DNA Sequence Matching Using Dynamic Programming within a Map-Reduce Environment

Garrett Poppe

Computer Science Department
California State University, Dominguez Hills

Overview

- DNA Sequence Matching Problems
- Sequence Matching and Dynamic Programming
- Map-Reduce
- DNA Sequencing Using Map-Reduce
- Current Work
- Future Work
- References

Clustal

ClustalW/ClustalX

- "Classic Clustal"
- GUI (ClustalX), command line (ClustalW), web server versions available

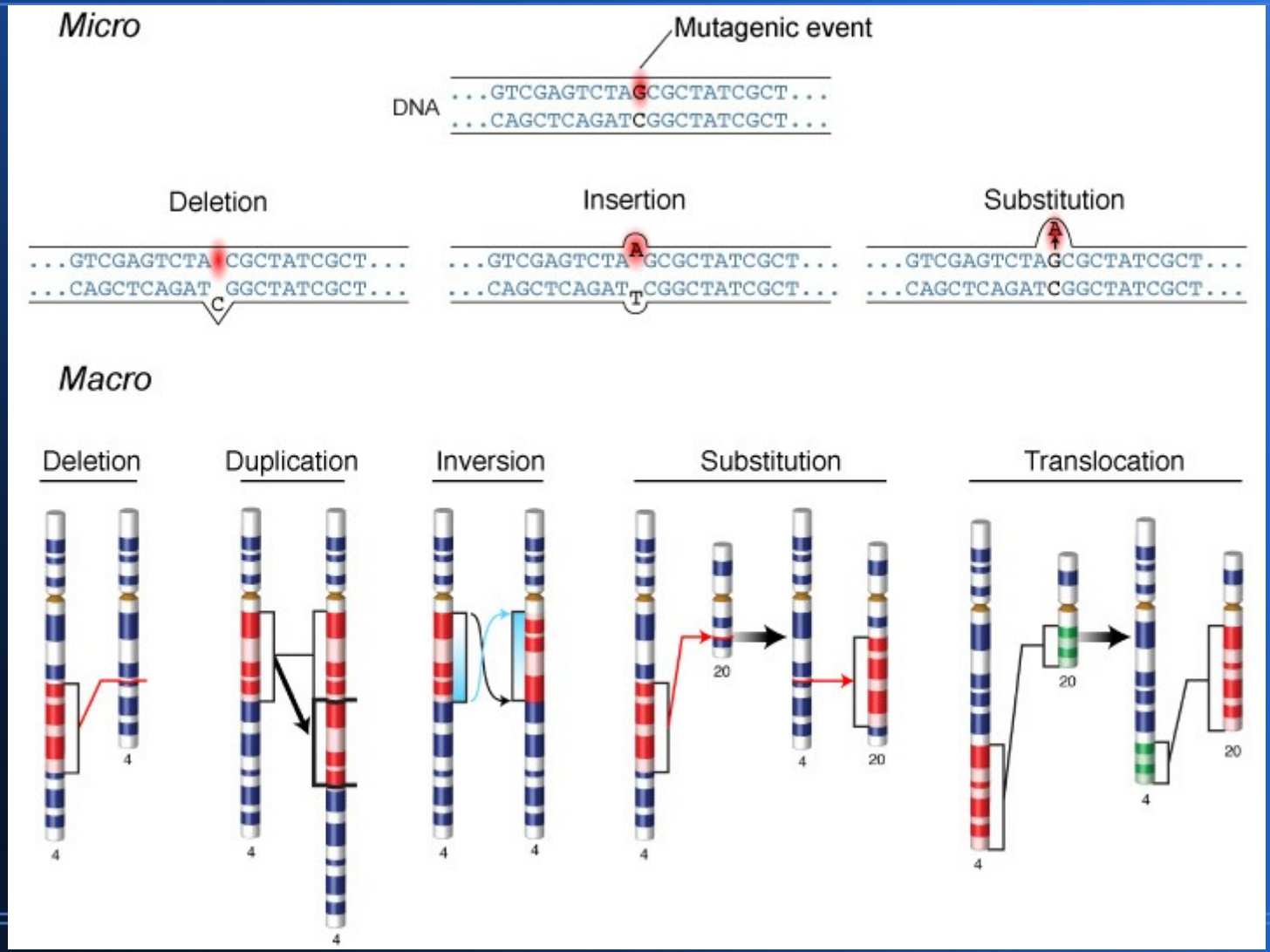
Clustal Omega

- Latest version of Clustal - fast and scalable (can align hundreds of thousands of sequences in hours), greater accuracy due to new HMM alignment engine
- Command line/web server only (GUI public beta available soon)

Problems

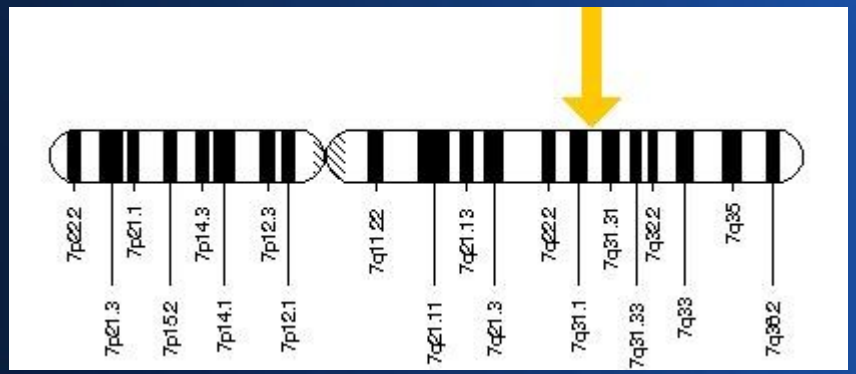
- Multiple DNA sequence matching is an NP complete problem (3 or more sequences), use heuristic methods (dynamic programming).
- Clustal can match 100 to <2000 sequences.
- The genome of C. fraxinea is 63 million bases long.
- The genome of the ash tree is even longer (approximately 954 million bases)

DNA Sequence Changes



DNA Sequence Changes

- Some changes cause genetic mutation diseases.
  - Cystic Fibrosis
  - Hemophilia
  - Tay-Sachs



DNA Sequence Changes

- Some changes cause disease resistance.
  - Chalara dieback of ash.
- Chalara a serious disease of ash trees caused by a fungus called Chalara fraxinea (C. fraxinea), including its sexual stage, Hymenoscyphus pseudoalbidus (H. pseudoalbidus). The disease causes leaf loss and crown dieback in affected trees, and is usually fatal.

Problem

- “At present, there is little scientists can do about the disease other than monitor its progress. However, laboratories are on the trail of genetic variants among the ash tree population that exhibit resistance to or tolerance of the disease. If they can isolate the genes responsible for conferring these qualities, they might be able to cross-breed UK ash trees with resistant or tolerant strains, and so save the widespread populations of ash.”

Problem

- “More than 300,000 people have played Phylo since it launched in 2010, and Waldispuhl's team reported in a 2013 Genome Biology paper that up to 50 percent of the time, a casual gamer can match the performance of expert players; and up to 40 percent of the time, Phylo players can improve on the solution found by a computer program. “

Fraxinus

- This requires vast amounts of data to be crunched by computer, identifying patterns that could show the DNA sequences most likely to be useful.
- Scientists have worked with computer games experts to develop a free app that will encourage people to compare patterns in order to highlight the genetic sequences most likely to be of use.

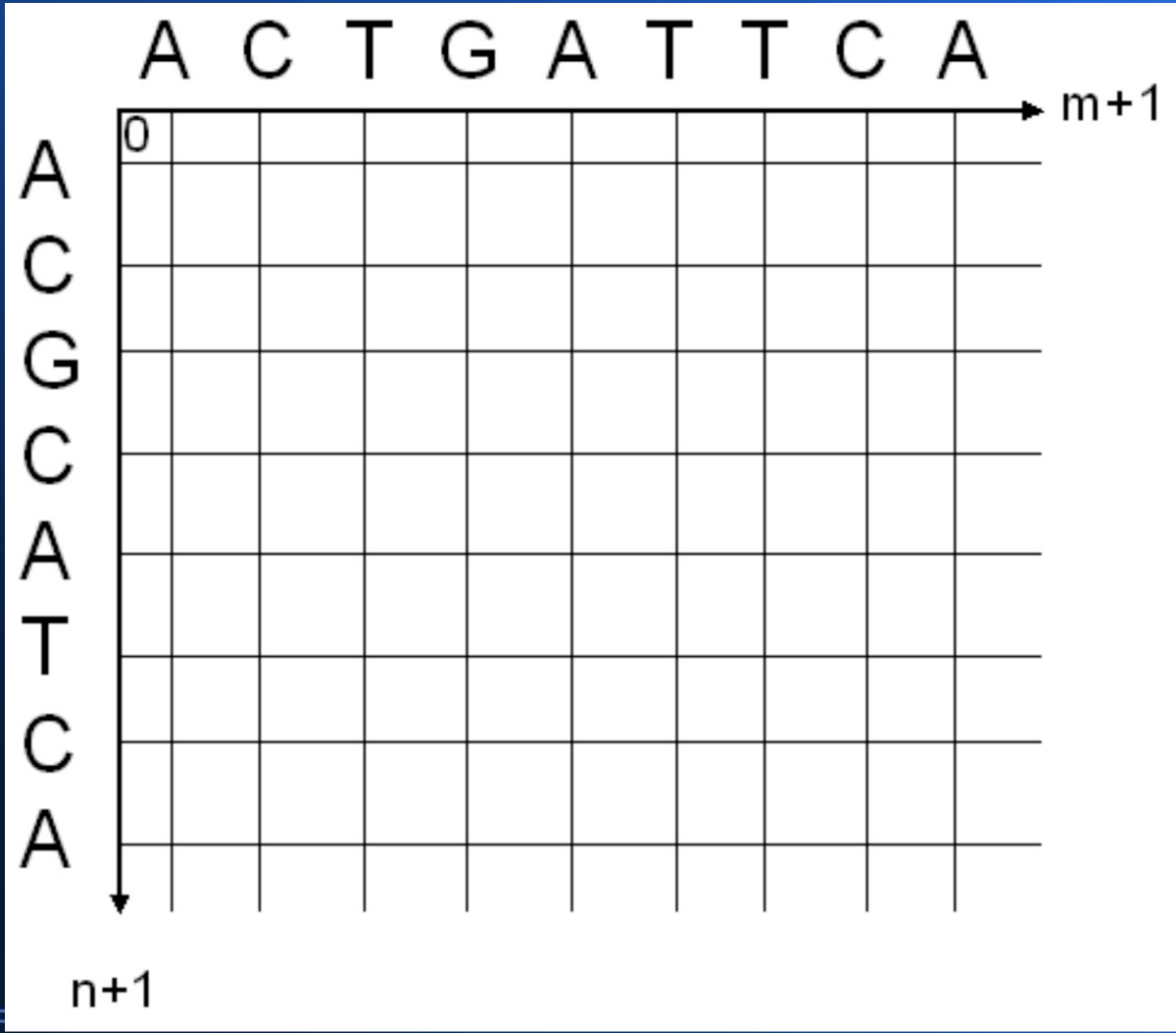
Fraxinus


FRAXINUS
Match This Pattern
Pattern value
502
Your Previous Best:
502
Current Holder:
Multiple Claimants
1881
0%
Target Score: 1881
Reset to Saved
Reset to Default
Go Back to Menu

Dynamic Programming

- Find the best alignment using a penalty and reward system.
  - Gap Penalty
  - Mismatch Penalty
  - Match Bonus
  - ? Mutation, Add, Drop ?
- Comparing 2 DNA sequences
  - ACTGATTCA
  - ACGCATCA

Sequence Matching



Sequence Matching

- Draw a Matrix for the two sequences
  - Length m and Length n
  - Dimensions (m+1) x (n+1)
- Assign scores to the Matrix based on scoring system.
  - -2 for a gap
  - -3 for a mismatch
  - +2 for a match

Sequence Matching

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Sequence Matching

- The optimal path can be traced from the bottom right to the upper left.
- This will lead to an “optimal” score.
- The best scoring sequence in this system is 8
  - ACTG--ATTCA
  - AC--GCAT--CA

Sequence Matching

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Sequence Matching

- Score is additive so you can use recursive methods to remember past values.
- Create a dictionary for sequences already matched and check new sequences against dictionary.
- Local sequences can be checked independent of global sequences!

Sequence Matching Algorithms

- Using the indices of the matrix as i and j we can enumerate using a sample pseudo code.

```
1  create array cost(m+1,n+1)
2  cost(0,0) := 0
3  for i := m+1 do
4      for j := n+1 do
5          cost(i,j) := min[ cost(i-1,j) + insertion cost target(i-1)    //add 1
6                            cost(i-1,j-1) + replace cost source(j-1)target(i-1)    //add 2
7                            cost(i,j-1) + delete cost source(j-1)        //add 1
8                          ]
```

Sequence Matching Algorithms

- Enumerating matrix
  - Viterbi or forward-backward algorithm
- Search Algorithms
  - Needleman-Wunsch
  - Smith-Waterman
  - Maximum Contiguous Sub-sequence sum
  - Affine Gap Penalty
  - Hirschberg’s algorithm

MapReduce

- Prepare the Map input
- Run the user-provided Map code
- Shuffle the Map output to the Reduce processors
- Run the user-provided Reduce code
- Produce the final output

Prepare the Map input

- Break the DNA sequence into an even number of strings and assign a key value.
- Send entire DNA string to each client to compare against.
- Send another key string to the client ensuring no duplicates.
- Client 1
  - Key 1 and key3
- Client 2
  - Key1 and key2

Run user-provided Map code

- Use dynamic programs to find optimal path for each key pair.
- Return Path and score.
- Client 1
  - Return(segment index, Score)
- Client 2
  - Return(segment index, Score)

Shuffle the Map output to Reduce

- Set a minimum score value.
- Find all keys that meet minimum score value.
- Return keys and values.
- Find all input keys within 1 key near minimum score

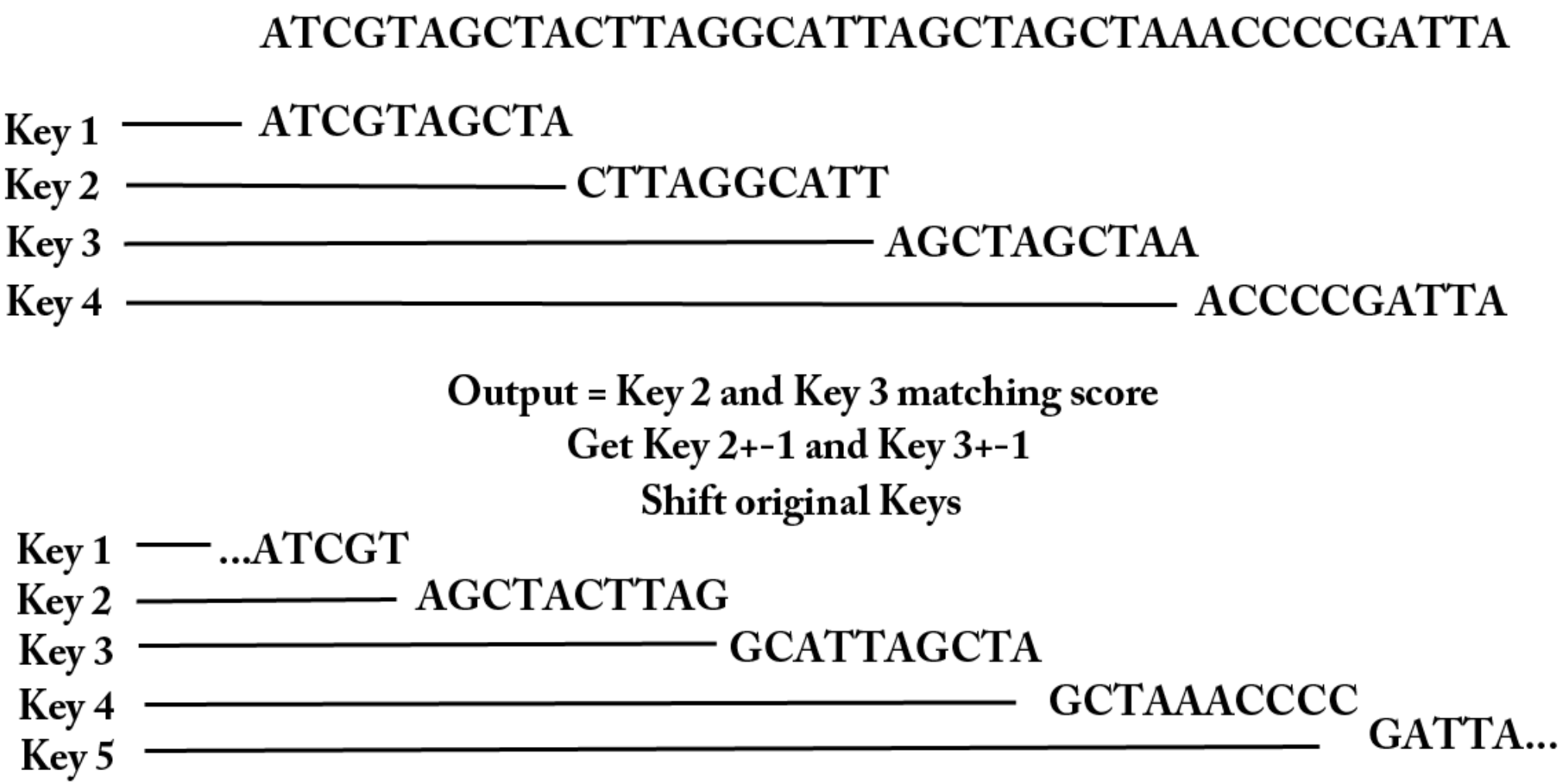
Run user-provided Reduce code

- Use newly reduced key value set to prepare another Map sequence.
- Each key will be prepared from the original DNA sequence with the complete new key having a sequence ranging from 1.5 to 2.5 of the original key or .5 to 1.5 if shifted left.
- New keys are prepared and Map program is run again.

Final Output

- A set of DNA sequences with a minimum match score.
- Running MapReduce multiple times can take advantage of global string matching algorithms to determine best length and value of each key.
- Each MapReduce process should decrease the number of required processes.

Visualization



Current Work

- Other groups and schools are using crowd-sourcing projects to find DNA sequence matches.
- Next-generation DNA sequencing projects provide sequences and toolkits for MapReduce.
- Currently these models are used for sequencing DNA as well as limited sequence matching.

Crowd-sourcing Phylo

- “More than 300,000 people have played Phylo since it launched in 2010, and Waldispuhl's team reported in a 2013 Genome Biology paper that up to 50 percent of the time, a casual gamer can match the performance of expert players; and up to 40 percent of the time, Phylo players can improve on the solution found by a computer program. “

Future Work

- Use the crowd-sourcing element with the parallel computing of MapReduce like SETI@home
- http://setiathome.ssl.berkeley.edu/
- Design efficient algorithms for mapping and searching.
- Reduce state space search (turning sequences into larger keys, turning sequences into individual protein producer types....)

References

- http://setiathome.ssl.berkeley.edu/
- http://abhishek-tiwari.com/post/mapreduce-and-hadoop-algorithms-in-bioinformatics-papers
- https://www.cs.umd.edu/sites/default/files/scholarly_papers/MichaelSchatz_1.pdf
- http://genome.cshlp.org/content/20/9/1297.short
- http://www.theguardian.com/environment/2013/aug/13/ash-dieback-facebook-fraxinus-game
- http://www.tgac.ac.uk/news/112/68/TGAC-releases-new-genetic-data-to-combat-ash-dieback-epidemic/
- http://www.forestry.gov.uk/chalara
- http://ghr.nlm.nih.gov/gene/CFTR
- http://www.cs.utoronto.ca/~brudno/bcb410/lec2notes.pdf
- http://www.aaai.org/Papers/ISMB/1997/ISMB97-008.pdf
- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-algorithms-for-computational-biology-spring-2005/lecture-notes/lecture5_newest.pdf
- http://biomedicalcomputationreview.org/content/biology-game-crowd
- http://www.clustal.org
- http://www.ebi.ac.uk/Tools/psa/