Computing in Systems Biology

Some challenges for your computing skills.

Peter Krusche
Warwick Systems Biology Centre

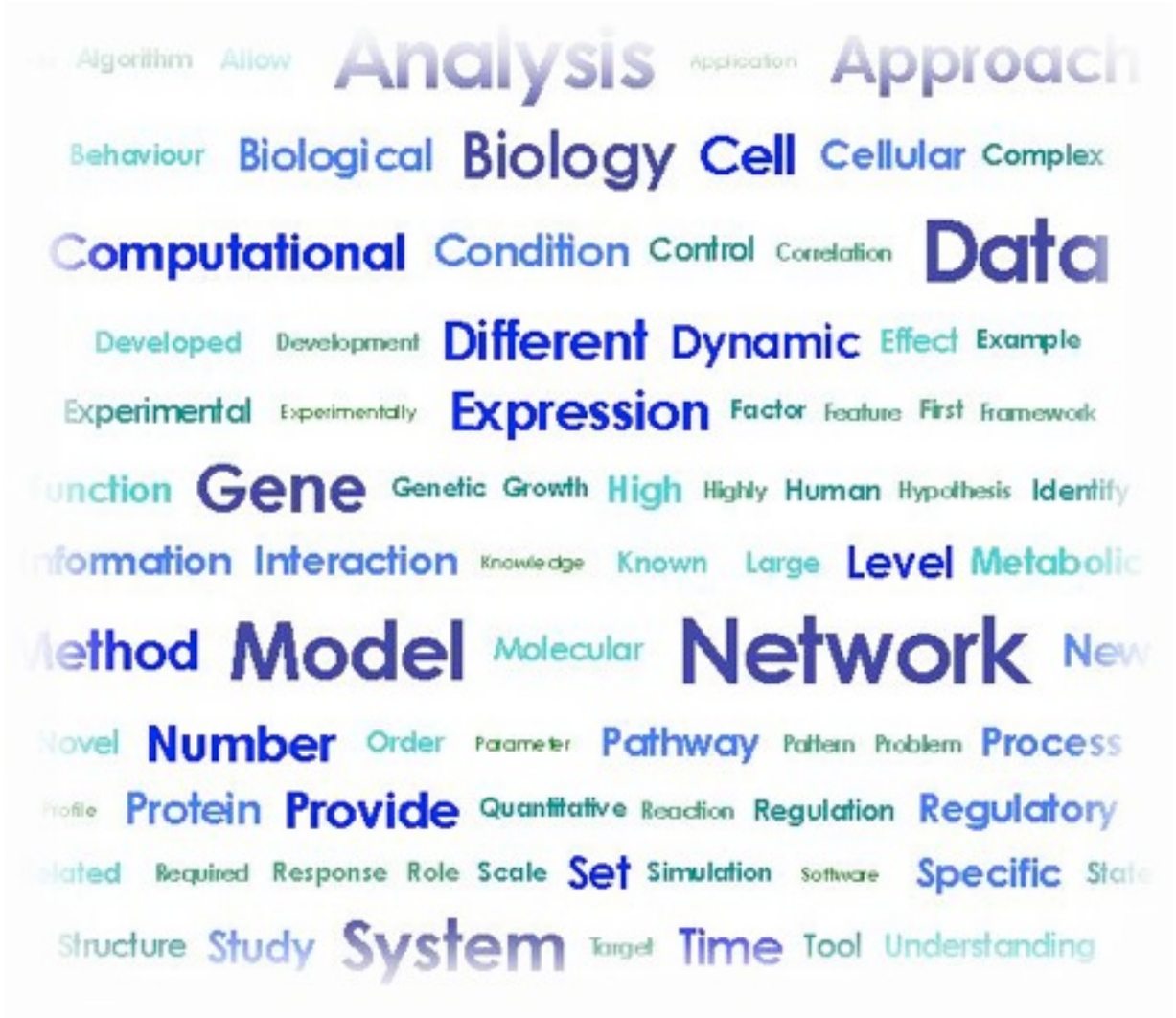
In this lecture...

What is Systems Biology?

[Some Biology and a computer scientist's perspective]

What does Computer Science have to do with it?

[An example involving character sequences, other interesting problems]

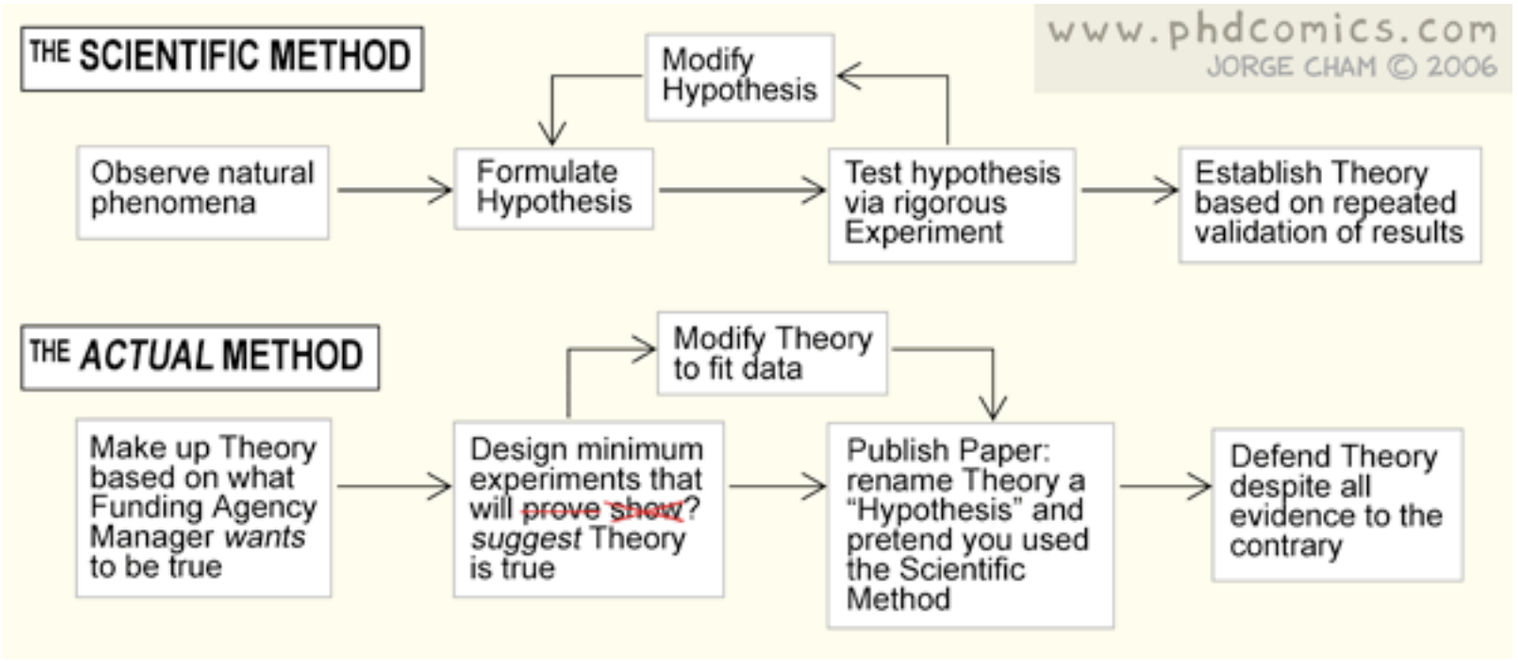


Systems Biology?

It's real science!

The goal: understand and explain the mechanisms that underlie life.

Another way to look at it: "Reverse-engineer" nature.



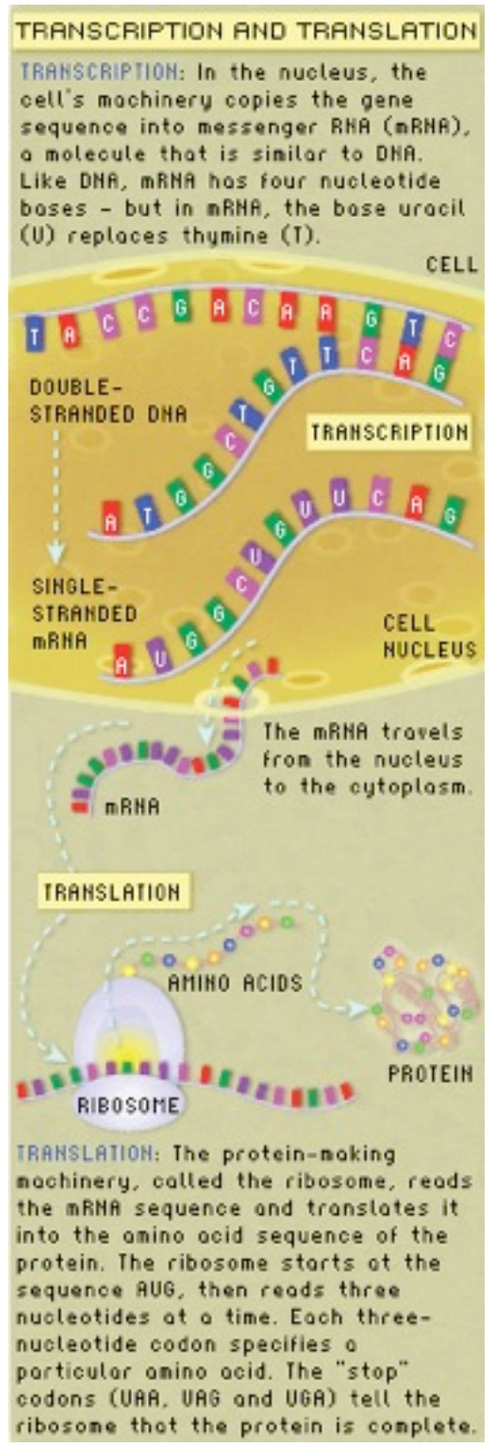
So, how does "life" work?

DNA material describes how to make proteins.

In living cells, the DNA is normally stored in the nucleus.

DNA can be *transcribed* into mRNA, which can be *translated* into proteins.

Proteins participate in most processes in cells.



Source: http://www.genome.gov/Glossary/

Proteins?

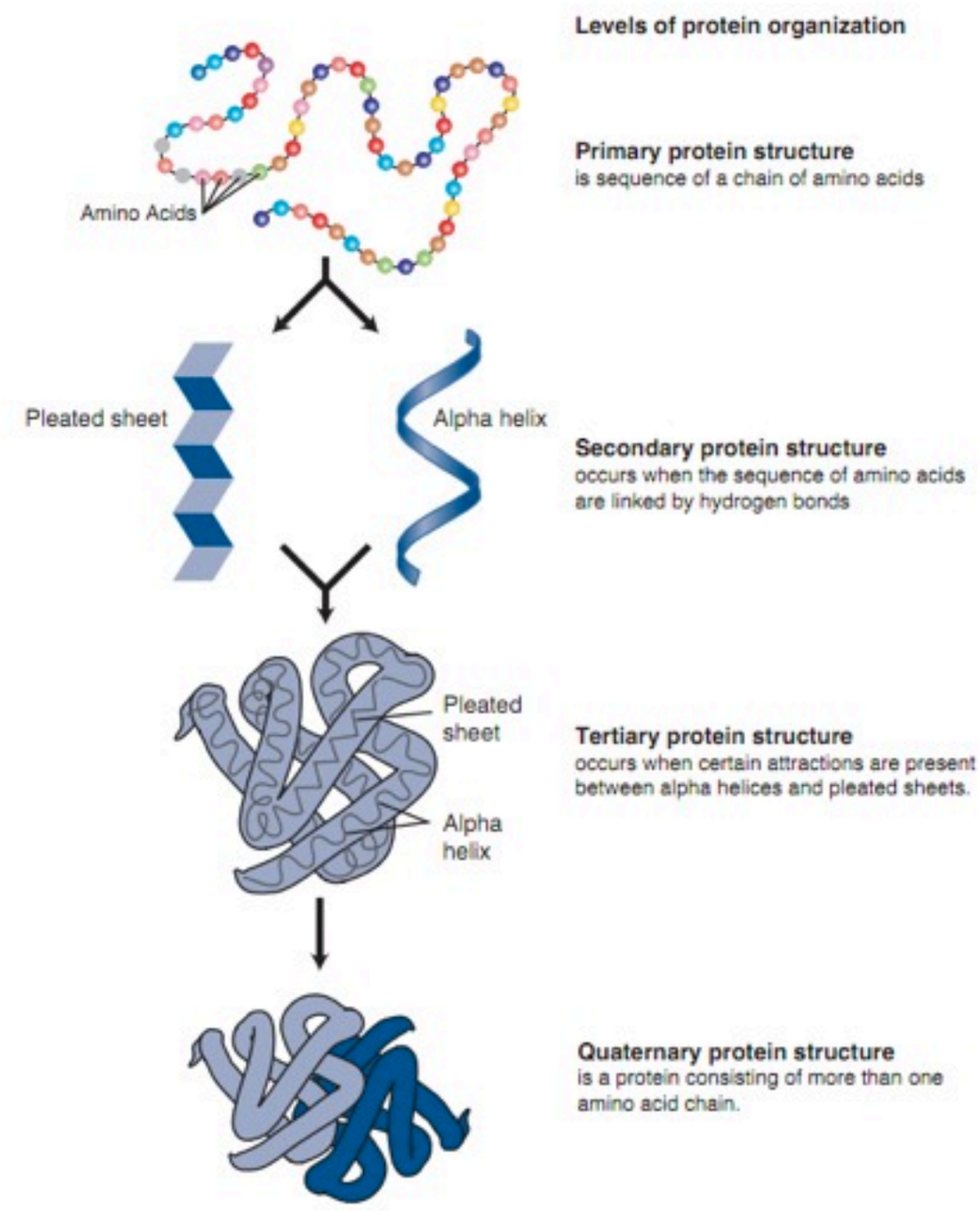
Complex macromolecules, which can have specific functions, such as:

Catalyse biochemical reactions (*Enzymes*)

Structural or mechanical functions (e.g. maintaining cell shape)

Cell signalling, immune responses, cell adhesion, and the cell cycle.

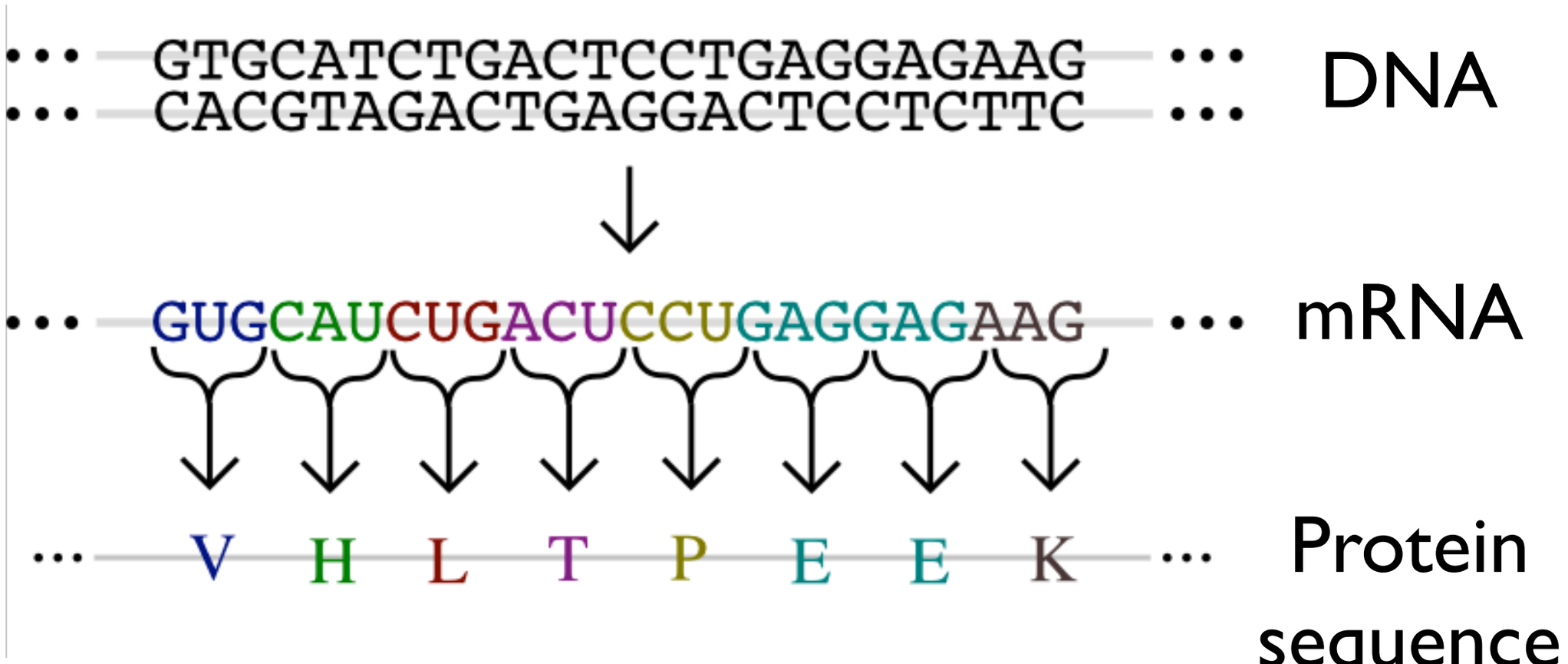
Transcription factors start up the transcription of genes.



Source: http://www.genome.gov/Glossary/

Proteins for computer scientists

Transcription and translation represented as character strings:



Source: http://en.wikipedia.org/wiki/Protein

Main Systems Biology Activities

1) Analysis of experimental data
   [statistics, sequence analysis, image analysis, ...]

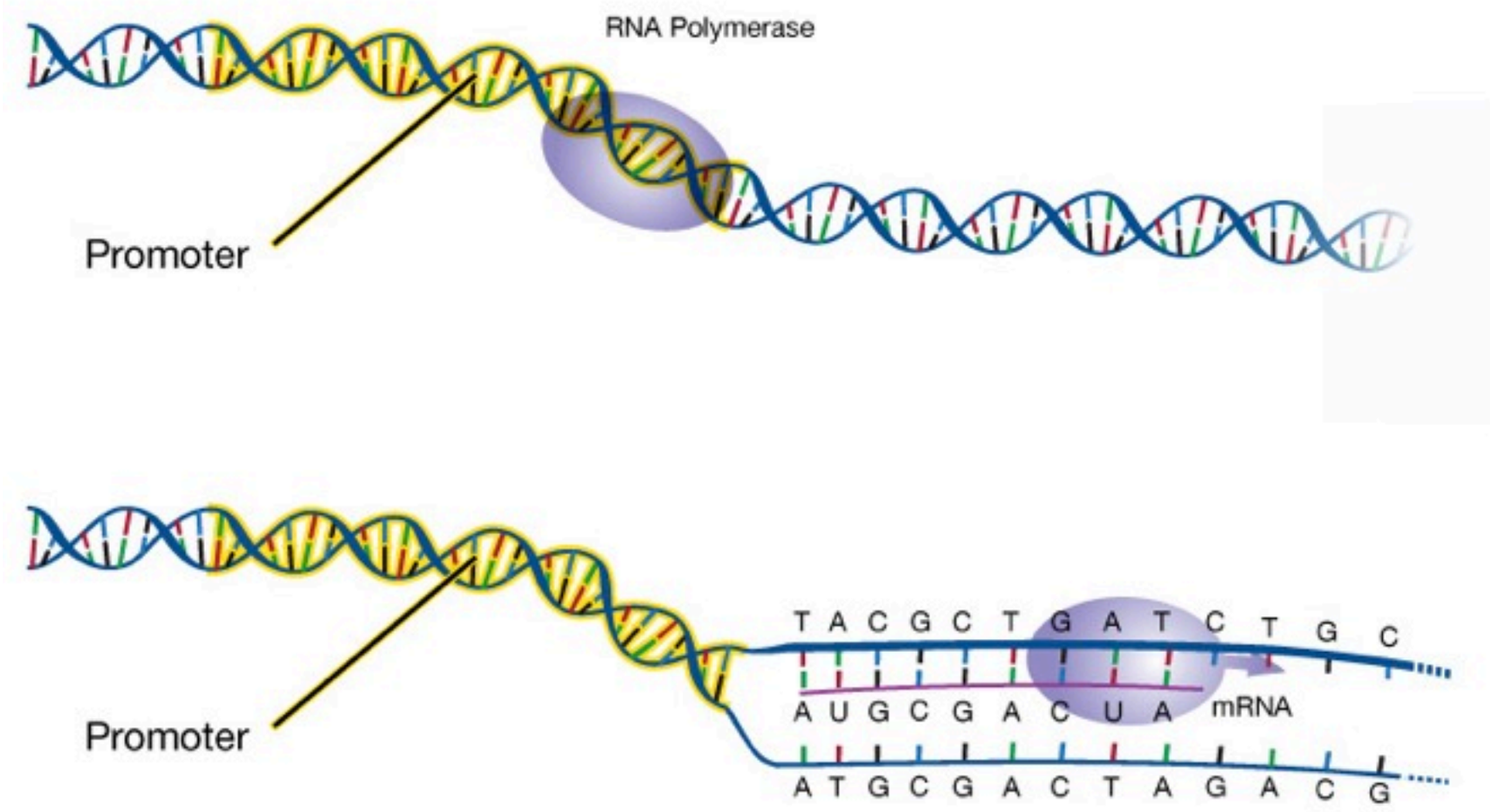
2) Create models e.g. of complex protein interactions and biochemical reactions
   [kinetics, network models, model parameter estimation, ...]

3) Suggest new experiments to provide evidence for models
   [communicate with biologists]

Computing in Biology

Promoter Sequence Analysis

Promoter sequences are the DNA sequences that surround "coding areas" that will be transcribed+translated into protein.

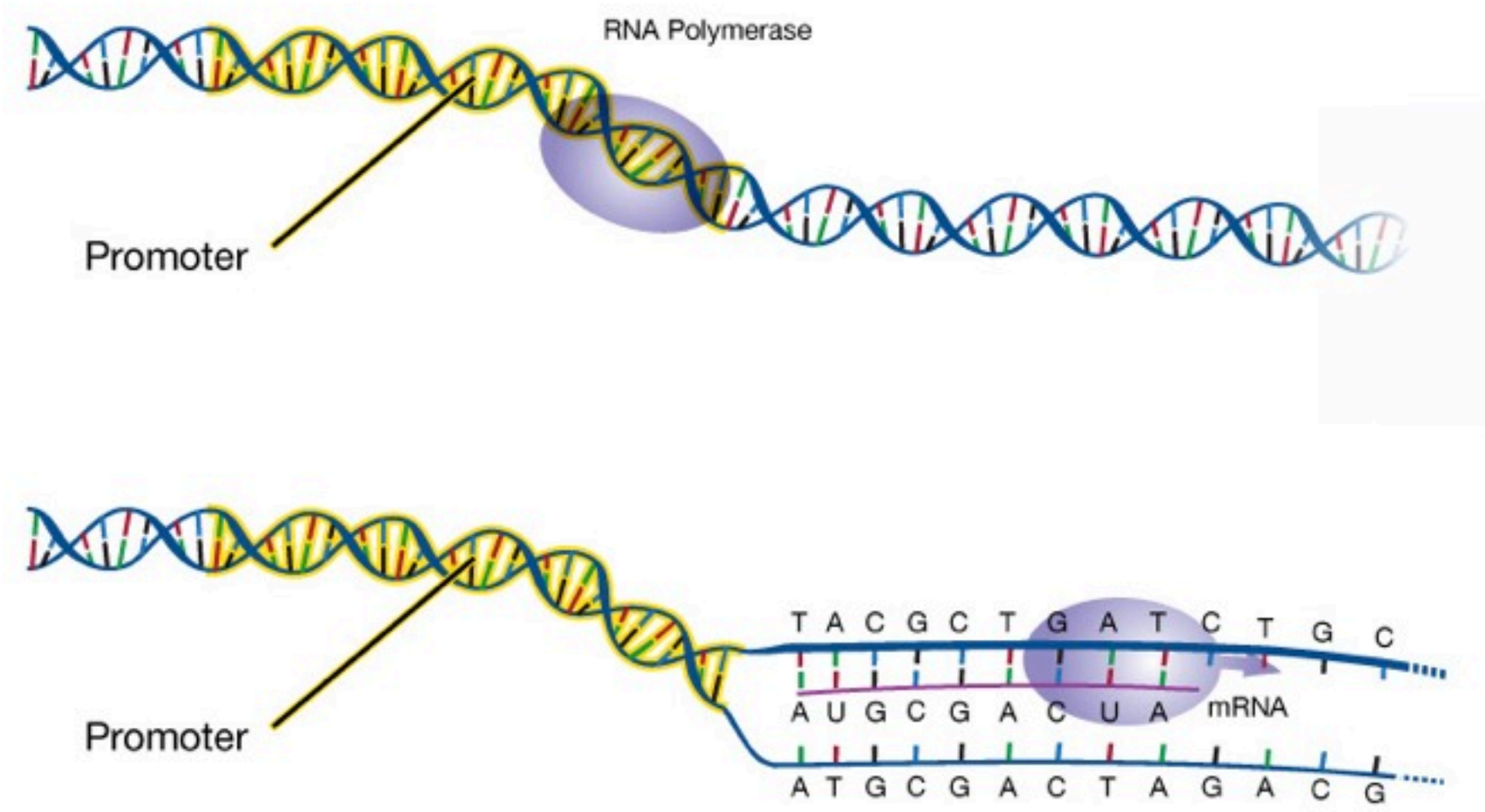


Source: http://www.genome.gov/Glossary/

Promoter Sequence Analysis

Promoter sequences are the DNA sequences that surround "coding areas" that will be transcribed+translated into protein.

The promoter sequence normally controls when a gene gets *expressed* as a protein.



Source: http://www.genome.gov/Glossary/

How do promoters "control" expression?

Promoter sequences can contain *Motifs*.

Motifs are short sequence fragments, which will attract transcription factor proteins (which then causes transcription and production of a protein).

Example:

E-box:

normally present in promoters of genes which are expressed rhythmically.

Sequence Conservation

Sequences that are very similar across multiple species can be *evolutionarily conserved*.

Source: http://www.ncbi.nlm.nih.gov

Sequence Conservation

Sequences that are very similar across multiple species can be *evolutionarily conserved.*

Conservation indicates similarity in function for genetic sequence regions.

Source: http://www.ncbi.nlm.nih.gov

Sequence Conservation

Sequences that are very similar across multiple species can be *evolutionarily conserved.*

Conservation indicates similarity in function for genetic sequence regions.

Conserved regions in promoters are likely to contain areas that are relevant to expression regulation.

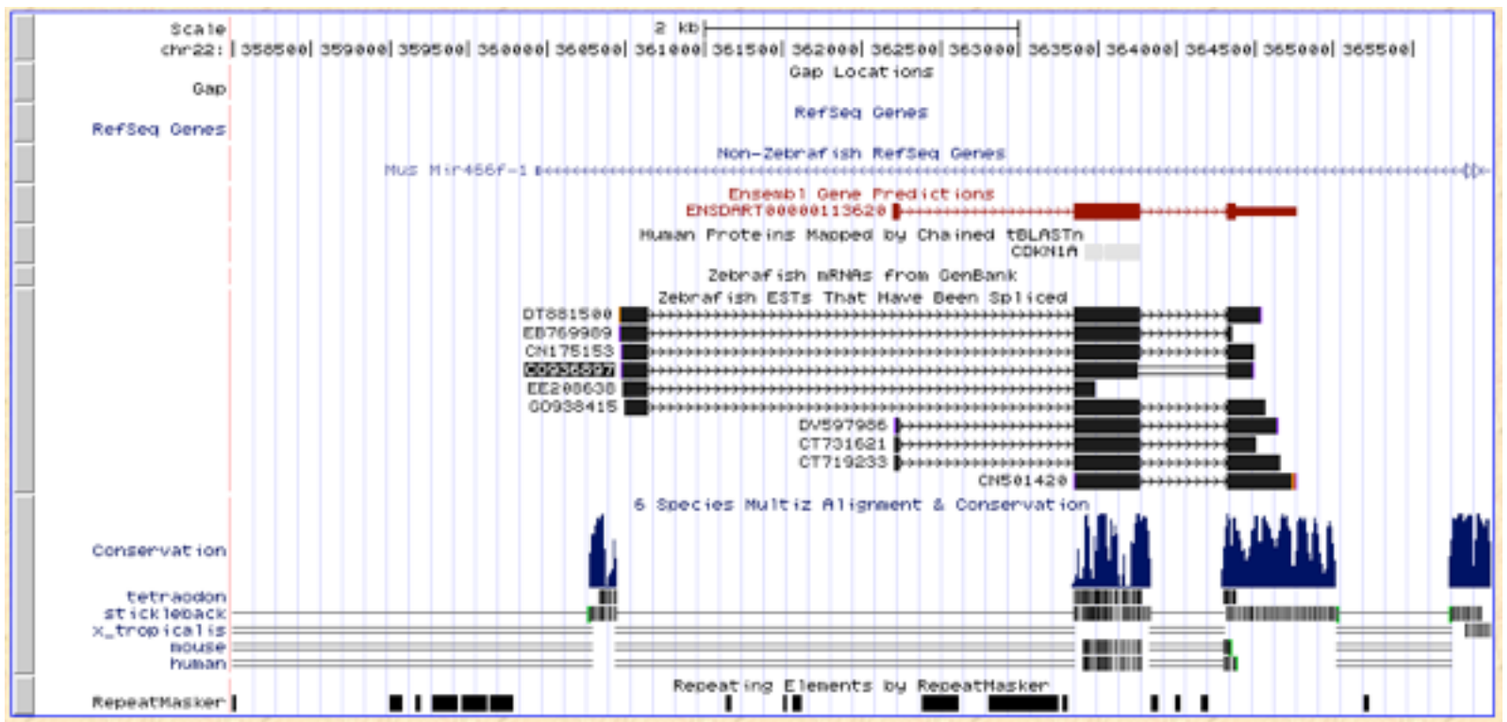
Source: http://www.ncbi.nlm.nih.gov

Sequence Conservation

Sequences that are very similar across multiple species can be *evolutionarily conserved.*

Conservation indicates similarity in function for genetic sequence regions.

Conserved regions in promoters are likely to contain areas that are relevant to expression regulation.



Source: http://www.ncbi.nlm.nih.gov

Motif finding (one way)

We want to find possible locations of motifs that are significant for controlling the expression of a certain gene.

1. Locate areas in which we are likely to find such motifs.

2. Do a statistical analysis of these areas to locate likely motif locations.

3. Biologists do experiments by mutating these locations in DNA, see if expression changes (very time-consuming).

Being able to narrow down where to look is important!

Some Algorithms for Genetic Sequences

Basic problem types:

- String search
- String comparison

Example (exact substring search):

Find TCA in ACCAGTCACCT.

String Basics

1) A string is a sequence of characters from a fixed size alphabet, e.g. {A, G, C, T} for DNA.

2) Contiguous subsequences are called *substrings*, *windows*, or *factors*.

TCA is a substring in ACCAGTCACCT

3) We also consider not necessarily contiguous *subsequences*.

AATAC is a subsequence of in ACCAGTCACCT

String Comparison

1) *Hamming Distance:* count mismatches.
dist(AACACCTACG, AAGACCAACT) = 3

2) *String alignment*: align maximum number of letters, preserving order.

```
AACT ACCCTAA C G
||   |||  ||  |
GAA  GACC  AAGCT
```

The aligned characters form the longest common subsequence (LCS).

LCS distance: dist(x,y) = |x| + |y| - 2 * |LCS(x, y)|

String comparison

Other measures of string (dis-)similarity:

1) *Edit distance* of strings x, y: minimum number of edit operations (character insertion, deletion, substitution) to transform x into y, or back.

2) *Gapped alignment:* we assign score 1 to matching character pairs, 0 to mismatching pairs, -0.5 to gap insertion, maximize score for input strings.

3) *Weighted alignment:* We assign arbitrary weights from a pairwise score matrix to matching arbitrary characters from the input alphabet.

Classical Solutions for String Alignment

1) Longest common subsequence: Wagner/Fischer, '74

2) Global (weighted) alignment: Needleman/Wunsch, '70

Both computations can be carried out in time $O(n^2)$

Local alignment can also be done in time $O(n^2)$, see Smith/Waterman '81

Local String Alignment

We are looking for regions of local similarity in two input strings (worst case example: compare all pairs of substrings in our two input sequences).

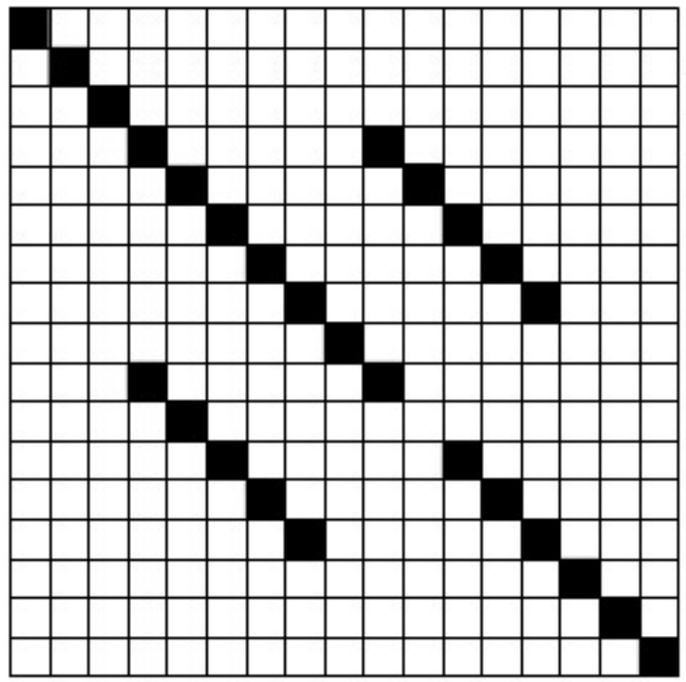
1) BLAST (and similar algorithms): Start with exact substring matches to *seed* gapped alignments. Very fast heuristic method.
   Drawback: Can miss alignments in regions with low similarity.

2) Dot Plots: Compare all pairs of fixed-size windows using the Hamming distance.

Alignment Plots

Main drawback of dot plots:
Hamming score not very accurate.

We will use an alignment score instead.



Input:
Strings x and y, window length w

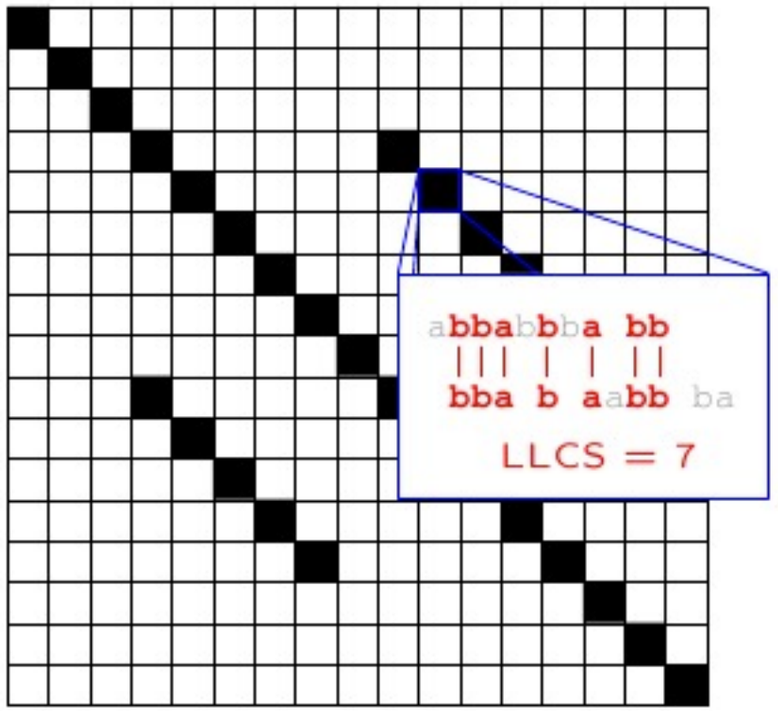
Output:
Scores for all pairs of w-windows in x and y.

Alignment Plots

Naive approach: $O(|x|\ |y|\ w^2)$

Ott et al. '08: Heuristic improvements, x25 speedup, same worst case.

Rasmussen et al. '05: Very fast algorithm for computing the scores of window pairs with >90% similarity.



How big? How fast?

We would like to compare many sequences that are very large.

- Worst case: entire genomes (starting at 30M bases, up to 1T bases).

- practically, we compare the area around a couple of genes, around 10k bases to 150k bases, pairwise across multiple species

- Window sizes: around 100 bases.

- The faster the better (less time per sequence pair allows to compare larger sequences).

How to do it faster (...than biologists)

Suggestions welcome.

How to do it faster (...than biologists)

Suggestions welcome.

- Buy faster CPUs.
- Use Parallel Computing.
- Use GPUs.
- Implement faster/better.
- Design better algorithms.

Why bother?

BLAST is very fast and pretty accurate. Why not use it?

For comparing highly similar sequences, we can do smart stuff (BLAST, Rasmussen et al., q-grams).

However: We are interested in sequences with similarities as low as 60%.

Then, more thorough comparison can pay off.

So, how fast can we go?

Pairwise alignment plot computation single CPU execution times on different datasets.

Data Set	Mikey	Berti	Jimmy	Henry
Input Size	2.7k × 0.6k	2.7k × 2.3k	15k × 97k	80k× 80k
Heur	5.1 (÷ 1.0)	41.1 (÷ 1.0)	2677 (÷ 1.0)	11708 (÷ 1.0)
BLCS	3.6 (÷ 1.4)	37.3 (÷ 1.1)	3680 (÷ 0.7)	16191 (÷ 0.7)
Sea-16	1.4 (÷ 3.6)	10.8 (÷ 3.8)	1026 (÷ 2.6)	4514 (÷ 2.6)
Sea-8	0.5 (÷ 10.2)	3.8 (÷ 10.8)	368 (÷ 7.3)	1614 (÷ 7.3)
Sea-8 SMP×2	0.3 (÷ 17.0)	3.4 (÷ 12.1)	210 (÷ 12.7)	821 (÷ 14.3)

(Execution times in seconds)

So, how fast can we go?

Pairwise alignment plot computation single CPU execution times on different datasets.

Ott et al. (the Biologists)

Data Set	Mikey	Berti	Jimmy	Henry
Input Size	2.7k × 0.6k	2.7k × 2.3k	15k × 97k	80k× 80k
Heur	5.1 (÷ 1.0)	41.1 (÷ 1.0)	2677 (÷ 1.0)	11708 (÷ 1.0)
BLCS	3.6 (÷ 1.4)	37.3 (÷ 1.1)	3680 (÷ 0.7)	16191 (÷ 0.7)
Sea-16	1.4 (÷ 3.6)	10.8 (÷ 3.8)	1026 (÷ 2.6)	4514 (÷ 2.6)
Sea-8	0.5 (÷ 10.2)	3.8 (÷ 10.8)	368 (÷ 7.3)	1614 (÷ 7.3)
Sea-8 SMP×2	0.3 (÷ 17.0)	3.4 (÷ 12.1)	210 (÷ 12.7)	821 (÷ 14.3)

(Execution times in seconds)

So, how fast can we go?

Pairwise alignment plot computation single CPU execution times on different datasets.

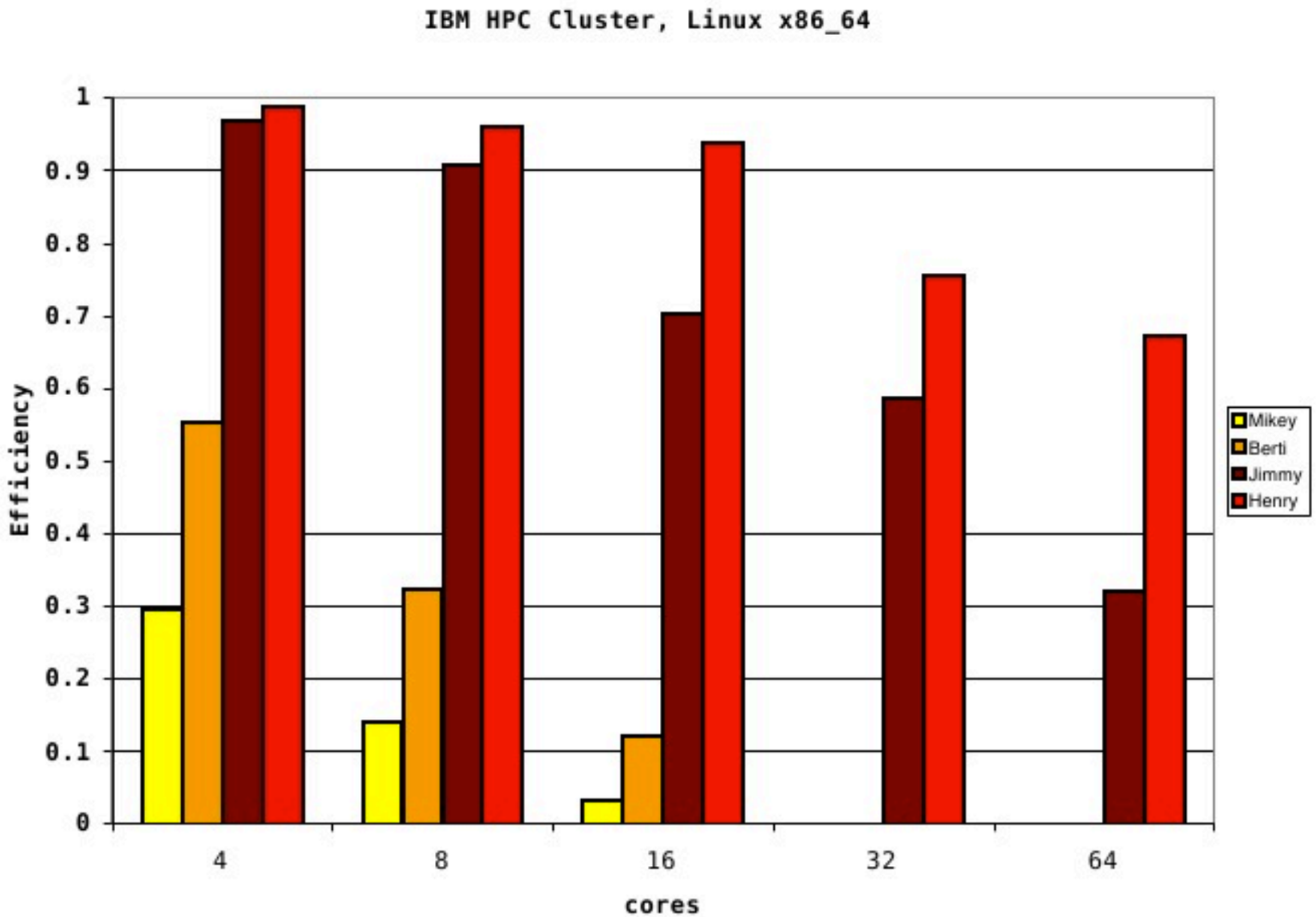
Ott et al. (the Biologists)

Data Set	Mikey	Berti	Jimmy	Henry
Input Size	2.7k × 0.6k	2.7k × 2.3k	15k × 97k	80k× 80k
Heur	5.1 (÷ 1.0)	41.1 (÷ 1.0)	2677 (÷ 1.0)	11708 (÷ 1.0)
BLCS	3.6 (÷ 1.4)	37.3 (÷ 1.1)	3680 (÷ 0.7)	16191 (÷ 0.7)
Sea-16	1.4 (÷ 3.6)	10.8 (÷ 3.8)	1026 (÷ 2.6)	4514 (÷ 2.6)
Sea-8	0.5 (÷ 10.2)	3.8 (÷ 10.8)	368 (÷ 7.3)	1614 (÷ 7.3)
Sea-8 SMP×2	0.3 (÷ 17.0)	3.4 (÷ 12.1)	210 (÷ 12.7)	821 (÷ 14.3)

(Execution times in seconds)

Us. (the Computer Scientists)

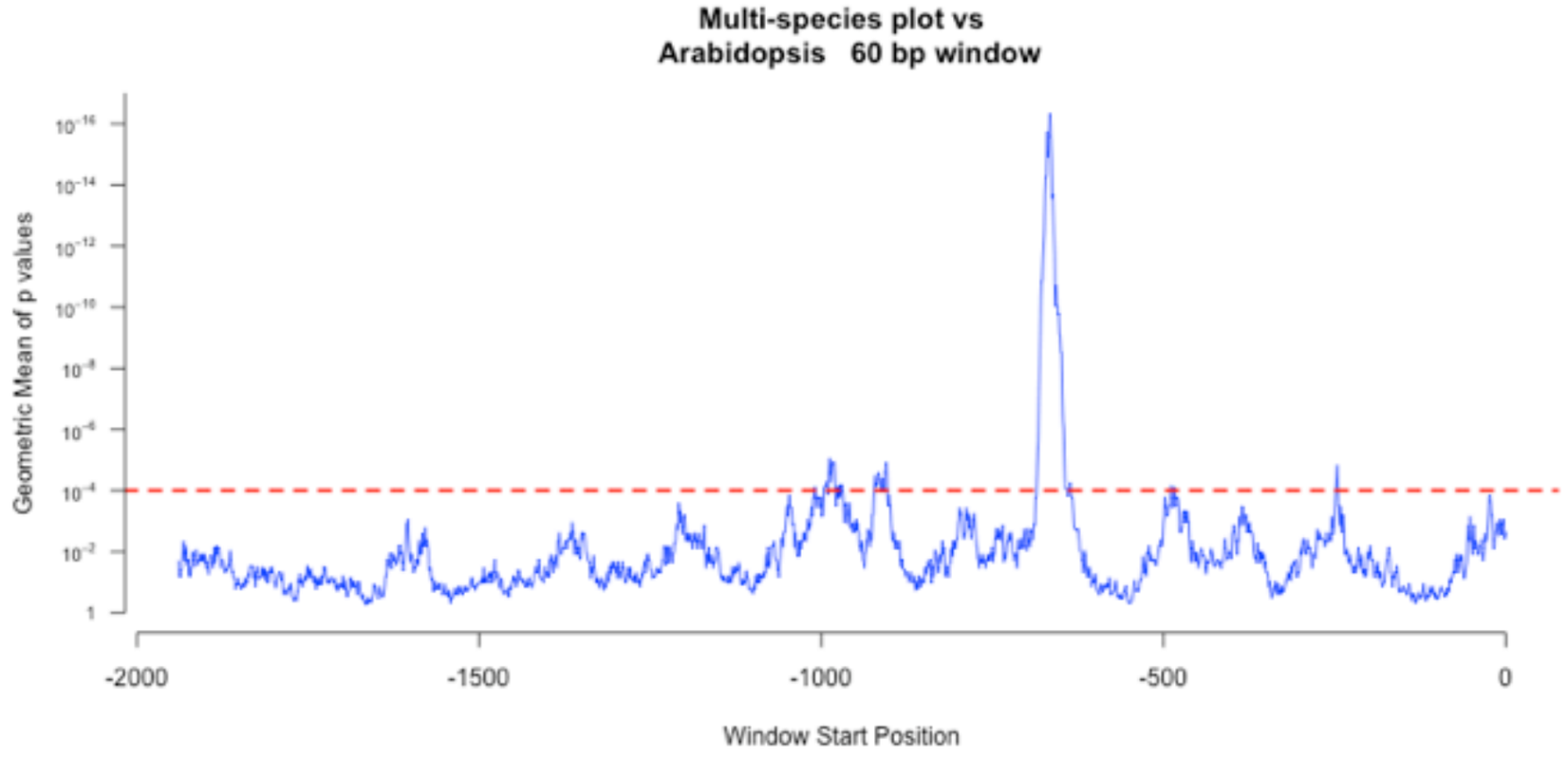
Parallel Computing vs. Alignment Plots



Alignment-plot based Conservation

We compute pairwise alignment plots of the promoter areas of the same gene in multiple species.

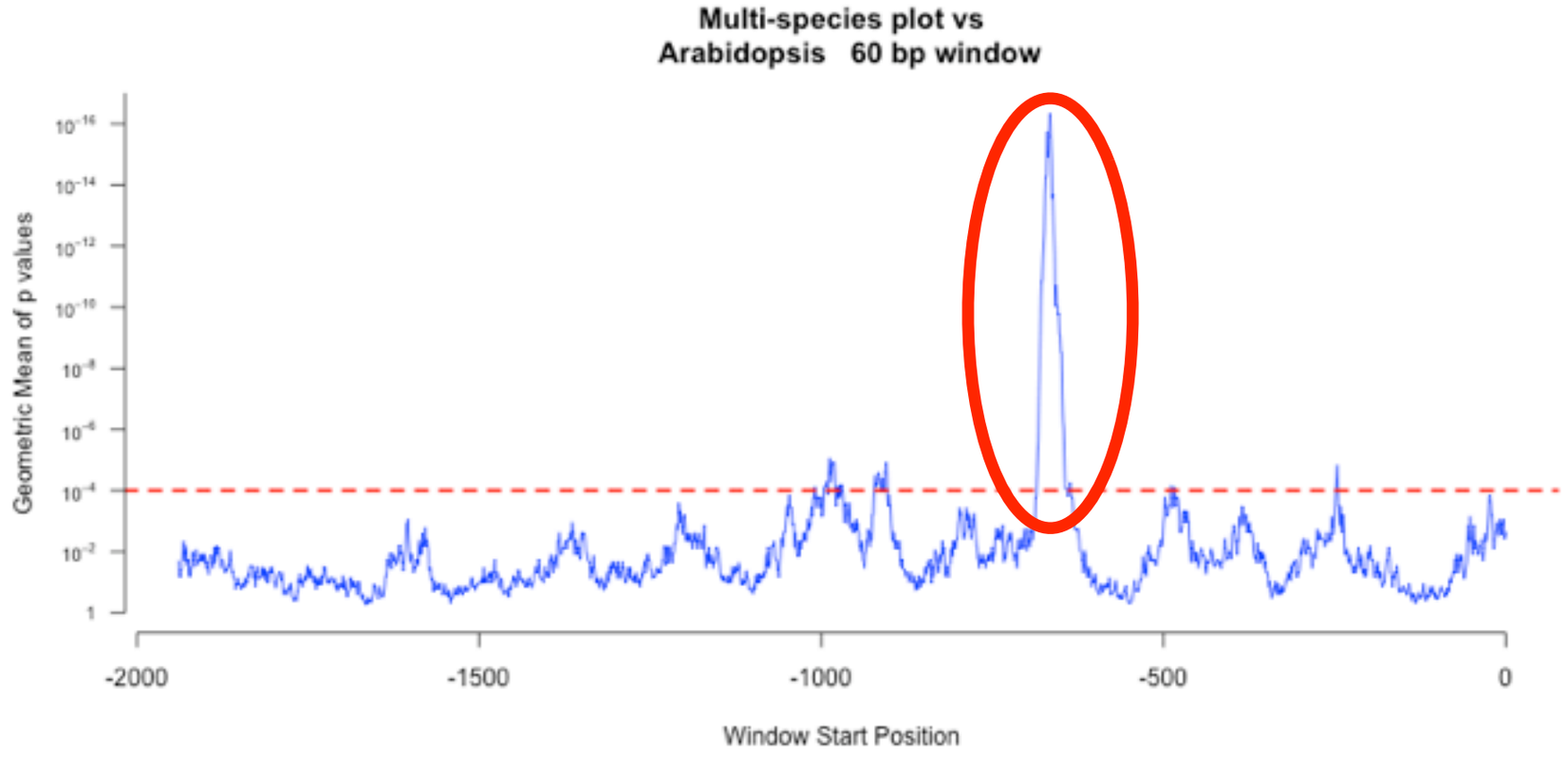
The maximum window score over each column of windows in the plot can be plotted as a profile.



Alignment-plot based Conservation

We compute pairwise alignment plots of the promoter areas of the same gene in multiple species.

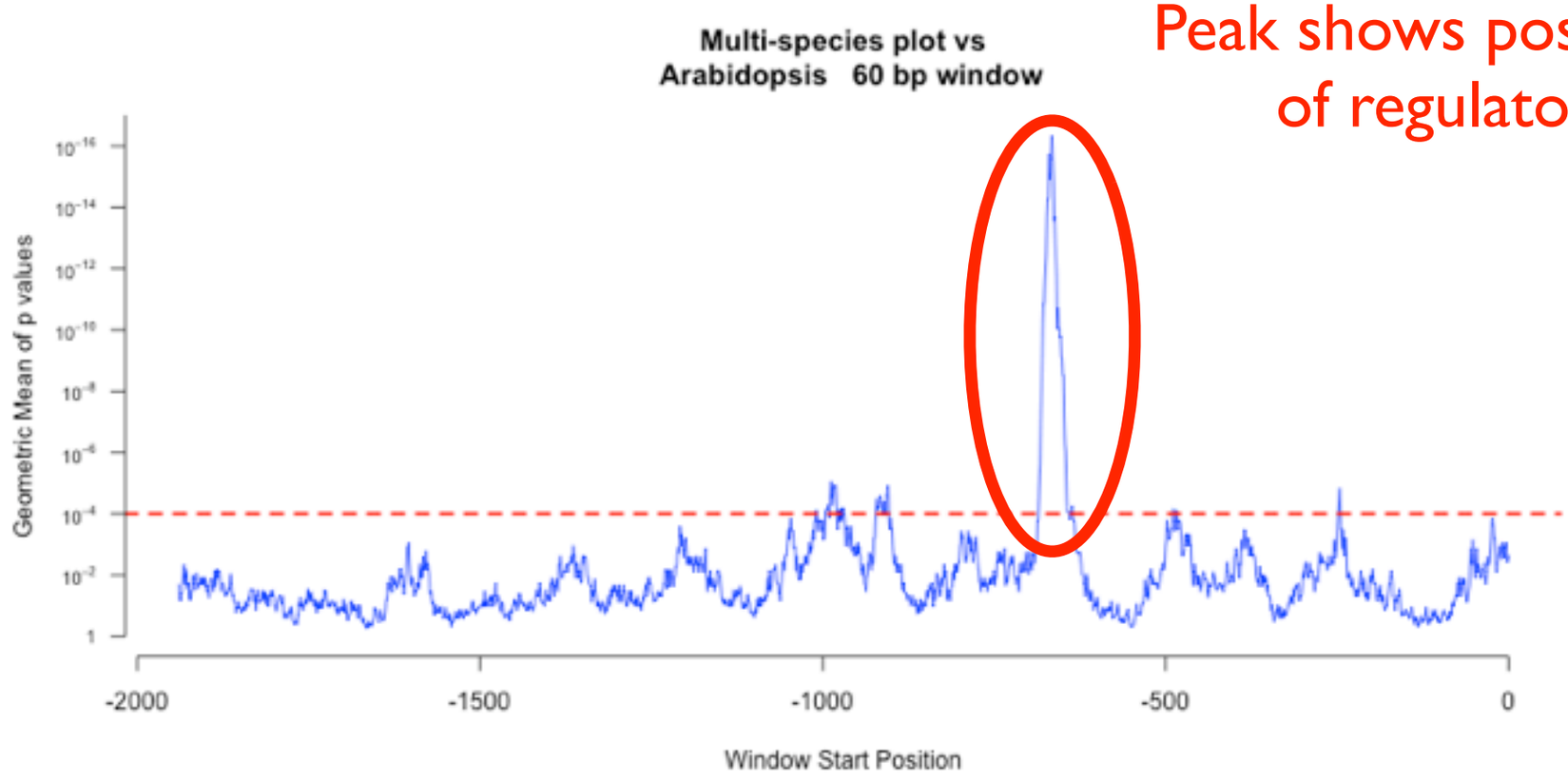
The maximum window score over each column of windows in the plot can be plotted as a profile.



Alignment-plot based Conservation

We compute pairwise alignment plots of the promoter areas of the same gene in multiple species.

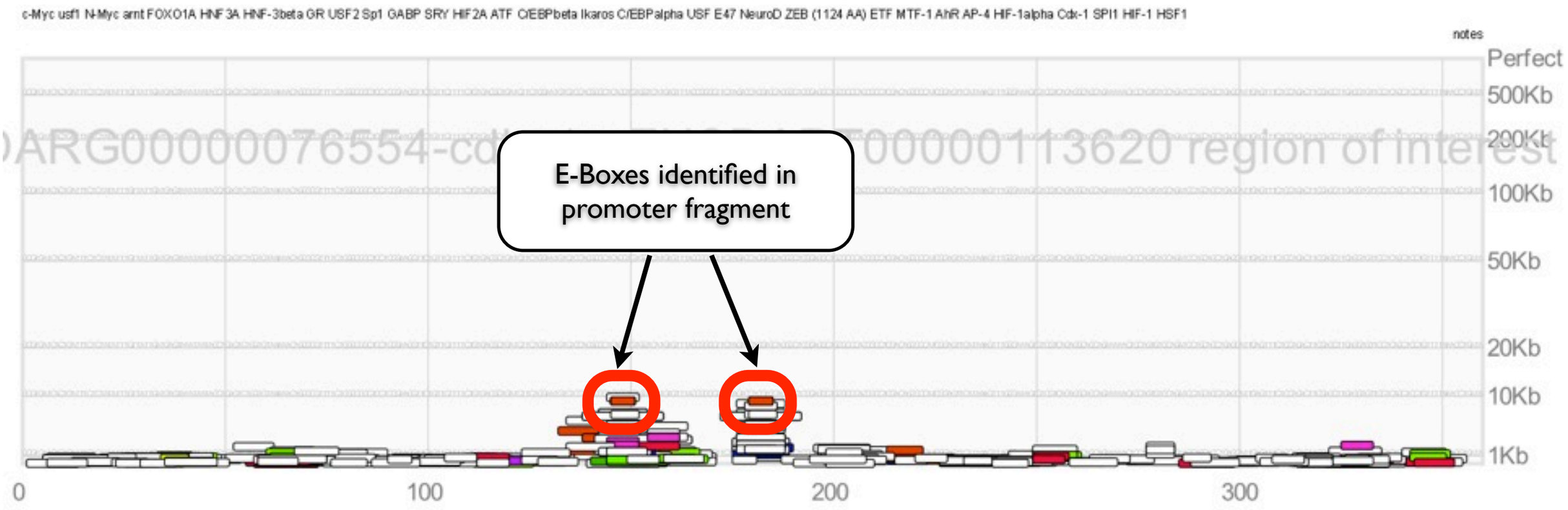
The maximum window score over each column of windows in the plot can be plotted as a profile.



Peak shows possible location of regulatory motifs

Once we know where to look...

BiFa tool outputs possible binding site locations in genomic sequences, scored by statistical significance



Alignment Plots on a GPU

Data Set Input Size	Berti 2712×2305	Jimmy 15097×96901	Henry 80001×80001
Heur	40	2571	11708
Sea-nonoverlap-SSE2 ($k = 1$)	5.8	554	2410
Sea-nonoverlap-GPU ($k = 1$)	5.1	422	1759
Sea-overlap-GPU ($k = 4$)	4.8	381	1596

(Execution times in seconds)

Other Things CS can do.

Data storage and data exchange.

Problems:

- Experimentalists and Theoretical Groups normally are located at different locations (around the world).

- Experimental results need to be documented, stored, and should be searchable in a meaningful way.

- We would like to connect theoretical and practical work (modelling data vs. experimental data)

Issues with Exchanging Biological Data

Confidentiality:

Experimental data (and the knowledge how it was obtained) are the core intellectual capital. Before publication, sharing it is not normally considered a good idea.

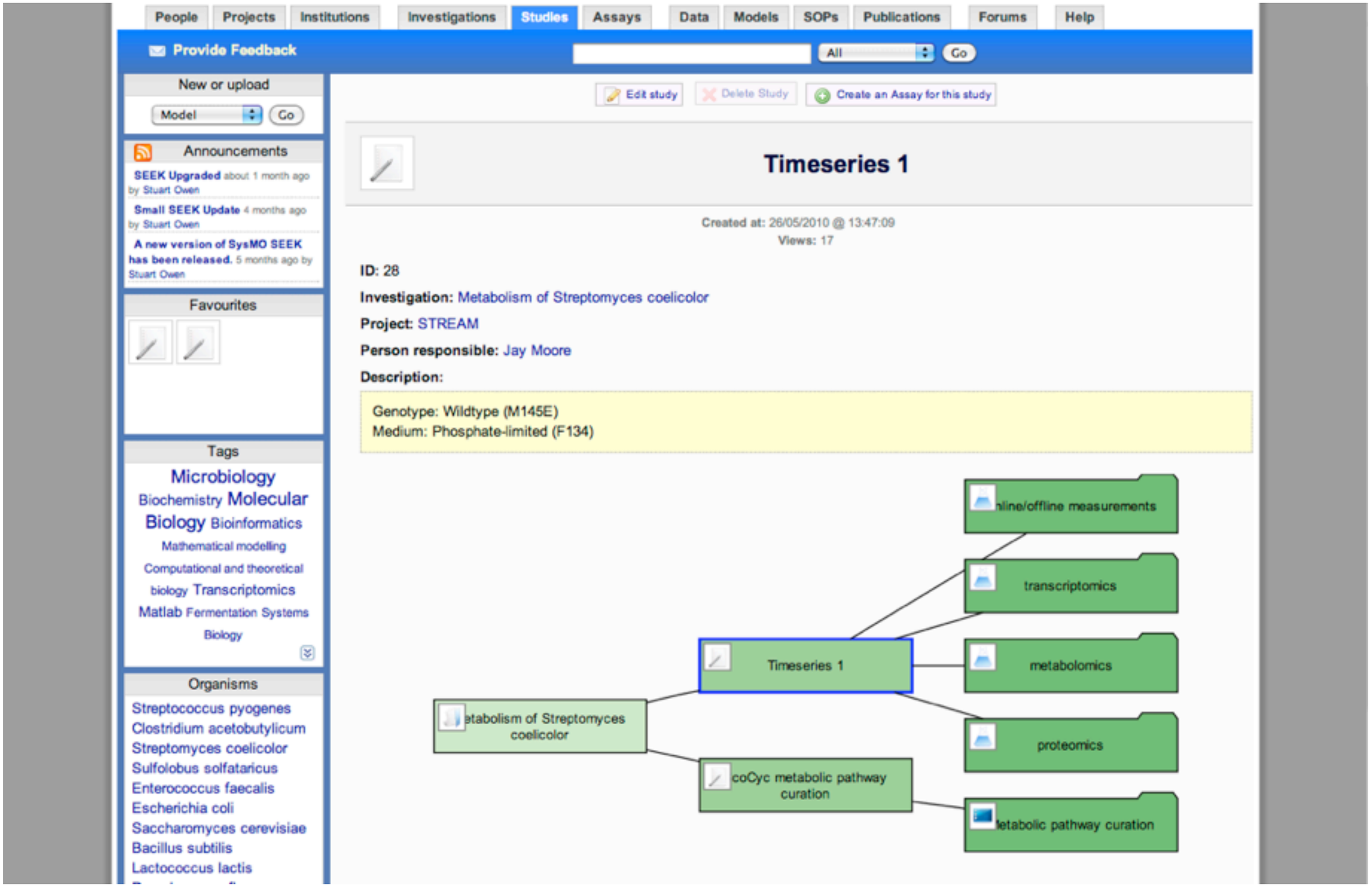
Size:

Data files can be reasonably-sized column data (time-series, ...), or quite large (sequences with annotations, imaging data, ...).

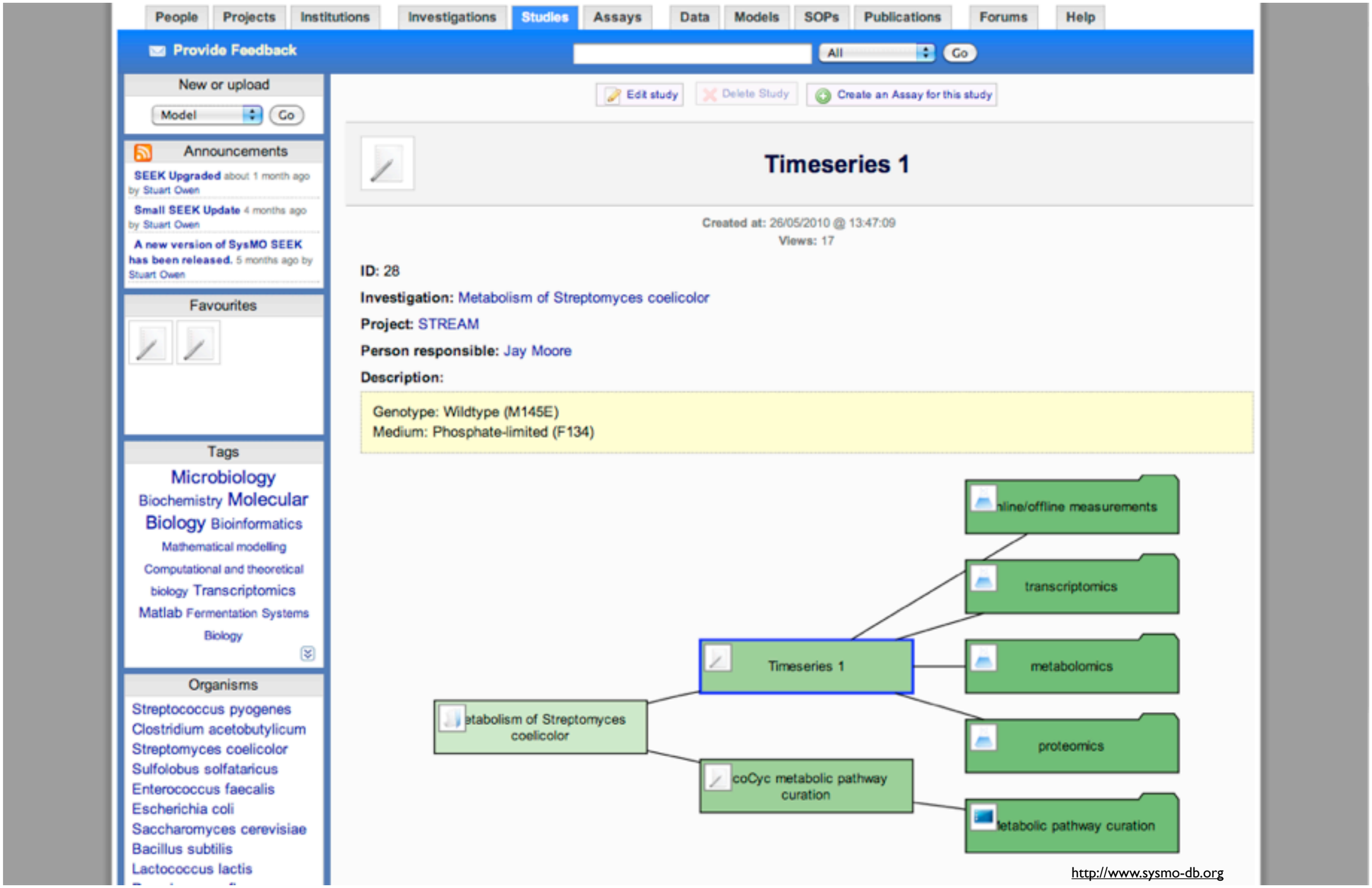
Ease of use:

Experimental biologists rarely know SQL.

Issues with Exchanging Biological Data



Issues with Exchanging Biological Data



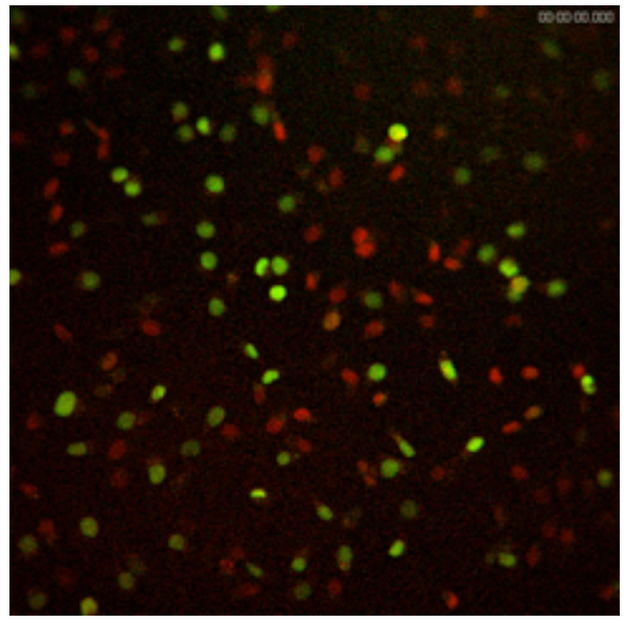
http://www.sysmo-db.org

Yet another problem...

Certain experiments generate imaging data.

Biologists can insert markers into genes to make them glow when expressed in a cell.

We want to track individual cells over time, and extract luminescence information.



Summary

We have seen a few applications of Computer Science in systems biology:

- Sequence comparison and analysis
  [Algorithms, High-performance computing]

- Data storage and management
  [Databases, web-based visualisation]

- Experimental data analysis
  [Image processing, handling large amounts of data, ...]

Outlook

(some things to do to get into Systems Biology)

- Faster algorithms for exact local sequence alignment.

  We could use sequence compression for larger sequences. Theoretical algorithmic improvements are (perhaps) also possible.

- Enable biologists to use new technologies.

  Libraries that use AVX/GPUs, Grid/Cloud computing, ...

- Visualising results.

  Biologists normally use web-applications. Visualising data using Java/Javascript (examples: Chronoscope, Flare, Prefuse, ...).

- Better ways for organising and searching experimental data.

Questions? Discussion?