Quiz section 10

June 1, 2018
Logistics

• Bring: 1 page cheat-sheet, simple calculator
• Any last logistics questions about the final?
Logistics

• Bring: 1 page cheat-sheet, simple calculator
• Any last logistics questions about the final?
• Please do course evals for all three of the instructors (expire today!)

UW Course Evaluations via IASy:
to Hannah

Dear Professor Pliner,
Course Takeaways

• Bioinformatics is not magic: there are always assumptions, uncertainties, ambiguities

• Have an understanding of what is happening inside black boxes – it might not be as complicated as it seems initially

• Start small and be clear in your own analysis and programming
Topics from first half

- **Alignments**
  - Reasons to align sequences
  - Needleman-Wunsch algorithm
  - Smith-Waterman algorithm
  - Effects of parameter variation (including gap penalties)
  - Testing for statistical significance of an alignment

- **Phylogenetic trees**
  - Rooted and unrooted topologies
  - Defining the best tree with UPGMA and Neighbor Joining
  - Concept of parsimony
  - Fitch algorithm: quantifying how parsimonious a tree is, assigning internal states
  - Finding the most parsimonious tree: Hill climbing w/ Nearest-Neighbor interchanges
  - Bootstrapping to quantify confidence in tree partitions

- **Clustering**
  - Defining a clustering problem
  - Hierarchical clustering
    - Impact of using single/complete/average linkage
  - K-means: Objective and algorithm

- **Networks**
  - Reasons to make and analyze networks
  - Basic network definitions
  - Dijkstra’s Algorithm
  - Network motifs and their uses
Topics from second half

• Gene prediction
  • Experimental methods
  • Homology based
  • Ab initio/Ad hoc
  • Markov chain/model and property
  • Hidden markov models
  • Viterbi algorithm
  • Forward-backward algorithm (general concept only)

• Machine learning
  • Definitions (objects, features, model, training data, overfitting)
  • Supervised versus unsupervised and tasks for each
  • Variant effect prediction
  • Feature trees
  • Decision trees
  • Tree generation
  • Contingency tables/confusion matrices
  • Performance stats (accuracy, TPR, TNR)

• DNA Sequencing and alignment
  • Sequencing technologies
  • Uses of HTS
  • Short read mapping (hashing, seeding, spaced seeds, difference of BWA)
  • Paired read structural variation
  • Variant calling (Phred scores)
  • Genome assembly (N50, graph-based assembly)
Markov chain/model versus hidden Markov model?
Viterbi algorithm

• What is it for?

• Big picture of how it works?
Viterbi: determine the likeliest hidden state sequence for an observed sequence

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-rich</td>
<td>0.4</td>
<td></td>
<td>0.9*0.8= ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T-rich</td>
<td>0.1</td>
<td></td>
<td></td>
<td>0.5*0.2= 0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Viterbi: determine the likeliest hidden state sequence for an observed sequence

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-rich</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T-rich</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Observed sequence

- A-rich: 0.4 → 0.288
- T-rich: 0.1
Viterbi: determine the likeliest hidden state sequence for an observed sequence

**Observed sequence**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-rich</td>
<td>0.4</td>
<td></td>
<td>0.288</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T-rich</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Viterbi: determine the likeliest hidden state sequence for an observed sequence

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-rich</td>
<td>0.4</td>
<td>0.288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T-rich</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Observed sequence
What’s a decision tree?

• How do you construct an optimal decision tree?
What is hashing?

• A **hash function** maps some object \( x \) to an integer \( i \)

• A hash function allows us to have a hash table, which is like a list that allows indexing by arbitrary objects (a python Dictionary!)

• We can compute the value of the hash function and find the index in the hash table in constant time – fast!!

\[
\text{hash(‘hello’) } \rightarrow 3
\]

Hash table with key ‘hello’

0 1 2 3

‘hello’
Hashing Improves Search

• A **hash function** assigns a unique key to each unique data element (DNA sequence in our case)

\[
\text{hash(“ATGCTG”) } = \text{ key1 } \\
\text{hash(“TTTCTG”) } = \text{ key2 }
\]

... 

• **Keys** encode strings in a short, easily comparable format (e.g. a number)
Hashing Improves Search

- A hash function assigns a unique key to each unique data element (DNA sequence in our case).

- The hash table is an associative array that describes the relationship between the key and the sequence and its genomic location.

<table>
<thead>
<tr>
<th>Key</th>
<th>Hashed index</th>
<th>Genomic location</th>
</tr>
</thead>
<tbody>
<tr>
<td>“GCTAGC”</td>
<td>Key1</td>
<td>Chr1 123412</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>“TTTAGC”</td>
<td>KeyN</td>
<td>Chr6 988472</td>
</tr>
</tbody>
</table>
Is hashing really faster?

constructing dictionary: 0.0440001487732
Using the dictionary: 0.00799989700317
Using reference.index: 0.0120000839233

Constructing the dictionary is expensive, but you only have to do it once, and you keep reaping the benefits
What is seeding? Spaced seeding?

• With this spaced seeding, how many mismatches will be allowed? Why?
Graph-based genome assembly

• Where is the read “CAATG” represented?
Any other questions??