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

Kairavi Chahal, Tony Yang, Julian Zhou

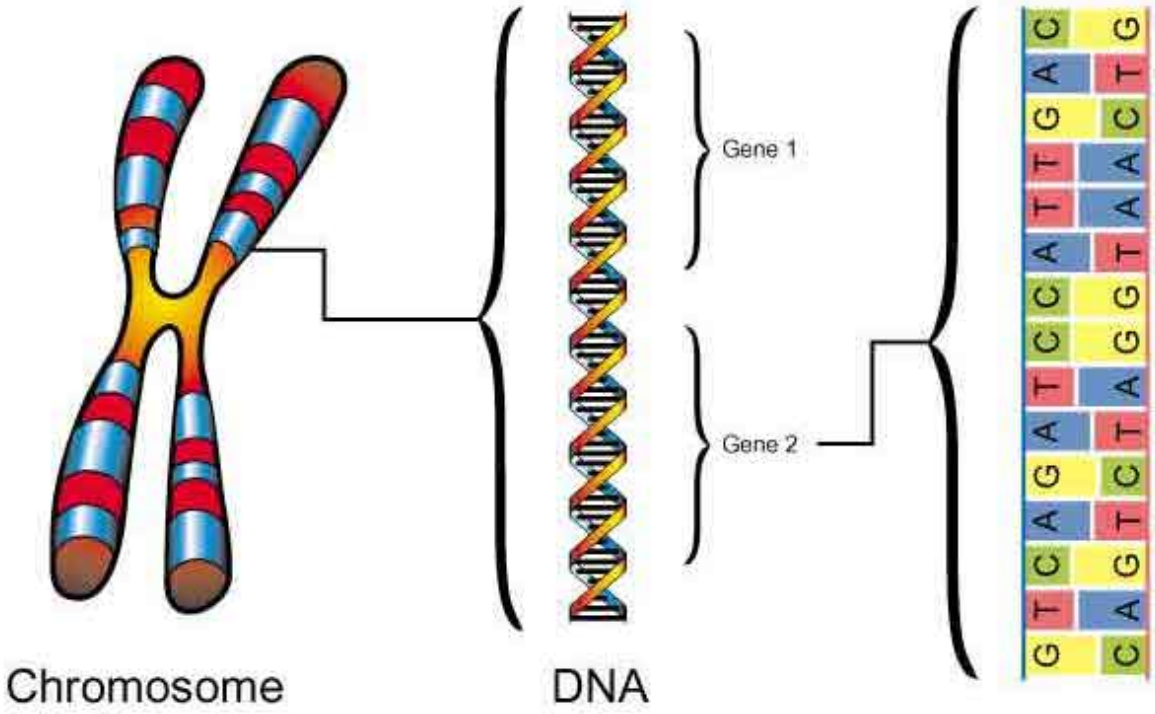
Department of Statistics
Carnegie Mellon University

December 9, 2013

Chromosomes, DNA sequences, nucleotide bases

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

Source: Alex Wild, Scientific American



Source: Plant & Soil Sciences eLibrary

k^{th}-th Order Markov Chain

- A single-stranded DNA sequence: $B_1,\ B_2,\ ...,\ B_n$; $b = \{A,\ C,\ G,\ T\}$
- $P(B_i = b | B_{i-1},\ ...,\ B_1) = P(B_i = b | B_{i-1},\ ...,\ B_{i-k})$
- 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 $(i-1)^{th}$ bases - 'prior sequence'
- Column: i^{th} base
- Dimension: 4^kx4
- Each row sums up to 1
- E.g. 2nd-order transition matrix based on original sequence in Chromosome 2

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

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

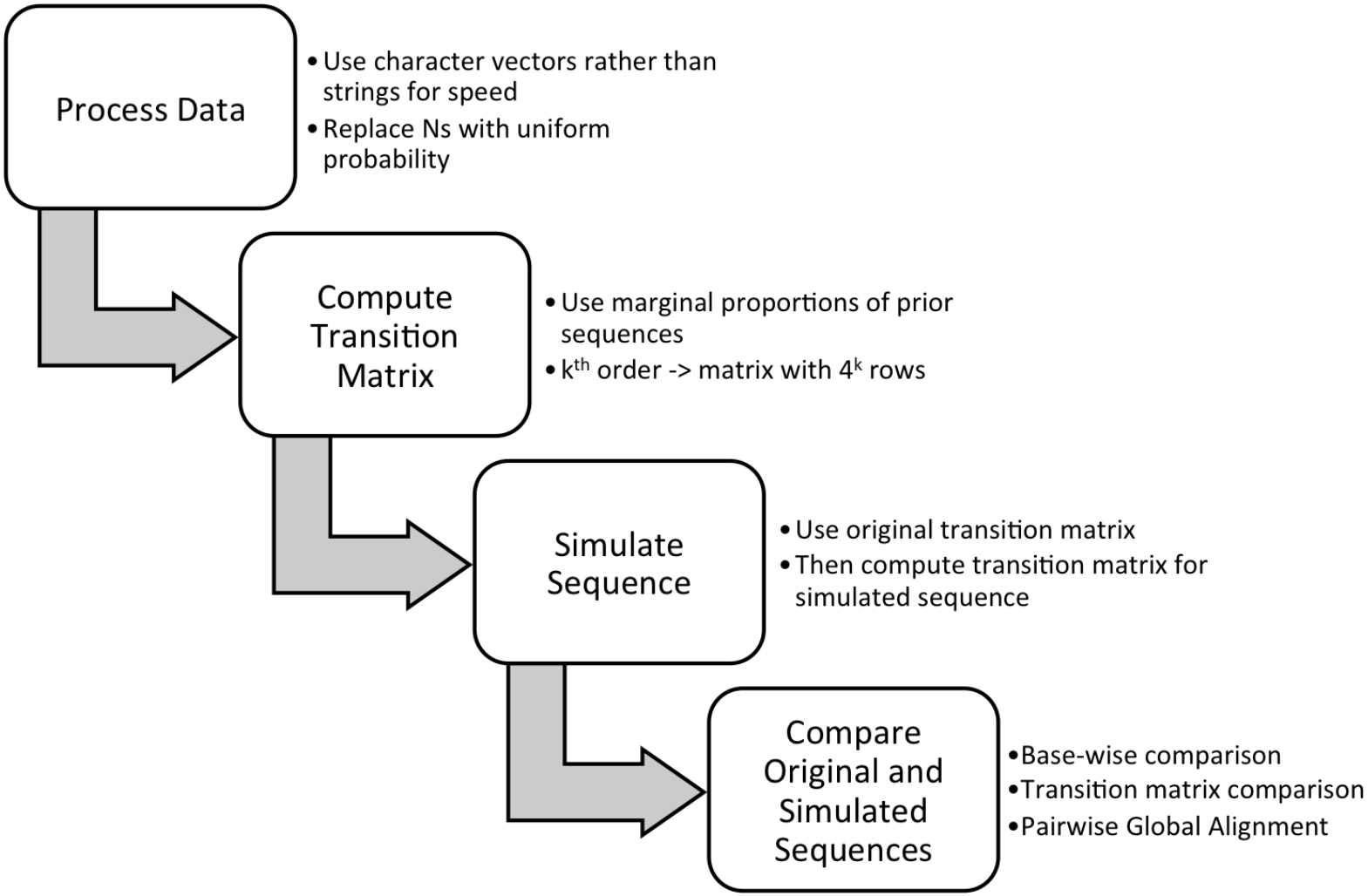
Questions of Interest

- 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

Approach

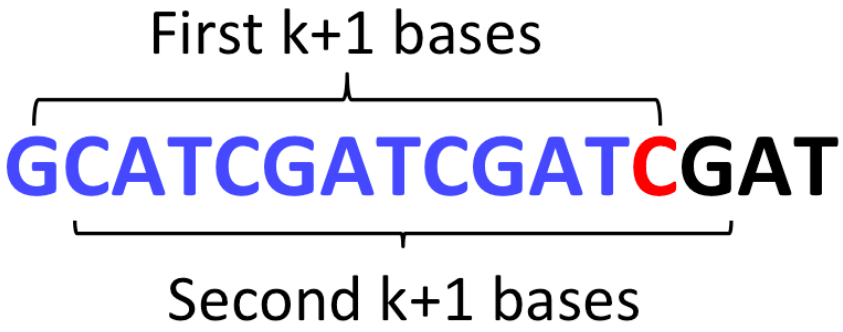


Replacing Ns

- Missing data: Ns in place of some bases which are not known
- **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 0th 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

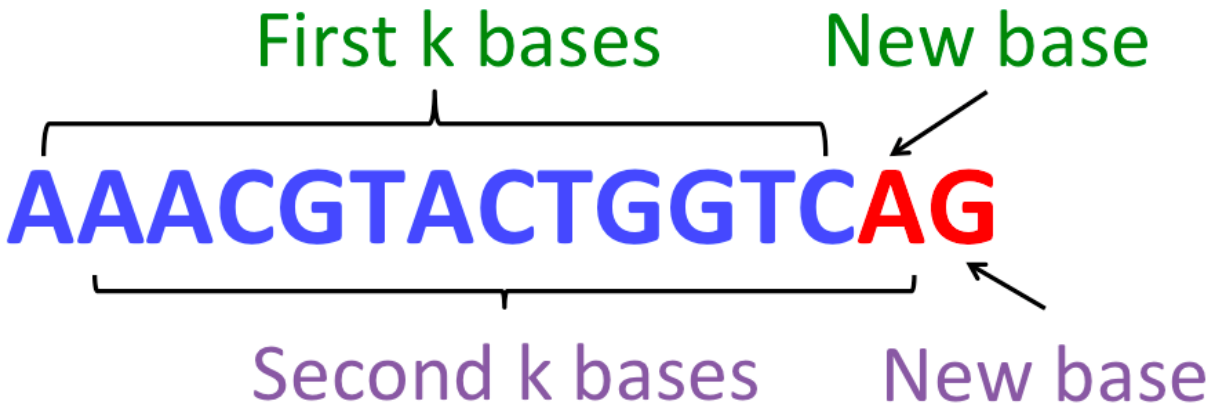
Computing Transition Matrices

- Choose order k, and obtain all consecutive subsequences of length k+1
- Group subsequences by which nucleotide is in last (k+1th) position
- Count occurrences 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

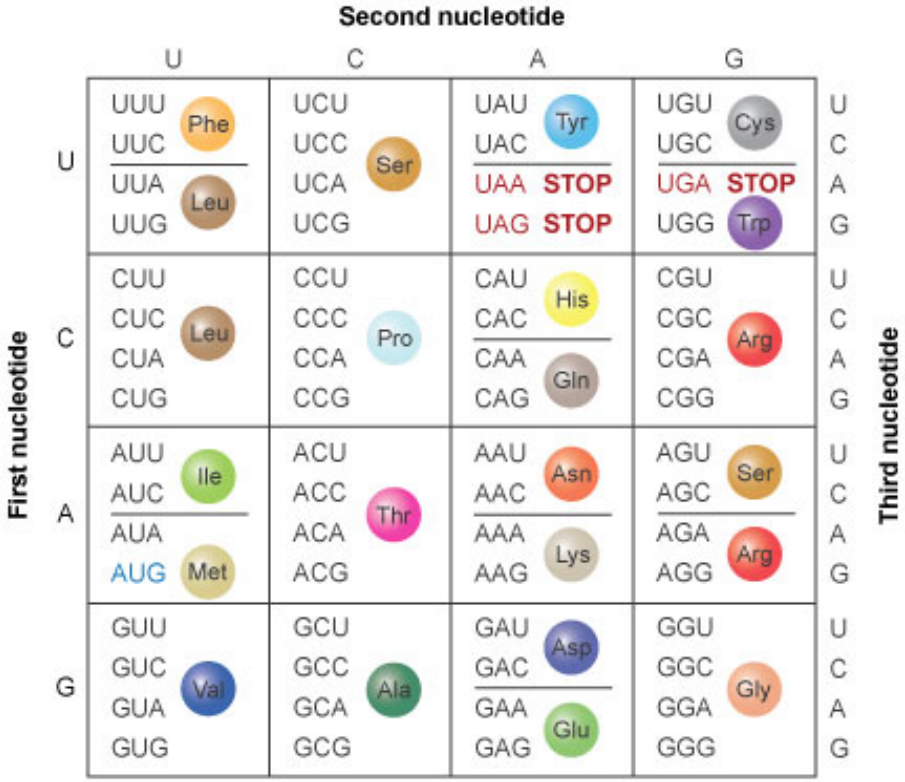
- Take the first k nucleotide bases from the original sequence as a starting point
- Use the transition matrix based on the original sequence ($\mathbb{M}_{orig}$) to simulate the next base
- Take the last k nucleotide bases in the current simulated sequence to generate the next base, using $\mathbb{M}_{orig}$; repeat



- 1st, 2nd and 3rd order Markov chains for 11 chromosomes, each simulated 10 times $\Rightarrow$ 330 simulations.

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 Comparison

- Proportion 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 simulated sequence
- $\hat{\theta} = \dfrac{\sum_{i=1}^{4^k} \sum_{j=1}^{4} |\mathbb{M}_{orig_{ij}} - \mathbb{M}_{sim_{ij}}|}{4^{k+1}}$
- Standardized by the number of entries in the matrix

Pairwise Global Alignment

- As opposed to semi-global or local 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

$$Align(A[i], B[j]) = \max \begin{cases} Align(A[i-1], B[j-1]) + m, & A[i] = B[j] \\ Align(A[i-1], B[j-1]) + x, & A[i] \neq B[j] \\ Align(A[i-1], B[j]) + g \\ Align(A[i], B[j-1]) + g \end{cases}$$

best alignment on the first *i* characters of A and the first *j* characters of B

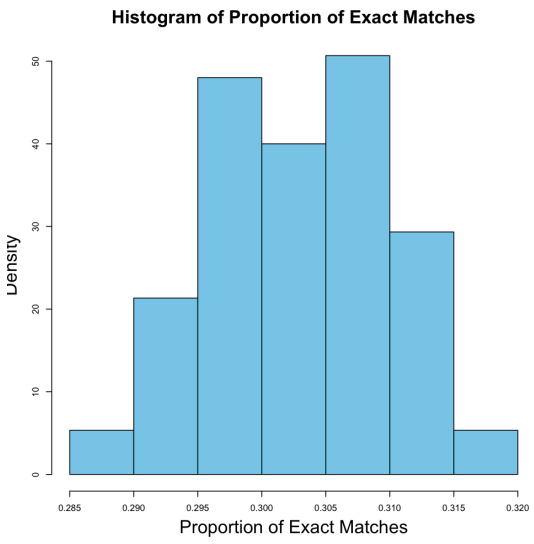
Source: Dr. R Schwartz's 02-250 slide

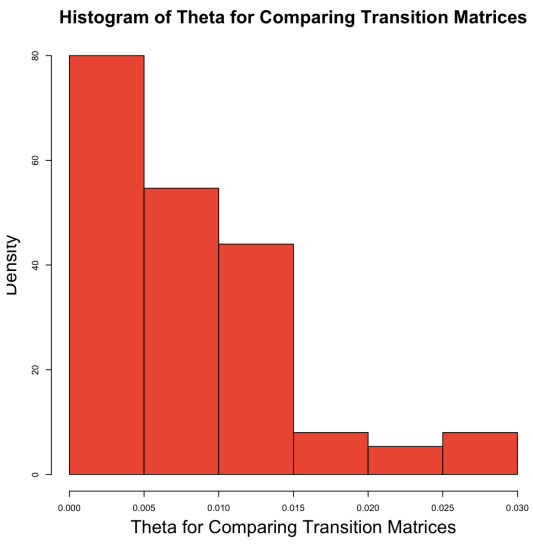
Pairwise Global Alignment - Scoring Scheme

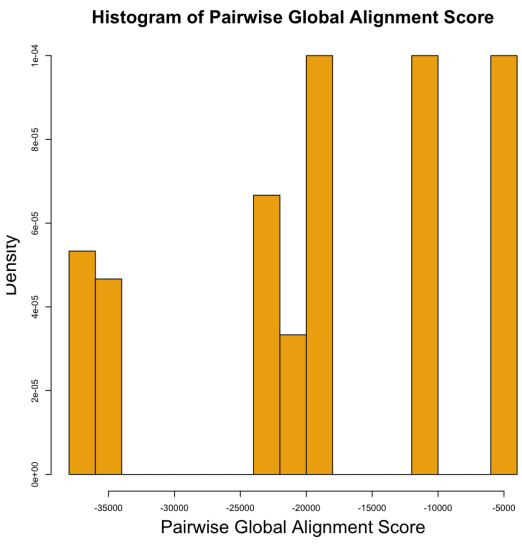
- General guidelines
  - If looking for closely related sequences, penalize mismatches/ gaps heavily
  - Heavier penalty for a gap opening; smaller penalty for subsequent gap extensions
- match$= +2$, mismatch-2, gap opening$= -5$, gap extension$= -2$

```
GAAGCTTC
GA---TTA
4m+1x+1o+3e =
4(2)+1(-2)+1(-5)+3(-2)=-5
```

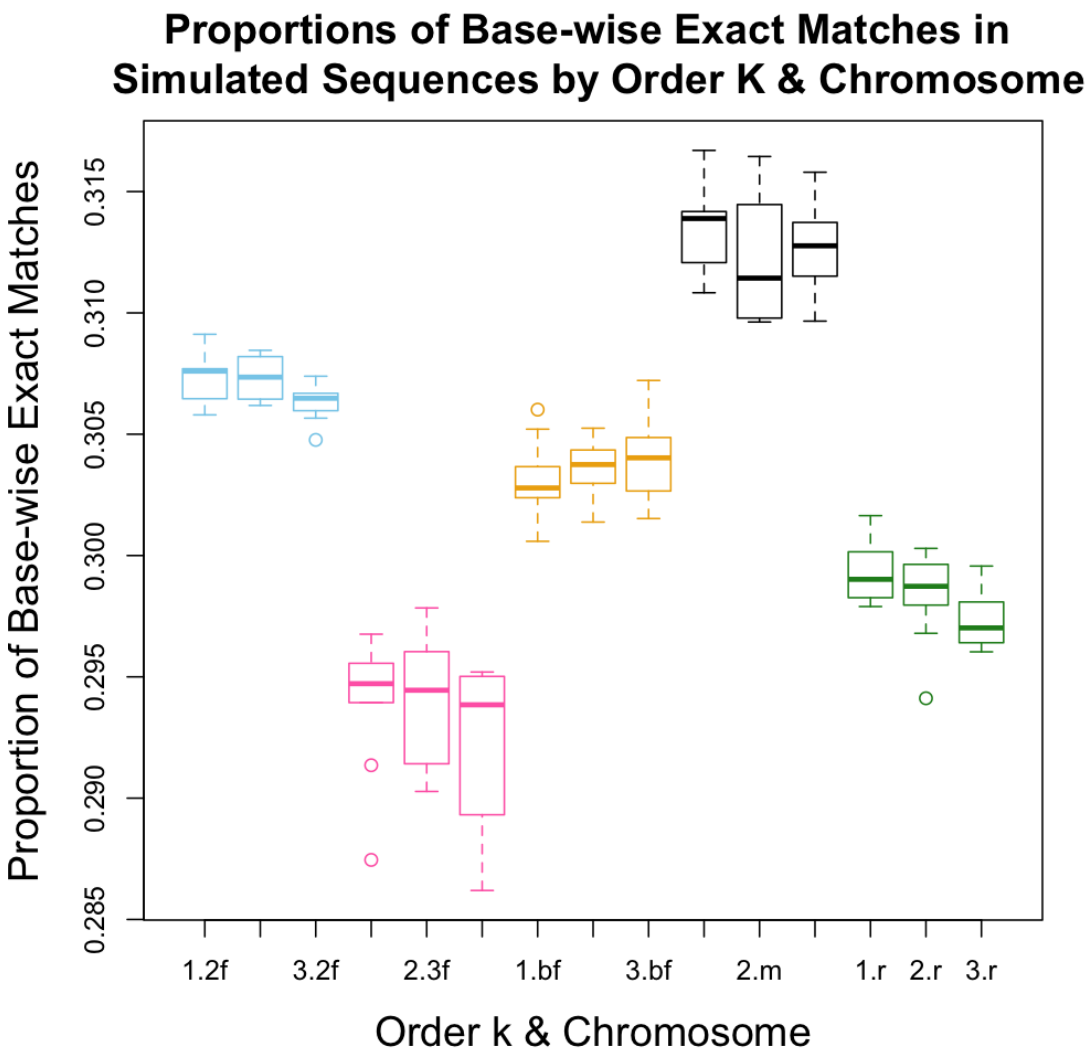
Univariate EDA



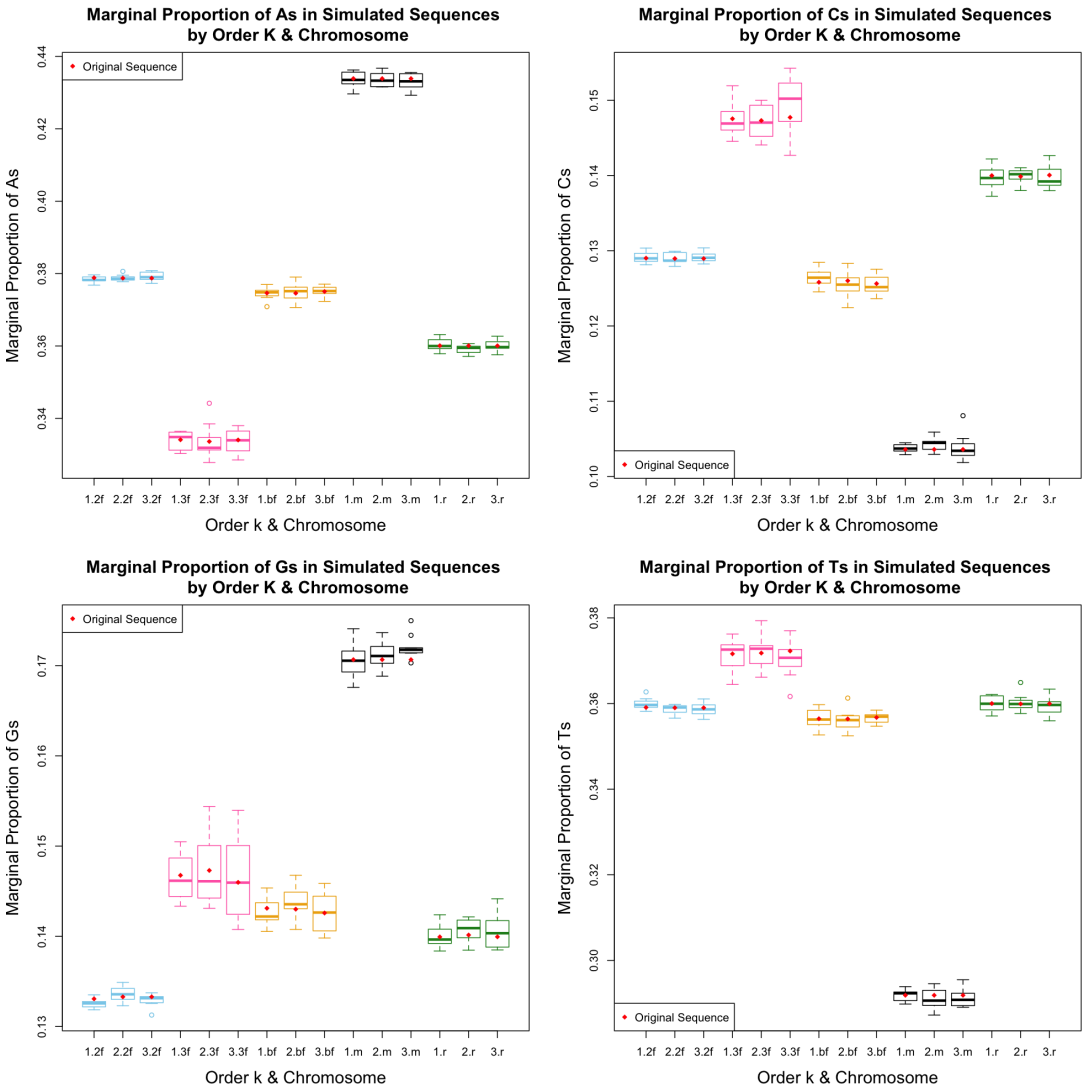




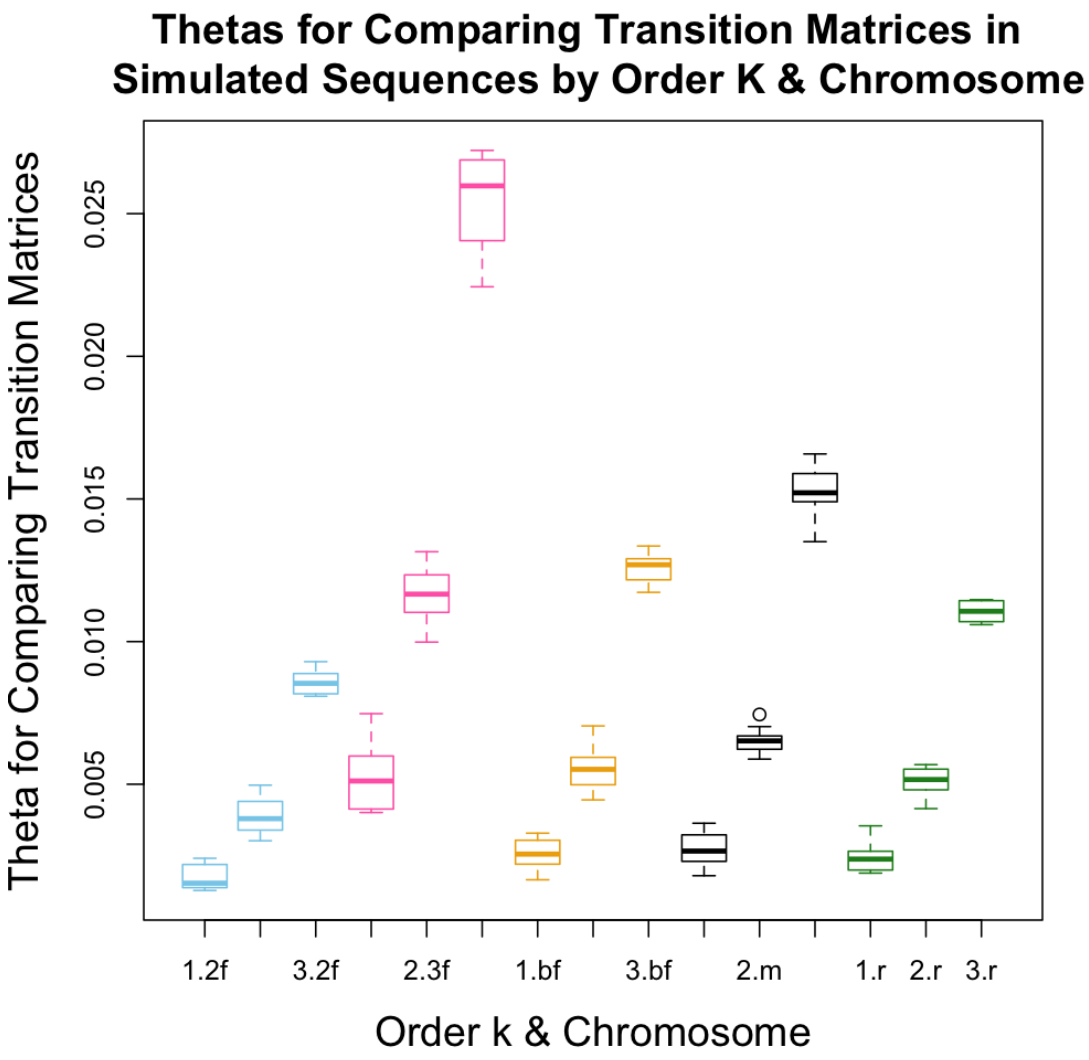
Multivariate EDA



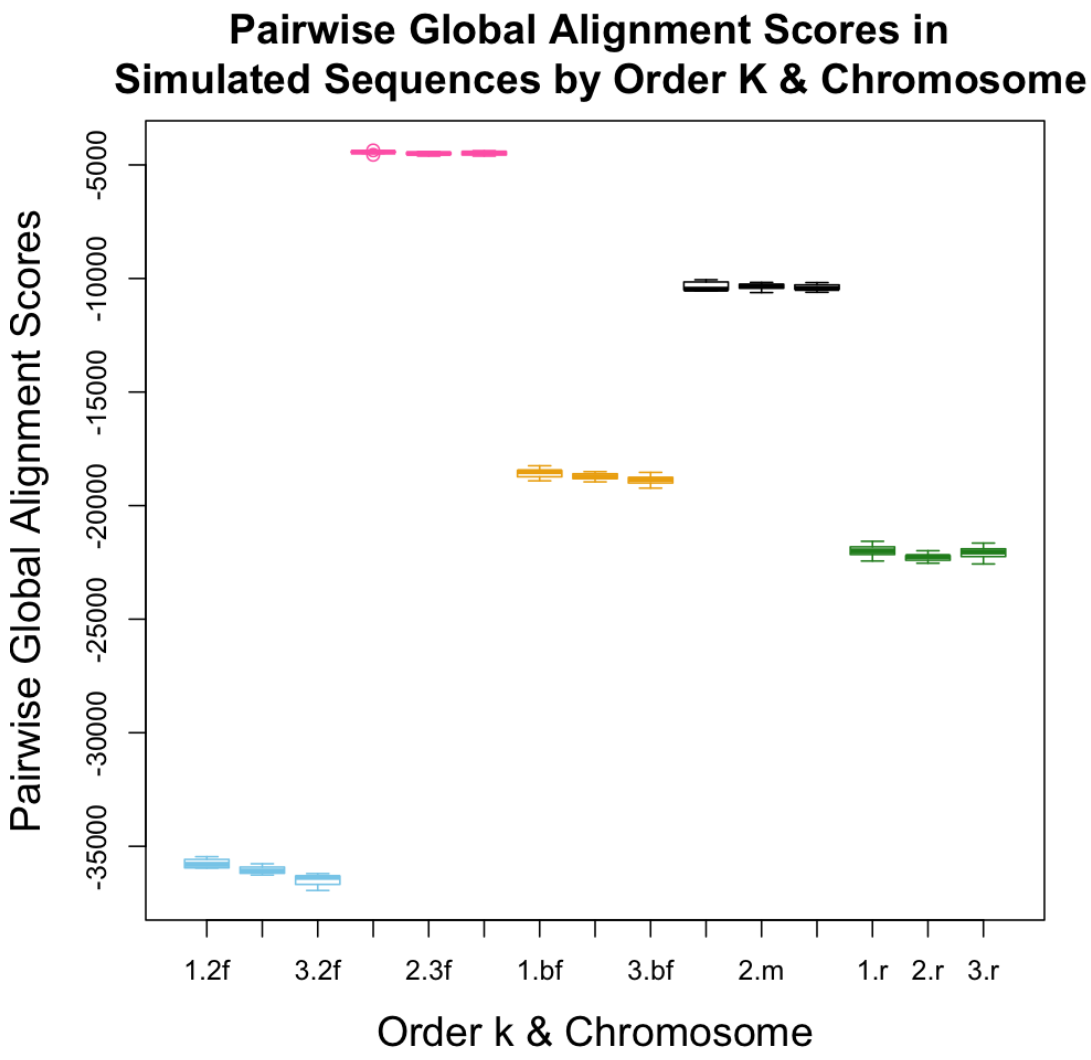
Multivariate EDA



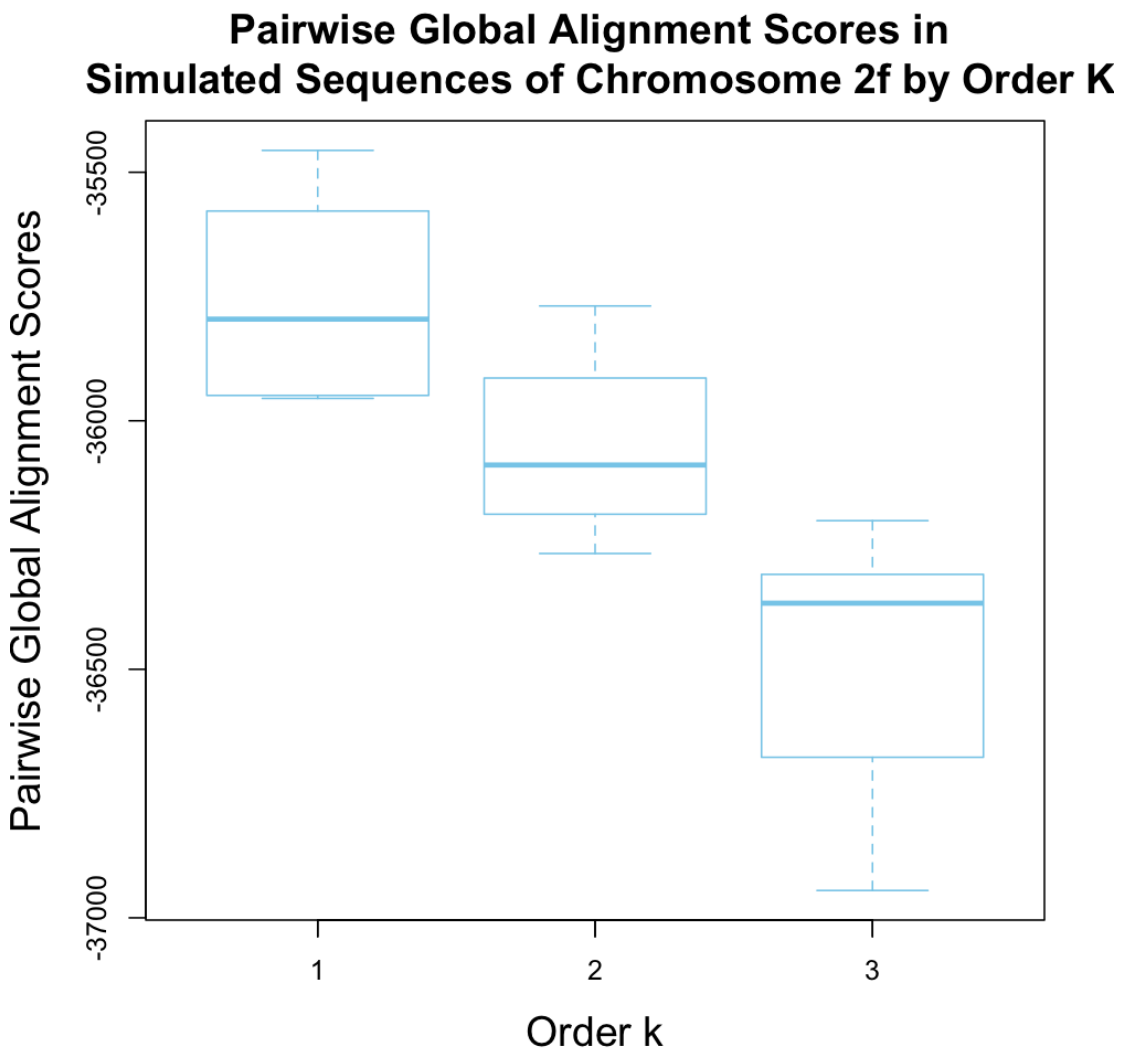
Multivariate EDA



Multivariate EDA



Zooming In



Formal Analysis

- `lm(Proportion of Base-wise Exact Matches ~ Chromosome + k)`
- `lm(Thetas for Comparing Matrices ~ Chromosome + k)`
- `lm(Thetas for Comparing Matrices ~ Total Length + k)`
- `lm(Pairwise Global Alignment Scores ~ Chromosome + k)`
- `lm(Pairwise Global Alignment Scores ~ Total Length + k)`

- Global F-tests for all models are highly significant with p-values $< 2 * 10^{-16}$
- $R^2_{adj} = .9272,\ .7844,\ .8545,\ .9736,\ ,.9996$ respectively
- *Total Length* and all categories of *Chromosome* are significant at $\alpha = .02$ in all of their respective models; many actually have p-values $< 2 * 10^{-16}$
- k appears significant at $\alpha = 0.02$ in the presence of *Total Length* or *Chromosome* in all but the 4th model (p-value=0.508)

Possible Improvements

- Represent bases with integers (1, 2, 3, 4) 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 requires manual input, which is prone to human error.
- Previously assumed that transition matrices remain constant throughout. Could take into account of evolution of transition matrices, mutation rates, etc.

References

- 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
- Lecture slides from Dr. R Schwartz's 02-250 Intro to Computational Biology