

## Search

- Motivations
  - Play tic-tac-toe
  - Play chess
  - Play Darwin\*

\* Except in Kansas

11/2/2005

1

## The Human Genome Project



- human DNA is a string of ~3 billion letters (A, T, G, C), making up about 20,000 genes

11/2/2005

2

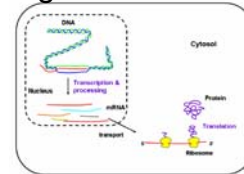
## The Human Genome Project

- Good news: truckloads of data
- Bad news: what does it *mean*?
- Figure it out (in part) by matching
  - match unknown sequence against sequences of known functionality
  - the hope: similarity of structure suggests similarity of function

11/2/2005

3

## Central Dogma of Modern Biology



Kuo, JBI 37 (2004) 293-303

- DNA encodes genes and is inherited
- DNA is transcribed under control of proteins into RNA
- RNA is translated into proteins by ribosomes
- Proteins run the cell, and thus organisms

11/2/2005

4

## Genetics

- Proteins are made up of amino acids
- DNA represents each amino acid by a triple of letters in the “alphabet” of 4 nucleotides: adenine, thymine, guanine, cytosine.
- Hence
  - two similar sequences of DNA letters →
  - two similar sequences of amino acids →
  - two similar structures in proteins →
  - similar biochemical behavior of the proteins

11/2/2005

5

## Matching

```
unk:  a t c g c c t a t t g t c g a c c
      ↑ ↑ ↑ ↑
known: a t a g c a g c t c a t c g a c g
```

11/2/2005

6

## The Biology Behind Matching

- Evolution happens.\*
- Changes to the genome during replication:
  - *Point mutations*: change a letter, e.g., C → A
  - *Omissions*: drop a letter
  - *Insertions*: insert a letter
- Similarity of sequence useful to discover
  - Similarity of function
  - Evolutionary history

\* Except in ...

11/2/2005

7

## More Complex Example

```

a a t c a g c a g c t c a t c g a c g g
  | | | | | | | | | |
a g a t c a g c a c t c a t c g a c g g

a a t c a g c a g c t c a t c g a c g g
                ^
a x a t c a g c a c t c a t c g a c g g
    
```

11/2/2005

8

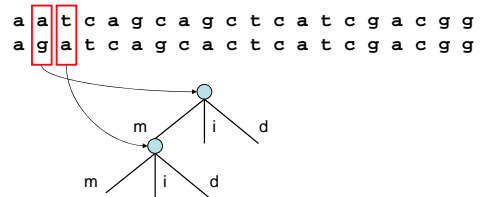
## Matching

- Every differing position has 3 possible explanations:
  - mutation
  - insertion
  - deletion

11/2/2005

9

## Matching As Tree Search

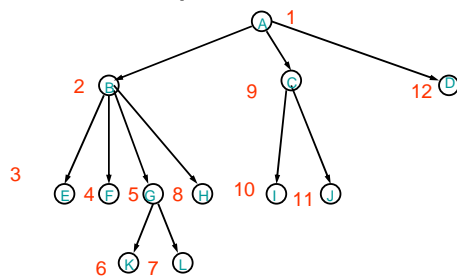


Every path through the tree is an hypothesis about how one sequence matches another

11/2/2005

10

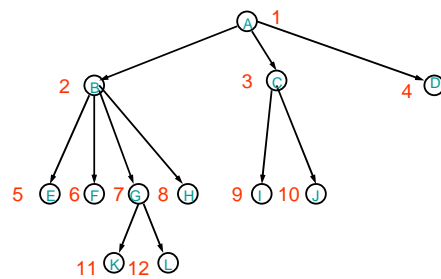
## Depth first search



11/2/2005

11

## Breadth first search



11/2/2005

12

## If it's 6.001

- It's gotta have code:

```
(define (dfsearch start-state)
  (define (search1 queue)
    (cond ((null? queue)
           (display "done"))
          (else
           (display "visiting ")
           (display (car queue))
           (search1 (append (children (car queue))
                             (cdr queue))))))
  (search1 (list start-state)))
```

11/2/2005

13

## If it's 6.001

- It's gotta have code:

```
(define (bfsearch start-state)
  (define (search1 queue)
    (cond ((null? queue)
           (display "done"))
          (else
           (display "visiting ")
           (display (car queue))
           (search1 (append (cdr queue)
                             (children (car queue))))))
  (search1 (list start-state)))
```

11/2/2005

14

## Matching

```
a t c a g c c t a t t g t c g a c c
  ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
a t a g c c t a t t g t c g a c c

a t x a g c c t a t t g t c g a c c
a t a g c c t a t t g t c g a c c
```

11/2/2005

15

## Define a Distance Metric

- Given two sequences, s1 & s2,
  - Distance is 0 if they are identical
  - Penalty for each point mutation
    - Different for different mutations
  - Penalty for insertion/deletion of nucleotides
  - "Distance" is sum of penalties
- Now we can get the best explanation.

11/2/2005

16

## Representing Mutation Penalty

	A	C	G	T
A	0	.3	.4	.3
C	.4	0	.2	.3
G	.1	.3	0	.2
T	.3	.4	.1	0

11/2/2005

17

## We have the Penalties

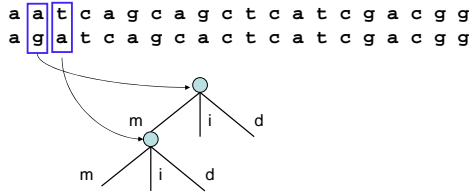
```
point-mutations
>>
(table2
 table1
 (t (table1 (t 0) (g 0.1) (c 0.4) (a 0.3)))
 (g (table1 (t 0.2) (g 0) (c 0.3) (a 0.1)))
 (c (table1 (t 0.3) (g 0.2) (c 0) (a 0.4)))
 (a (table1 (t 0.3) (g 0.4) (c 0.3) (a 0))))

(define omit-penalty .5)
(define insert-penalty 0.7)
```

11/2/2005

18

## Matching As Tree Search



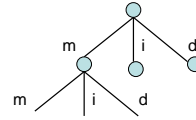
*Time complexity?*

11/2/2005

19

## Observation

a a t c a g c a g c t c a t c g a c g g  
a g a t c a g c a c t c a t c g a c g g



11/2/2005

22

## Memory to the Rescue

- "Memoization"
- Store the results of computing sub-paths and substitute lookup for computation
- Still,  $\sim n^2$

11/2/2005

23

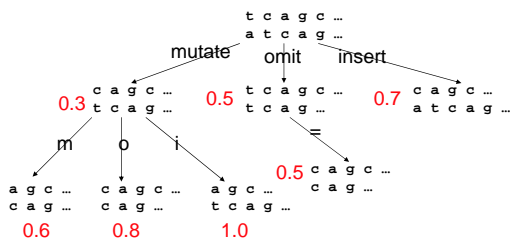
## Can We Be Smarter Still?

- Cut off bad paths:
  - Estimate an upper bound on matches of interest
  - Declare any match worse than this to be infinitely bad (and stop pursuing it)
- Advantages?
- Disadvantage?

11/2/2005

24

## Idea: Pursue "Best" Matches



11/2/2005

25

## Best First Search

- Extend only the best sequence

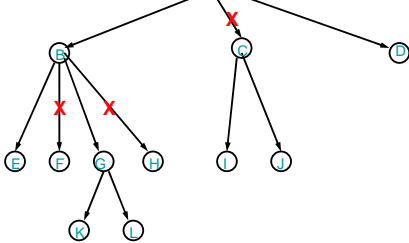
```
(define (bestsearch start-state)
  (define (search1 queue)
    (cond ((null? queue)
          (display "done"))
          (else
           (display "visiting ")
           (display (car queue))
           (search1 (merge (sort (children (car queue)))
                           (cdr queue))))))
  (search1 (list start-state)))
```

11/2/2005

26

## Beam Search

- **Beam:** like best-first, but keep only  $n$  best children of a node



11/2/2005

27

## Framework for Search

```
(define (search start-state done? succ-fn merge-fn)
  (define (search1 queue)
    (if (null? queue)
        #f
        (let ((current (car queue)))
          (if (done? current)
              current
              (search1
               (merge-fn (succ-fn current)
                         (cdr queue)))))))
  (search1 (list start-state)))
```

- Have we reached "goal"?
- Order in which to explore moves
- What "moves" can we make from current state?

11/2/2005

28

## Varieties of Search

- depth first  
(append (children (car queue))(cdr queue))
- breadth first  
(append (cdr queue)(children (car queue)))
- best first  
(merge (sort (children (car queue))) (cdr queue))
- beam search  
(merge (list-head n(sort (children (car queue)))) (cdr queue))

11/2/2005

29

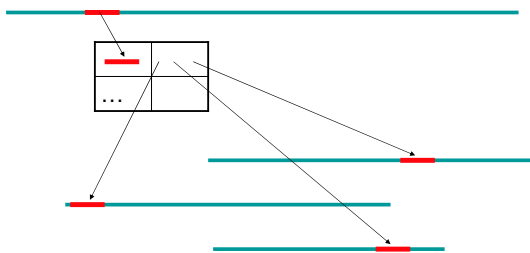
## Return of the Biologists

- Some large subsequences are common ( clichés)
- Good matches will contain large identical subsequences
- Pre-compute table of all occurrences of specific patterns
- Extend match outward (both directions) from these exact matches

11/2/2005

30

## BLAST: Find common, extend

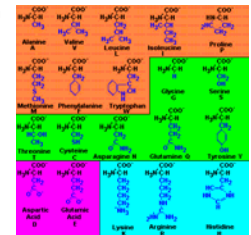


11/2/2005

31

<http://www.people.virginia.edu/~rjh9u/aminacid.html>

## Generalize

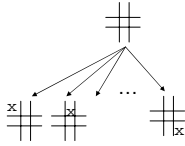


- DNA
  - Nucleotides: A, C, T, G
  - Mutation rates
  - Insertion/omission penalties
- Proteins
  - Amino Acids: val, leu, ile, met, phe, asn, glu, gln, ...
  - Mutation rates
  - Insertion/omission penalties

11/2/2005

32

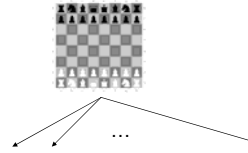
# Let's Play Games



11/2/2005

33

# Let's Play Games



11/2/2005

34