# Bioinformatics for biomedicine

# Sequence search: BLAST, FASTA

Lecture 2, 2006-09-26

Per Kraulis

http://biomedicum.ut.ee/~kraulis

# Previous lecture: Databases

- General issues
  - Data model
  - Quality
  - Policies
    - Updates, corrections
- EMBL, GenBank, Ensembl
- UniProt
- Access: EBI, NCBI (Entrez)

# Course design

1. What is bioinformatics? Basic databases and tools
2. **Sequence searches: BLAST, FASTA**
3. Multiple alignments, phylogenetic trees
4. Protein domains and 3D structure
5. Seminar: Sequence analysis of a favourite gene
6. Gene expression data, methods of analysis
7. Gene and protein annotation, Gene Ontology, pathways
8. Seminar: Further analysis of a favourite gene

# Sequence searches

Two tasks:

3) Compare two sequences: How similar?
4) Search for similar sequences

# How to do it? Computer program

- Algorithm
  - Appropriate
  - Correct
  - Speed

- Database
  - Content

# Consider!

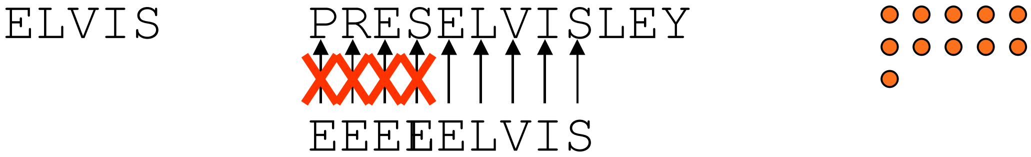
- Sensitivity
  - Are correct hits found?
- Specificity
  - Are false hits avoided?
- Statistics: Significant match?
- Biological judgement
  - “Strange” features in sequences
  - Are assumptions OK?

# What is an algorithm?

- “Procedure for accomplishing some task”
  - Set of well-defined instructions
    - Cookbook recipe
  - Produce result from initial data
    - Input data set -> output data set

- All software implements algorithms

# Example: substring search



ELVIS

PRESELVISLEY

EEEEELVIS

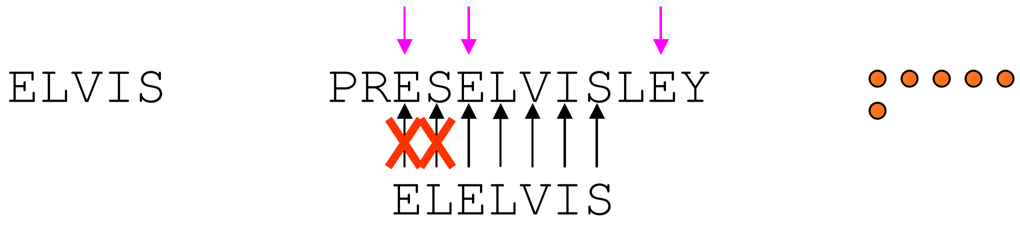
Naïve algorithm: 11 operations

# Substring search with lookup

PRESELVISLEY

Index lookup table

| | | | |
|---|---|---|---|
| P | 1 | L | 6,10 |
| R | 2 | V | 7 |
| E | 3,5,11 | I | 8 |
| S | 4,9 | Y | 12 |



Improved algorithm: 6 operations

But: preprocessing required

# Algorithm properties

- Execution time
  - Number of operations to produce result
- Storage
  - Amount of memory required
- Result
  - Exact: Guaranteed correct
  - Approximate: Reasonably good

# Analysis of algorithms, 1

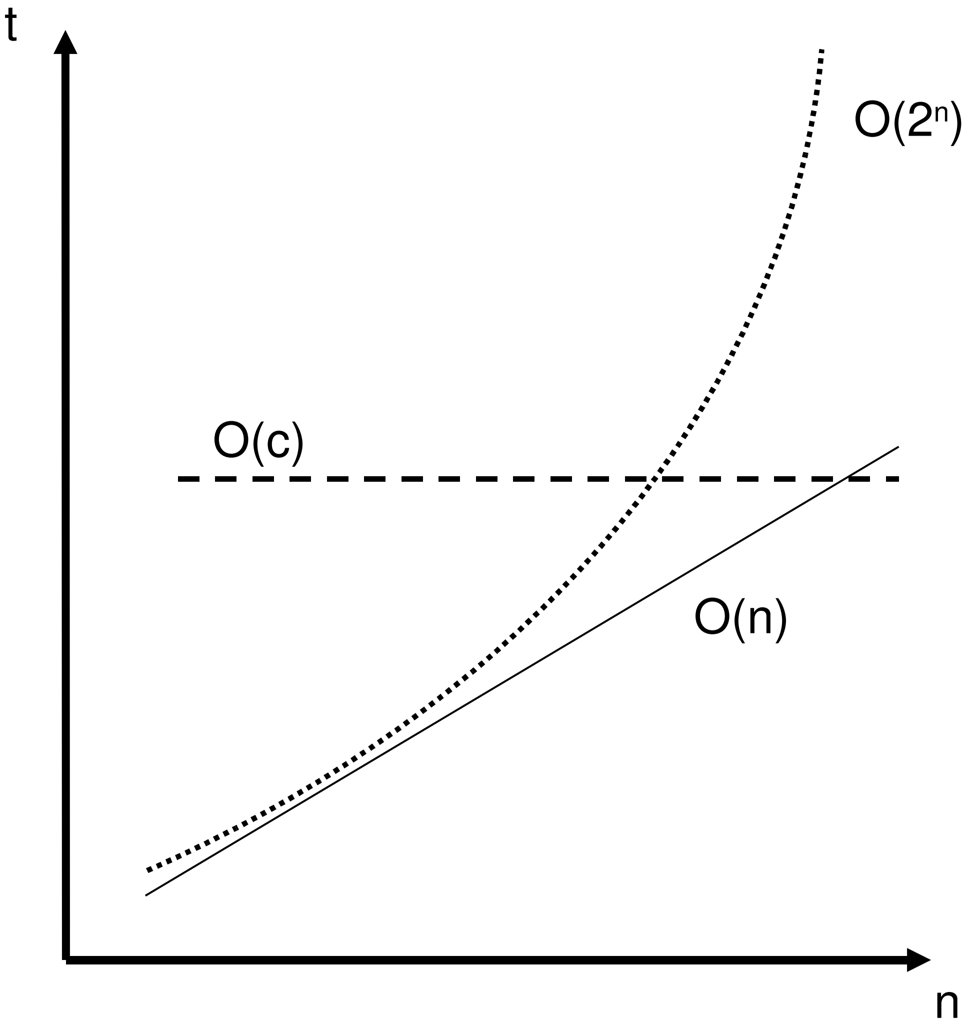
- Larger input data set: what happens to
  - Execution time?
  - Storage?

- Examples:
  - Longer query sequence
  - Larger database
  - More sequences in multiple alignment

# Analysis of algorithms, 2

- “Complexity” of an algorithm
  - Behaviour with larger input data sets
  - Time (speed) and storage (memory)

- Big-O notation: general behaviour
  - $O(c)$ constant
  - $O(\log(n))$ logarithmic
  - $O(n)$ linear
  - $O(n^2)$ quadratic
  - $O(2^n)$ exponential


t
O(2ⁿ)
O(c)
O(n)
n

# Example: O(n)

- Compute mol weight $M_W$ of protein

- Table: $M_W$(aa) for each amino acid residue
- For each residue in protein, add $M_W$
  - $M_W$(Met) + $M_W$(Ala) + … + $M_W$(Ser)
- Add $M_W$ for water

- O(n) for protein size

# Example: $O(2^n)$

- Given mol weight $M_W$ for a protein, compute all possible sequence that might fit

- Table: $M_W$(aa) for each amino acid residue
- Produce all permutations up to $M_W$
  - MAAAA, MAAAG, MAAAS, MAAAT, …

- Naïve implementation: $O(2^n)$ for $M_W$

# Heuristic algorithms

- Less-than-perfect
  - Reasonably good solution in decent time
- Why?
  - Faster than rigorous algorithm
  - May be the only practical approach
- Specific to the task
  - Reasonable or likely cases
    - Rule-of-thumb
    - Use biological knowledge

# Sequence comparison

- Sequences related by evolution
  - Common ancestor
  - Modified over time
  - Biologically relevant changes
    - Single-residue mutations
    - Deletion/insertion of segments

- Sequences may be related by evolution, although we cannot detect it

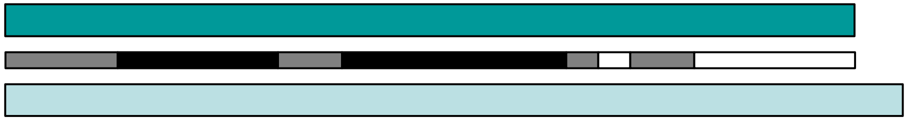
# Alignment

```
PRESELVISLEY
||| ||.||| |
PREPELIISL-Y
```

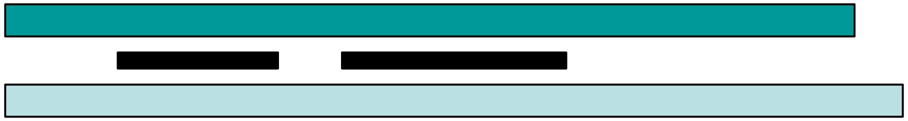
- Corresponding segments of sequences
- Identical residues
- Conserved residues
- Gaps for deletion/insertion

# Local vs. global alignment

- Global alignment: entire sequences



- Local alignment: segments of sequences



- Local alignment often the most relevant
  - – Depends on biological assumptions

# Alignment matrix, 1

- Mark identical residues
- Find longest diagonal stretch
- Local alignment
- O(m*n)

```
    PRESELVISLEY

E     X X     X
L        X   X
V         X
I          X
S      X    X
```

# Alignment matrix, 2

- Mark similar residues
  - Substitution probability
- Find longest diagonal stretch
  - Above some score limit
- Local alignment
  - High Scoring Pair, HSP

```
    PRESEIVISLEY

E     X X     X
L            X
V         X
I        X X
S      X    X
```

```
    PRESEIVISLEY

E     X X     X
L        . . X
V         X
I        X X .
S      X    X
```

# Substitution matrix

- The probability of mutation X -> Y
  - M(i,j) where i and j are all amino acid residues
  - Transformed into log-odds for computation

- Common matrices
  - PAM250 (Dayhoff et al)
    - Based on closely similar proteins
  - BLOSUM62 (Henikoff et al)
    - Based on conserved regions
    - Considered best for distantly related proteins

# Gap penalties

- To model deletion/insertion
  - Segment of gene deleted or inserted

```
PRESELVISLEY
||| ||.||| |
PREPELIISL-Y
```

- Gap open
  - Start a gap: should be tough
- Gap extension
  - Continue a gap: should be easier

# Alignment/search algorithms

- Needleman-Wunsch
- Smith-Waterman
- FASTA
- BLAST

# Needleman-Wunsch, 1970

- Global alignment
- Rigorous algorithm
  - Dynamic programming
  - Simple to implement
- Slow; not used for search
- http://bioweb.pasteur.fr/seqanal/interfa

# Smith-Waterman, 1981

- Local alignment
- Rigorous algorithm
  - Dynamic programming
  - Fairly simple to implement
- Precise, sensitive alignments
- Slow; not used for search
- SSEARCH in the FASTA package
- http://pir.georgetown.edu/pirwww/search/

# FASTA, Lipman Pearson 1985

- Local alignment
- Heuristic algorithm
  - Table lookup, “words” of length ktup
    - Higher ktup: faster but less sensitive
    - Protein: ktup=2
    - Nucleotide: ktup=6
  - Extension of hits into alignments
- Faster than Smith-Waterman
- Useful for searches

# FASTA statistics

- Fairly sophisticated statistics
  - But still fallible
- E-value (expectation value)
  - The number of hits with this score expected, if query were a random sequence
  - Values should be low
    - Below 0.001 almost certainly significant
    - 0.001 to 0.1 probably significant
    - 0.1 to 10 may be significant
    - 10 and above probably rubbish

# FASTA example

- http://www.ebi.ac.uk/fasta33/
- Example search
  - Query: UniProt P04049 (RAF1_HUMAN)
  - Standard parameters, fasta3, UniProt
  - Kept only 24 hours at EBI
  - http://www.ebi.ac.uk/cgi-bin/sumtab?tool=fasta

# BLAST, Altschul et al 1990

- Basic Local Alignment Search Tool
- Heuristic algorithm
  - Basically similar ideas as FASTA
  - Did not originally allow gaps
  - BLAST2 allows gaps
- ~50 faster than Smith-Waterman
- Faster than FASTA, less sensitive
- E-value statistics: same idea as FASTA

# BLAST example

- http://www.ncbi.nlm.nih.gov/BLAST/
- Example search
  - Query: UniProt P04049 (RAF1_HUMAN)
  - Standard parameters, human proteins
  - http://www.ncbi.nlm.nih.gov/BLAST/Blast.cgi
  - 1159262528-13488-10186840195.BLASTQ2
- http://www.ebi.ac.uk/blast/index.html

# BLAST and short nucleotides

- Default BLAST parameters are for genes and proteins

- Oligonucleotides require other parameters for meaningful results
- http://www.ncbi.nlm.nih.gov/BLAST/ special link

# BLAST and low complexity regions

- Some proteins contain “low complexity” regions, e.g. S, T, Q in long peptides
- Spurious high significance
  - Does not make biological sense
- Filter out such regions
  - BLAST uses SEG algorithm
  - Regions masked out; replaced by “XXXX”
  - May go wrong; check results!

# Variants of search programs

| **Query** | **Database** | **Program** | **Comment** |
|---|---|---|---|
| Protein | Protein | blastp<br>fastp | |
| Nucleotide | Nucleotide | blastn<br>fastn | Use only if nucleotide comparison is really wanted |
| Nucleotide | Protein | blastx<br>fastx3 | Translate query to protein;<br>6-frame |
| Protein | Nucleotide | tblastn<br>tfastx3 | Translate DB on the fly;<br>6-frame |
| Nucleotide | Nucleotide | tblastx | Translate both query and DB (gene-oriented); 2* 6-frame |