15-853:Algorithms in the Real World

Computational Biology V
- Sequencing the "Genome"

Thanks to: Dannie Durand for some of the slides.
Various figures borrowed from the web.

Tools of the Trade

Cutting:

Arber, Nathans, and Smith, **Nobel Prize in Medicine** (1978) for "the discovery of restriction enzymes and their application to problems of molecular genetics".

Copying:

Mullis, **Nobel Prize in Chemistry** (1993) for "his invention of the polymerase chain reaction (PCR) method"

Reading: (sequencing)

Gilbert and Sanger, **Nobel Prize in Chemistry** (1980) for "contributions concerning the determination of base sequences in nucleic acids"

Cutting

Cutting:

- Restriction Enzines:
  Cut at particular sites, e.g. ACTTCTAGAT
- Chemical, physical or radiation cuts
  Cut at random locations

Copying

Copying:

Cloning a strand of DNA

- Cosmids: clones sequences up to 40K bps
- BAC, PAC: up to about 200K bps
- YAC (yeast artificial chromosones): up to 1 M

Copying between two specific sites

- PCR (polymerase chain reaction): 500 bps

Cloning (copying fragments)

Isolate DNA

Isolate DNA

fragmentation

Isolate DNA

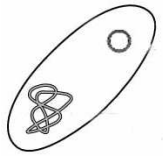
fragmentation

plasmid

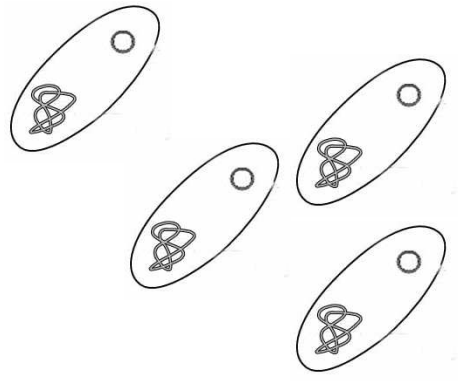
+

insert fragments

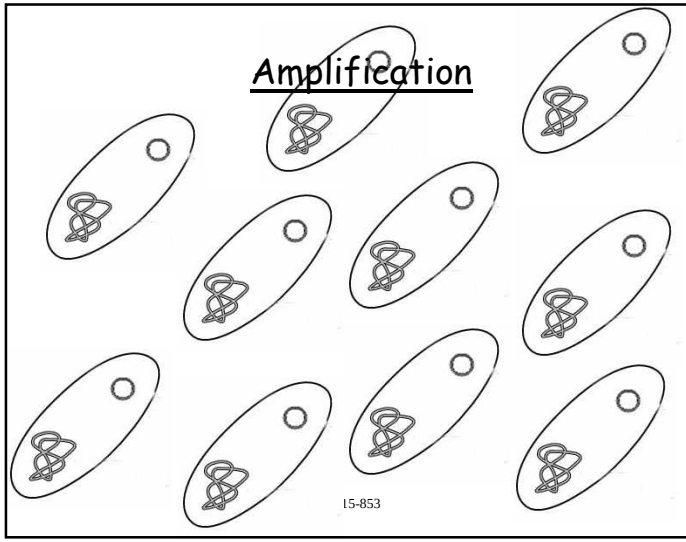
Amplification



Amplification



Amplification



PCR (Polymerase chain reaction)

Select two sequences that appear in the DNA sequence (e.g ATACTTAATG and TCTAAGATAG)

Design two synthetic "**primers**" identical to sequences

REPEAT:

1. **Denature:** Heat DNA to split into two strands
2. **Anneal:** cool and let primers attach
3. **Replicate:** let DNA attach in both directions

Note: cells copy DNA strands character by character

PCR (Polymerase chain reaction)

Reading: sequencing a fragment

Currently too expensive to actually read each bp.

Finding the length is cheap.

- The speed of a fragment in a gel when an electric charge is applied is proportional to its length (DNA has slight negative charge at one end).

Lengths are what are **used in Forensic DNA** analysis and for DNA "fingerprints"

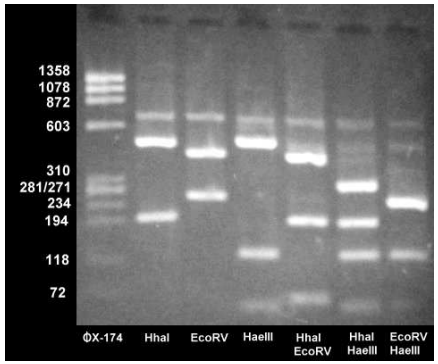
Gilbert and Sanger got the Nobel Prize for figuring out how to **use lengths to "read" a DNA strand** from one end.

Currently only good for about **500** bp.

Forensic DNA Analysis

For the two samples, and some "control" DNA

1. Copy using PCR if sample is small
2. Use restriction enzines to cut up DNA at particular sites (e.g. AATGATGGA)
3. Tag DNA with radioactive (or florescent) tracer This is a strand that will attach to particular sites of the cut DNA.
4. Put each sample (enzine and DNA sample) on its own track on a gel
5. Apply charge for fixed time
6. Expose film to see pattern of lengths

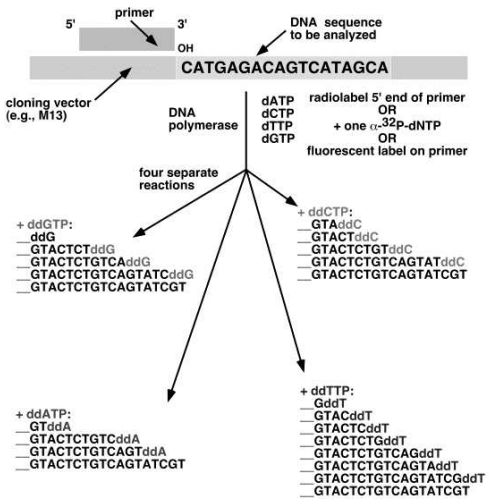


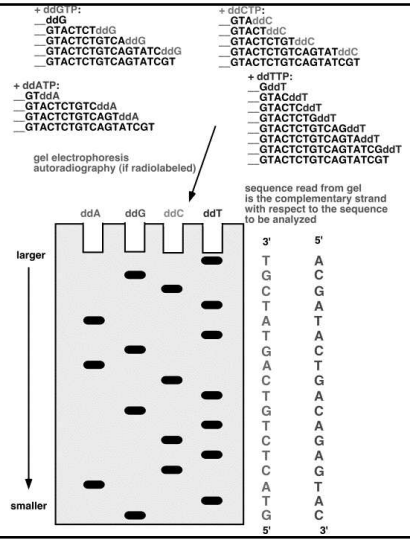
The "fingerprint" of a DNA sample cut by seven restriction enzines.

Reading using lengths

Can use special base-pairs that stop growth: DDC, DDA, DDT, DDG.

Will generate all prefixes that end in A, T, C or G.





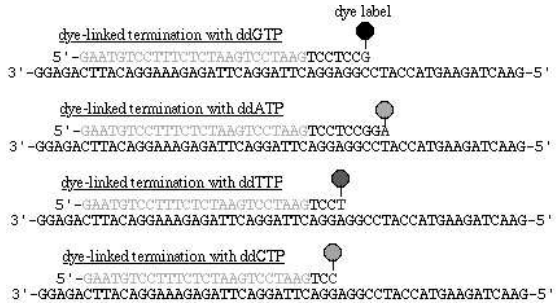
Improvements

Use fluorescent dies on the base pairs and laser to excite the die as it passes a certain point on the gel.

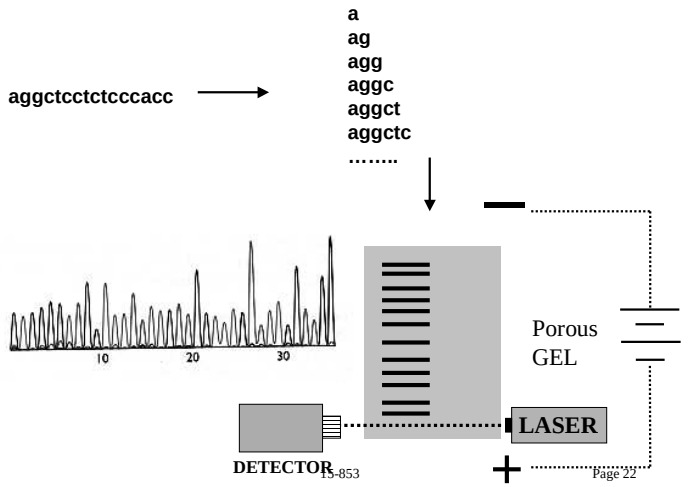
Improvements (1)

4 "test tubes", single track.

Improvements (2)



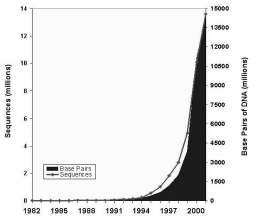
Single "test tube", single track



ABI 3700 sequencer

History of Sequencing

1971 Nobel prize for restriction enzymes
1973 First recombinant DNA
1980 Nobel prize for DNA sequencing
1988 Congress establishes Genbank
1995 First genomic sequence
1998 First multicellular organism
2000 Fly genome
2000 First plant genome
2001 Human genome
2003 Mouse genome



22 million sequences

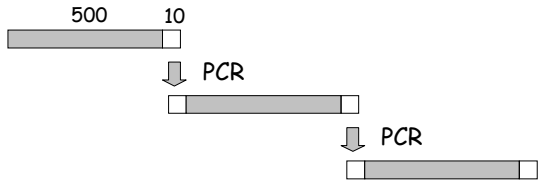
28 billion base pairs

Sequencing the Whole Genome

Problem: we only know how to sequence about 500 bps at a time in the lab.

1. Linear sequencing
2. The shotgun method
3. Hierarchical shotgun method
4. Whole genome and double-barreled shotgun methods

Linear Sequencing



Each step takes too long. Requires "wet" runs.

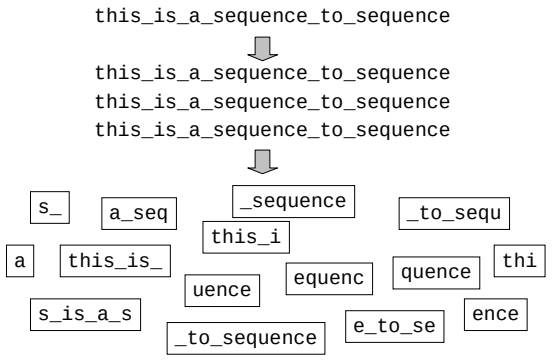
e.g. if each step took 4 hours, sequencing the human genome would take
$4 \times 3 \times 10^9/500$ hours = 3000 years

Also no interesting Computer Science ☺

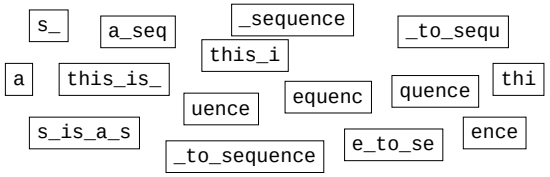
The Shotgun Method

1. Make multiple copies of the sequence.
2. Randomly break sequences into parts (e.g. using radiation or chemicals).
3. Throw away parts that are too small or too large.
4. Read about 500bp from the end of each part
5. Try to put the information together to reconstruct the original sequence

Example

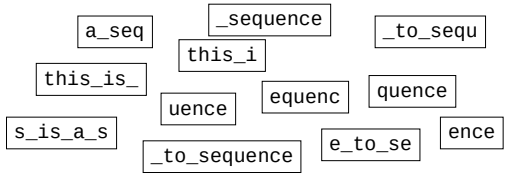


Example



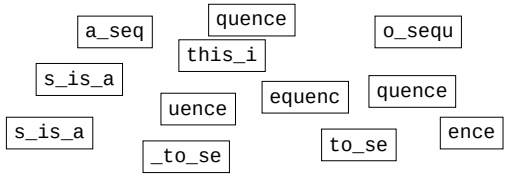
Remove strands that are too short (or too long)

Example



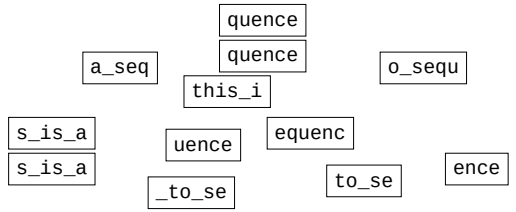
Sequence k characters from each (e.g. 6), from either end.

Example

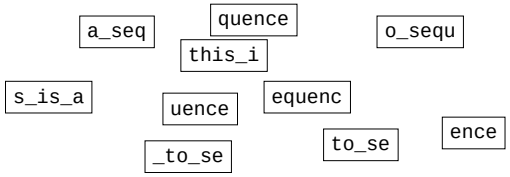


Find overlaps

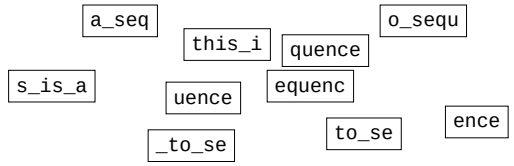
Example



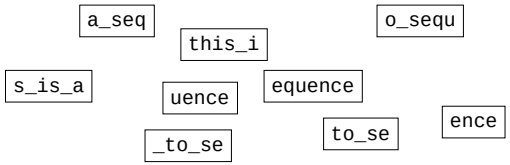
Example



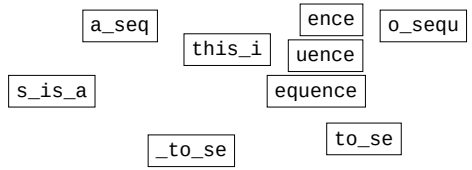
Example



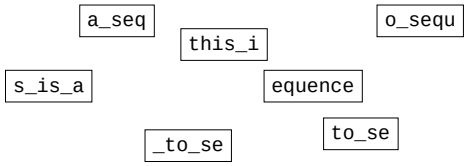
Example



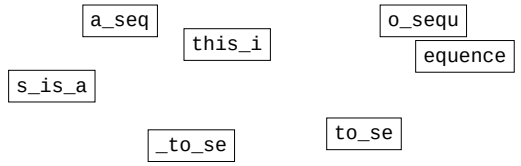
Example



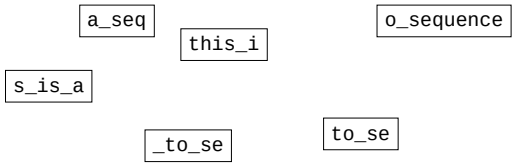
Example



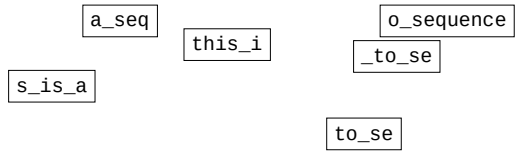
Example



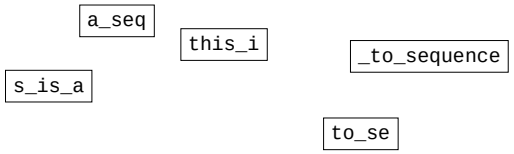
Example



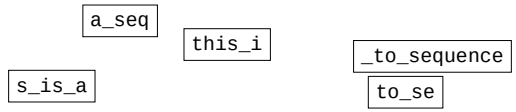
Example



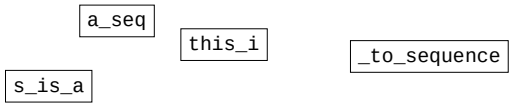
Example



Example



Example



Example

Example

a_seq

_to_sequence

this_is_a

Having a single character overlap might not be enough to assume they overlap.

Example

a_seq this_is_a _to_sequence

Example

a_seq this_is_a _to_sequence

We are left with **gaps**, and unsure matches.

Each covered region (e.g. `this_is_a`) is called a **contig**

Is there a systematic way to find or even define a "best solution"?

The SSP: an attempt

The shortest superstring problem: given a set of strings $s_1, s_2, \ldots, s_n$ find the shortest string S that contains all s_i.

NP-Hard, but can be reduced to TSP and solved approximately (nearly optimally in practice).

Even if easy to solve, are we done?

Our example gives:

```
this_is_a_seq_to_sequence
```

but this is the best we can do given the data.

This problem is caused by repeats.

Other problems?

Problems

In practice the data is noisy.

- Reads have up to a 1% error rate
- Samples could have contaminants
- Fragments can sometimes join up

The reads could be in either direction (front-to-back or back-to-front). Cannot distinguish.

Assembly in Practice

Score all suffix-prefix pairs

gatcgat_ga
attgactactatg

- This can use a variant of the global alignment prob. It is the most expensive step (n^2 scores).

Repeat:

- Select best score and check for consistency
- If score is too low, quit
- If there is a good overlap, merge the two.

Determine consensus:

- We know the ordering among strands, but since matches are approximate, we need to select bps. Can use, e.g., multiple alignment over windows.

Some Programs for Assembly

Phrap
SEQAID
CAP
TIGR
Celera assembler
ARACHNE

After using one of these programs to generate a set of "contigs" with some gaps, one can use the linear method to fill in the gaps (assuming they are small).

atgattagccagtacgtt t tcagcatcccagtacgttatgcac ttagccaga

Suquencing the Whole Genome

Problem: we only know how to sequence about 500 bps at a time in the lab.

1. Linear sequencing
2. The shotgun method
3. Hierarchical shotgun method
4. Whole genome and double-barreled shotgun methods

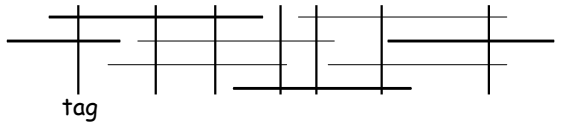
Shotgun on the Whole Genome?

Problems:

- Computationally very expensive
- 50% of genome consist of repeats. Causes major problems.
- Hard to partition work among multiple labs.

Hierarchical Shotgun

1. Generate clone Libraries (100K – 1M per clone)
2. Order the clones by finding "tags" that overlap multiple clones. Use these for ordering.
3. Identify a set of clones that cover the whole length (minimum tiling path)
4. Use shotgun technique on each identified clone
5. Put the results together.



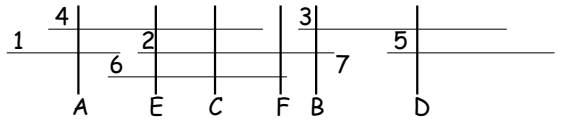
1. Clone Libraries

A "BAC" library will contain sequences of about 200K bps each. These can be cloned using "BAC Vectors" (Bacterial Artificial Chromosome)

A "YAC" library will contain sequences of about 1M bps each. These can be cloned using "YAC Vectors" (Yeast Artificial Chromosome)

These are typically stored at a common site and can be ordered. Many can be purchased from companies.

2. Ordering Clones



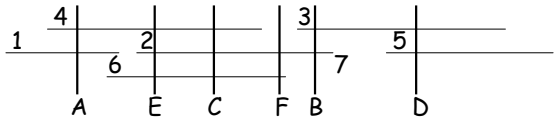
We have the clones, but we don't know their order or how they overlap.

Pick random small sequences that only appear once in one location covered by the library.

These are called STS (Sequence Tagged Sites)

Figure out which clones contain which STSs using PCR (use tag site to start copy…will only copy of the sequence contains the site).

2. Ordering Clones (cont.)



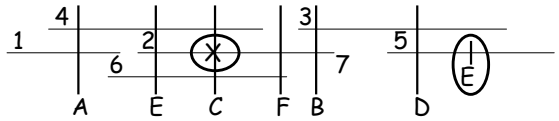
	A	B	C	D	E	F
1	1	0	0	0	0	0
2	0	1	1	0	1	1
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	0	0
6	0	0	1	0	1	1

Goal: Reorder the columns so that all the 1s in each row are contiguous.

Can be done in O(n) time, where n is the number of entries in the array.

But!!!, what about **errrors**?

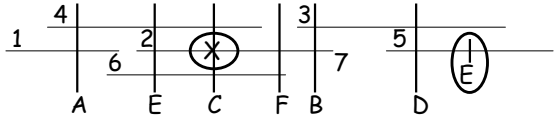
2. Ordering Clones (cont.)



	A	B	C	D	E	F
1	1	0	0	0	0	0
2	0	1	**0**	0	1	1
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	**1**	0
6	0	0	1	0	1	1

	A	E	C	F	B	D
1	1	0	0	0	0	0
2	0	1	**0**	1	1	0
3	0	0	0	0	1	1
4	1	1	1	0	0	0
5	0	**1**	0	0	0	1
6	0	1	1	1	0	0

2. Ordering Clones (cont.)



Find ordering that minimizes the number of zero-one and one-zero transitions (i.e. errors).

This is NP-hard, but can be posed as a Traveling Salesman Problem (TSP).

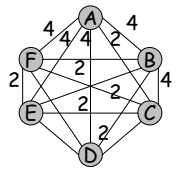
Any ideas?

2. Ordering Clones (cont.)

Create graph with one vertex per STS.

Edge weights = hamming distance (number of bits that differ).

	A	B	C	D	E	F
1	1	0	0	0	0	0
2	0	1	**0**	0	1	1
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	**1**	0
6	0	0	1	0	1	1

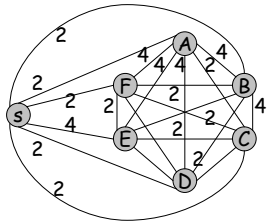


2. Ordering Clones (cont.)

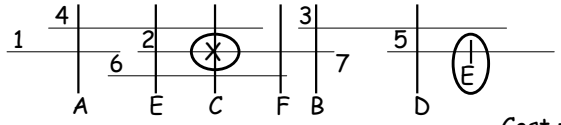
Add in source (s) node with weights equal to number of 1s in each row.

Solve TSP. Answer gives min number of transitions.

	A	B	C	D	E	F
1	1	0	0	0	0	0
2	0	1	**0**	0	1	1
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	**1**	0
6	0	0	1	0	1	1

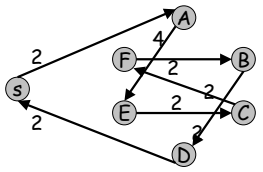


2. Ordering Clones (cont.)

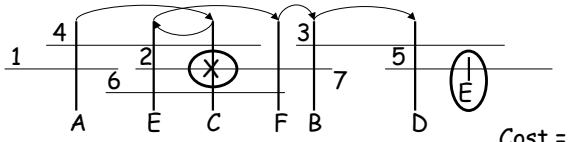


Cost = 16

	A	B	C	D	E	F
1	1	0	0	0	0	0
2	0	1	**0**	0	1	1
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	**1**	0
6	0	0	1	0	1	1

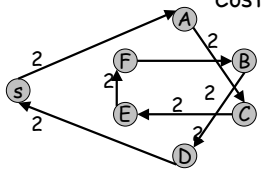


2. Ordering Clones (cont.)



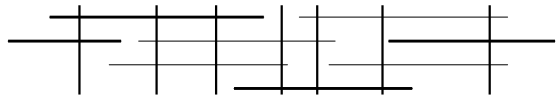
Cost = 14

	A	B	C	D	E	F
1	1	0	0	0	0	0
2	0	1	**0**	0	1	1
3	0	1	0	1	0	0
4	1	0	1	0	1	0
5	0	0	0	1	**1**	0
6	0	0	1	0	1	1



The "wrong" answer has smaller cost

3. Find "Minimum Tiling Path"



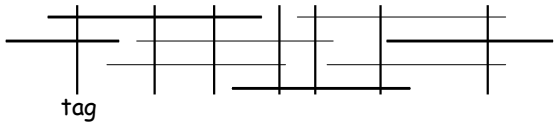
Minimum Tiling Path: Find a set of clones that cover the whole length and for which the total number of bps is minimized.

Can be posed as a shortest path problem.

Any ideas?

Hierarchical Shotgun (revisited)

1. Generate clone Libraries (100K – 1M per clone)
2. Order the clones by finding "tags" that overlap multiple clones. Use these for ordering.
3. Identify a set of clones that cover the whole length (minimum tiling path)
4. Use shotgun technique on each identified clone
5. Put the results together.

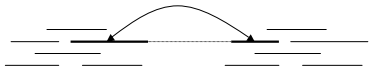


Celera's Method

Whole genome shotgun:

Use shotgun method on whole genome.

Use **double-barreled** approach: some sequences of known length (e.g. 2-5K) are sequenced at both ends. These can be used to bridge across repeats.



In practice they used some mapping (hierarchical) data from the NIST effort, which was freely available. This was needed to deal with long repeats.