDB0169550 |Chromosomal Sequence| Chromosome:  M position 1 to 55564
AATGAAATAAAAAAAAACGAAAATAAAAAAAAATAATGACAATAATAGCAATAAGTATAA
TGAATGTAGTGATAGGGATAGCAATATTAGGAGTAATATTAAGAAAGAAAATAATGCCGA
ACCAAAAATTTCAAAGAATATTTATATTAGGAGTACAAGGAATACTAATAGTATTAAGTG
```

- Observe that the data is in `FASTA` format
- Function `read.fasta` in package `seqinr` reads such format

Summary of Data

Chromosome	Length (*bases*)
1	4,923,596
2	8,484,197
2F	161,967
3	6,357,299
3F	16,660
4	5,450,249
5	5,125,352
6	3,602,379
BF	75,732
M	55,564
R	85,150

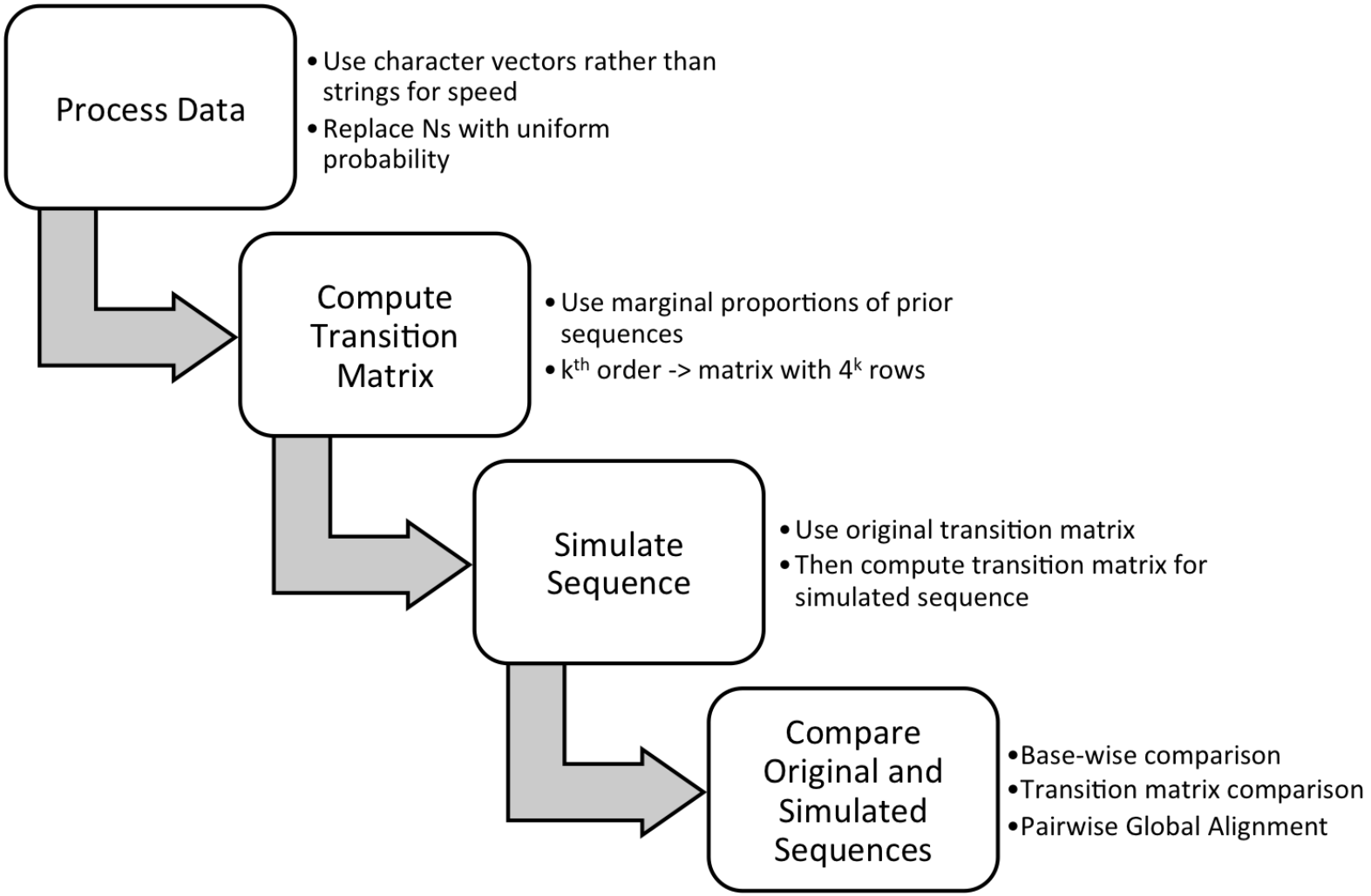
Questions of Interest

- How do simulation results change as k changes?
- Do simulation results differ from one chromosome to another?

Why not just concatenate the chromosomes?

- Doing so assumes that the starting sequence of one chromosome depends on the ending sequence of another
- Doing so assumes that transition matrices will be similar for each chromosome
- Markov matrices for the whole genome and individual chromosomes may be different
- Order in which to join the chromosomes is unknown
- Logistically, computing transition matrix and simulate the sequence for the entire genome would be overly time-consuming

Overview of Our Code



Replacing Ns

- **Idea:** replace the Ns with nucleotides using a uniform distribution
- **Alternative Idea:** calculating a matrix of marginal probabilities based on the rest of the sequence, and then applying it to the sequence of Ns as a 0^{th} order
  - **Problem:** The marginal prob. matrix is inconsistent with the overall transition matrix
- **Alternative Idea:** Drop all Ns
  - **Problem:** Will affect the transition matrix and results

Computing Transition Matrices

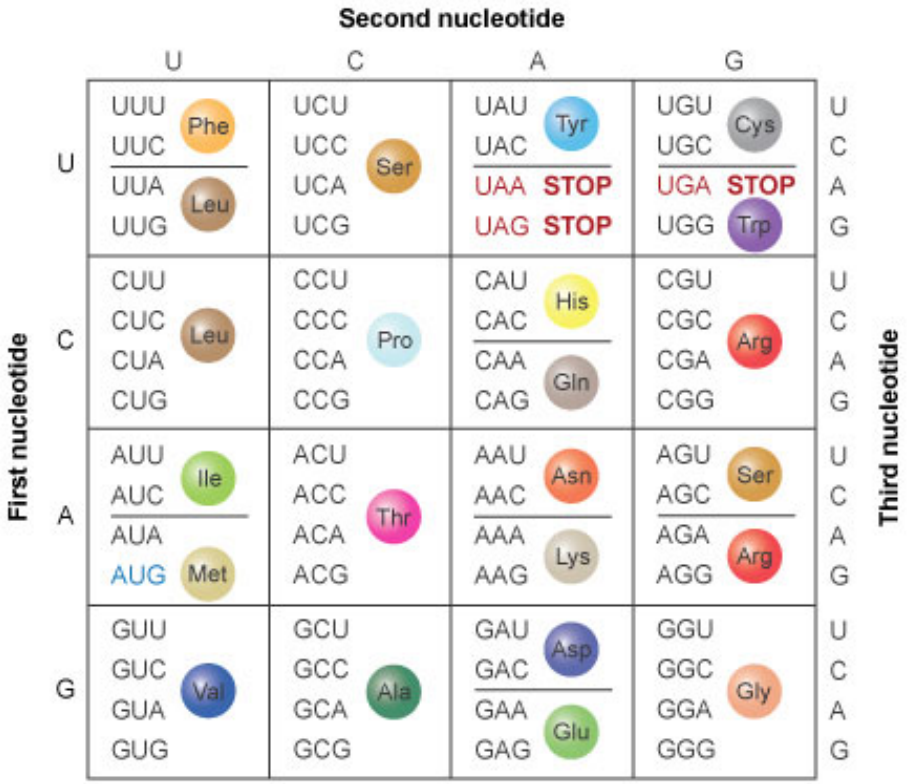
- Choose order k, and obtain all consecutive substrings of length k+1
- Group substrings by which nucleotide is in last (k+1th) position
- Count occurences of sequence of nucleotides in 1st to kth position within the four groups
- Divide each row by the sum of the row (ensures each row's probability is 1)
- If a row has all zeroes, then replace with all 0.25, since each combination is equally (un)likely

Simulations

- Idea:
  - Take the first k nucleotide bases as a starting point
  - Use the transition matrix to simulate the next base
  - Take the last k nucleotide bases in the current simulated sequence, use transition matrix, repeat
- 10 simulations per combination of chromosome & order k (ranging from 1 to 3)

Why not convert nucleotides into codons?

- Codons are degenerate
- Without knowledge of where the *coding* region begins, simply treating the first 3 bases as the starting codon could result in *frameshift mutation*



Source: Nature

Base-wise Comparision

- Number of exact matches
- Proportions of A, T, G, C (marginal probability distribution)

Comparing Transition Matrices

- $\mathbb{M}_{orig}$: transition matrix based on the original sequence
- $\mathbb{M}_{sim}$: transition matrix based on the osimulated sequence
- $\theta = \frac{\sum_{i=1}^{4^k} \sum_{j=1}^{4} |\mathbb{M}_{orig_{ij}} - \mathbb{M}_{sim_{ij}}|}{4^{k+1}}$

Pairwise Global Alignment

- *Needleman-Wunsch* algorithm (dynamic programming)
- Implemented by `pairwiseAlignment` in package `Biostrings` of Bioconductor
- Yields an optimal pairwise alignment score, based on a given scoring scheme

Pairwise Global Alignment - Scoring Scheme

- General guidelines
  - If looking for closely related sequences, penalize mismatches/ gaps a lot
  - Heavier penalty for a gap opening; smaller penalty for subsequent gap extensions
- match$= +2$, mismatch-2, gap opening$= -5$, gap extension$= -2$

17

Possible Improvements to Algorithm/Code

- Represent bases with integers (0, 1, 2, 3) instead of characters ('a', 'c', 'g', 't'), since R processes them faster
- Use BLAST for comparison, which is faster and has a standardized result. But possibility of human-error when copying/pasting.
- Previously assumed transition matrices remain constant throughout. Could take into account of evolution of transition matrices, mutation rates, etc.

threads of thought (temporary

-
- split/apply/combine - analyzing 11 chromosomes instead of 1 genome

Acknowledgment

- http://a-little-book-of-r-for-bioinformatics.readthedocs.org/en/latest/src/chapter1.html
- http://a-little-book-of-r-for-bioinformatics.readthedocs.org/en/latest/src/chapter4.html
- http://cran.r-project.org/web/packages/seqinr/seqinr.pdf
- http://tata-box-blog.blogspot.com/2012/04/introduction-to-markov-chains-and.html