Approaches to Repeat Finding

Beth Skwarecki

Cornell Genomics Forum, 2005-03-18

Why is repeat detection important?

- annotating repeat sequences
- Repeats interfere with assembly
- Repeats interfere with gene annotation, blast, etc.

What kinds of repeats are we looking for?

• **Simple Repeats** - Duplications of simple sets of DNA bases (typically 1-5bp) such as A, CA, CGG etc.

• **Tandem Repeats** - Typically found at the centromeres and telomeres of chromosomes, these are duplications of more complex 100-200 base sequences.

• **Segmental Duplications** - Large blocks of 10-300 kilobases which are that have been copied to another region of the genome.

• **Interspersed Repeats**

Processed Pseudogenes, Retrotranscripts, SINES - Non-functional copies of RNA genes which have been reintegrated into the genome with the assitance of a reverse transcriptase.

DNA Transposons

Retrovirus Retrotransposons

Non-Retrovirus Retrotransposons (LINES)

• **Low complexity sequence**

http://repeatmasker.org

Programs for repeat finding

- RepeatMasker
- REPuter
- RepeatFinder
- FORRepeats
- RECON

RepeatMasker

- Requires database of known repeats (GIRI)
- Uses cross_match (MaskerAid uses a similar approach with BLAST)
- Detects low complexity sequence in addition to repeats

	# of repeats	total bp
primates	563	664160
rodents	466	487006
other mammals	347	243730
other vertebrates	52	53994
Drosophila	65	167423
Arabidopsis	98	275516
grasses	27	67789

RepeatMasker-Performance

- Speed depends on size of database
- Linear time with sequence length
- Somewhat faster for sequences < 10kb

RepeatMasker-Pros and Cons

- False positives are rare (1/4440 in one test)
- Detects low-complexity sequence that may not be repetitive
- Don't rely on RM results for EST searches, gene prediction, primer design

REPuter

1. Use a suffix tree to find exact repeats (“seeds”)
2. Extend seeds with Hamming distance and edit (Levenshtein) distance to find approximate repeats

Suffix Trees

T1 = mississippi

T2 = ississippi

T3 = ssissippi

T4 = sissippi

T5 = issippi

T6 = ssippi

T7 = sippi

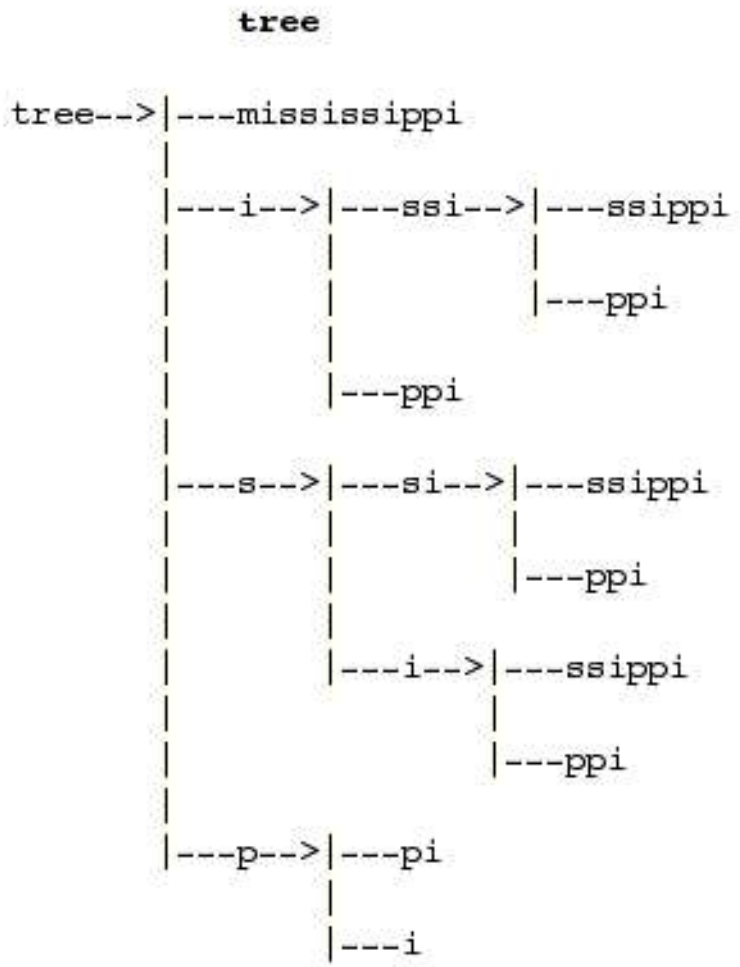
T8 = ippi

T9 = ppi

T10 = pi

T11 = i

T12 = <empty>



http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Tree/Suffix/

Hamming and Edit distance

Hamming distance counts substitutions in strings of the same length.

LETTER	distance = 3
LADDER	(3 substitutions)

Edit distance (aka Levenshtein distance) also counts insertions and deletions

LETTER	distance = 2
LATER	(1 substitution, 1 deletion)

REPuter - Performance

- Step 1 (exact repeats) in linear space and time: O(n+z)
- Step 2 (Hamming distance): O(n+zk)
- Step 2 (Edit distance): O(n+zk^3)

k = error parameter

z = number of seeds

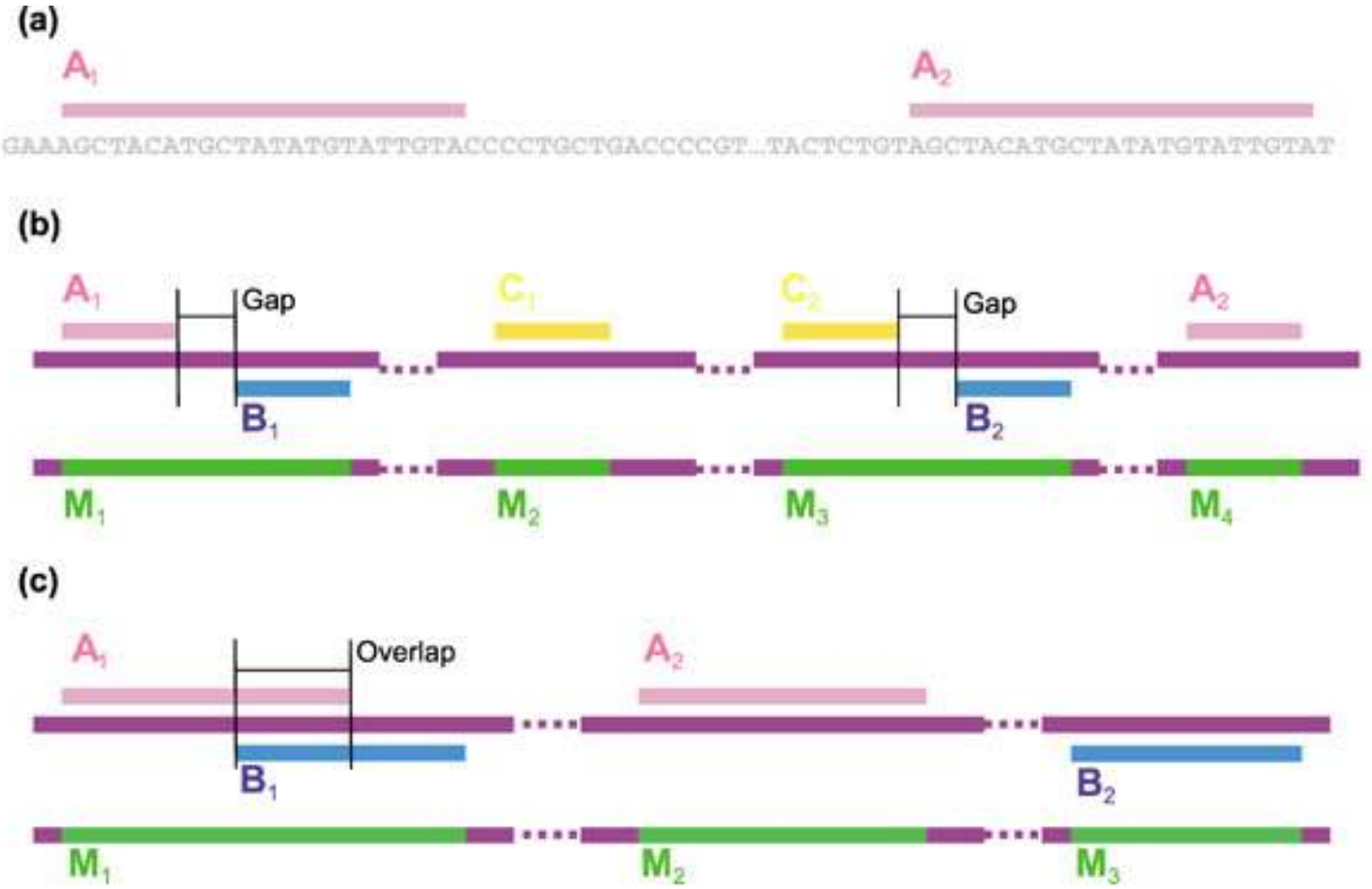
REPuter – Pros and Cons

- Detects exact and degenerate repeats
- Detects palindromic as well as direct repeats
- Detects all repeats within parameters – not heuristic

RepeatFinder

1. Identify exact repeats with RepeatMasker or REPuter
2. Merge repeats that overlap or are very close
3. Cluster repeats into families
4. BLAST to determine related repeats that are not exact

RepeatFinder – Merging and clustering



RepeatFinder – Performance

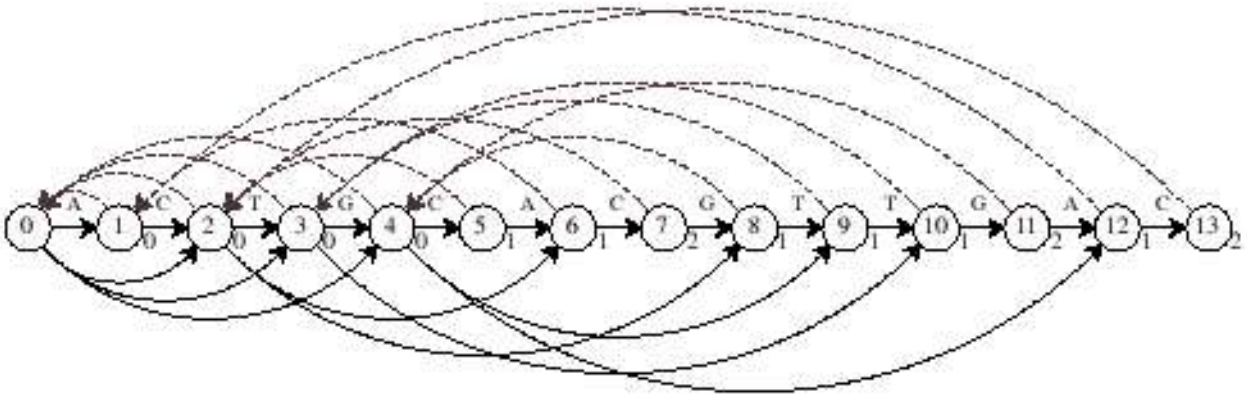
- BLAST step is 80% of running time, unnecessary if Step 1 finds approximate repeats
- Merging step takes minutes for microbial genomes, 2 days for rice
- Several gigs of memory needed for large eukaryotic chromosomes (to store suffix tree in Step 1)

RepeatFinder – Pros and Cons

- Finds families of related repeats
- Approximate repeats in Step 1 eliminate need for Step 4

FORRepeats

1. Find exact repeats using heuristic “Factor Oracle”
2. Extend exact repeats using Hamming distance



FORRepeats - Performance

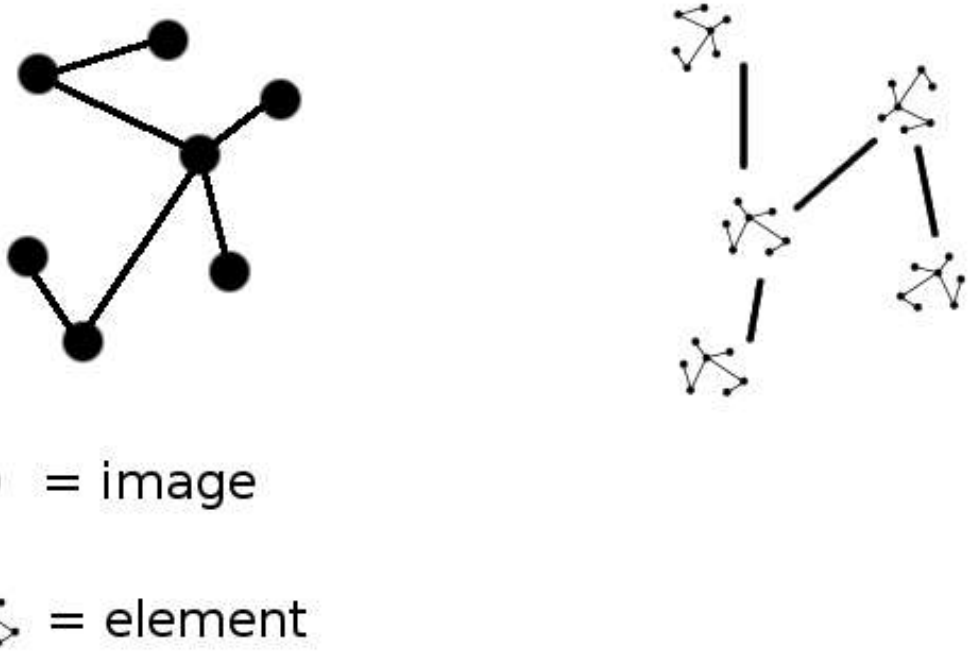
- Most accurate with longer repeats
- 10.5x memory for factor oracle compared to 12x for suffix tree
- Much faster than BLAST
- 2x as fast as REPuter, but not guaranteed to match everything.

FORRepeats - Pros and Cons

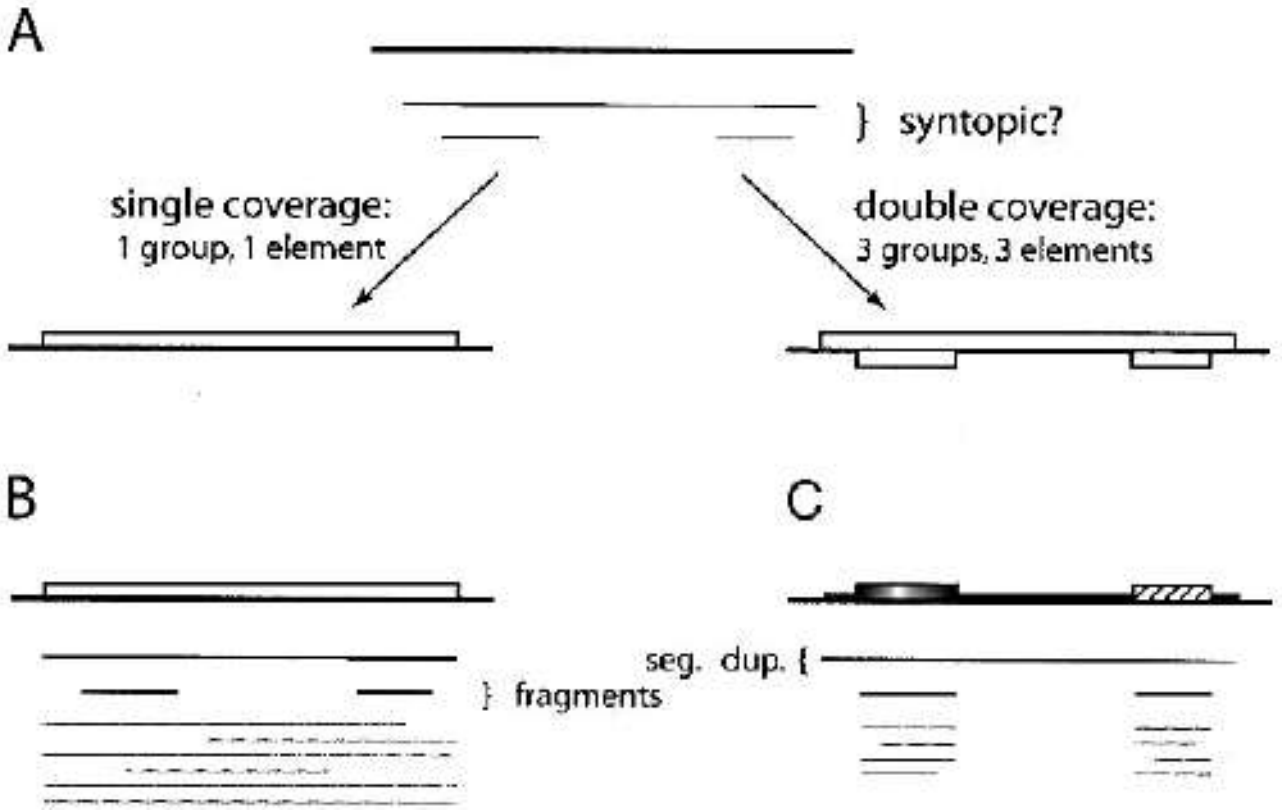
- Similar approach to REPuter, but Step 1 is faster
- Heuristic: will not detect all repeats

RECON

1. Find repeats with BLAST
2. Determine similarity using endpoints
3. Group images into elements, and elements into families

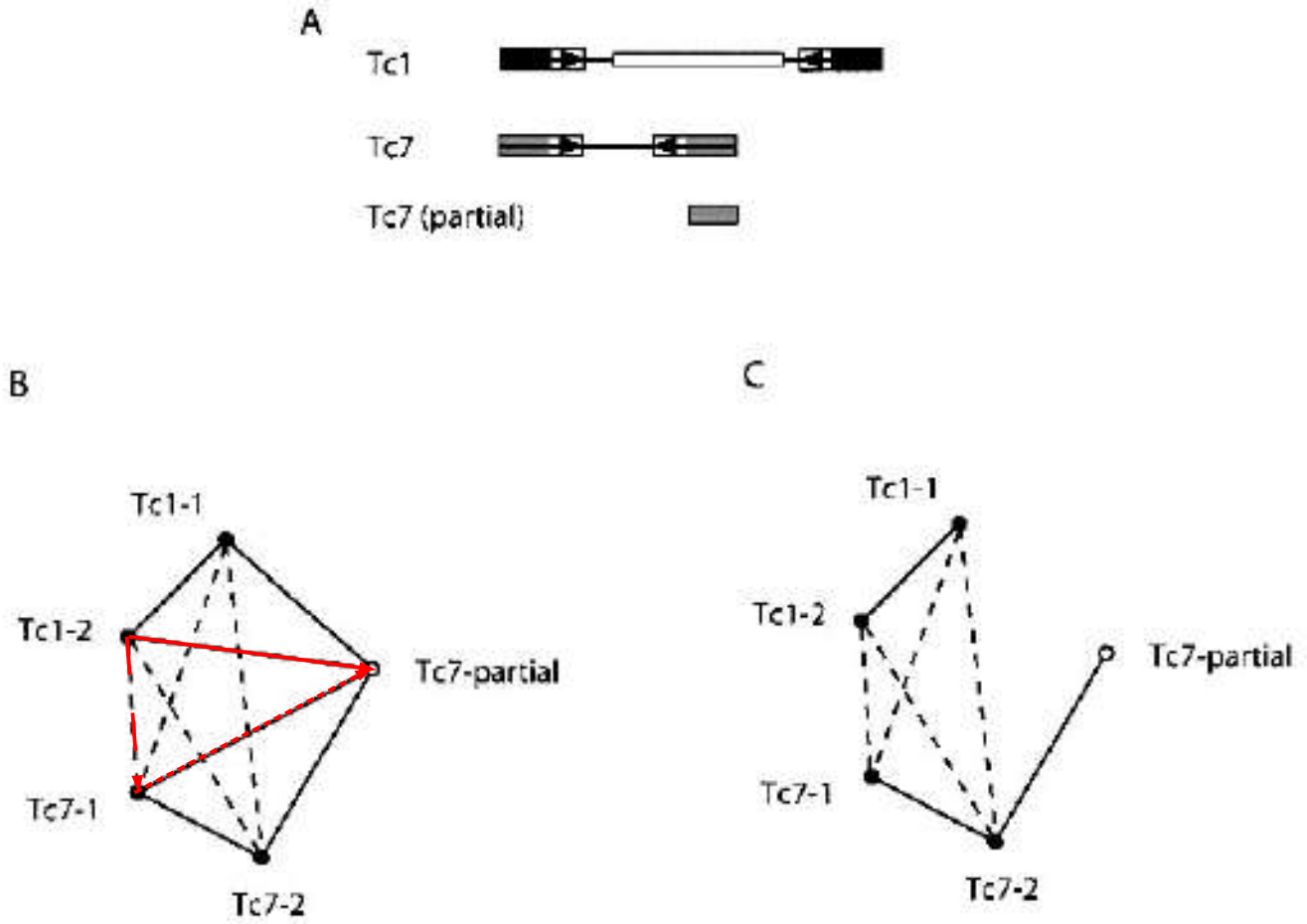


RECON


A
} syntopic?
single coverage:
1 group, 1 element
double coverage:
3 groups, 3 elements
B
C
seg. dup. {
} fragments

RECON

- Deletions can lead to incorrect family grouping



RECON - Performance

- Performance depends on repeat complexity and composition, not genome size

RECON - Pros and Cons

- Heuristic method
- Uses multiple alignment instead of pairwise
- Endpoints identify likely repeat boundaries
- Underclusters
- Fragments some families
- Splits elements when a partial element is common
- Used for initial analysis, like building RepeatMasker libraries

Summary

- **RepeatMasker** – knows your genome
- **REPuter** – finds and extends; accurate
- **RepeatFinder** – merges to find families
- **FORRepeats** – heuristic, somewhat faster than REPuter
- **RECON** – finds families, heuristic

References

- Smit, AFA & Green, P. **RepeatMasker** at http://repeatmasker.org
- Kurtz, S., Choudhuri, J. V., Ohlebusch, E., Schleiermacher, C., Stoye, J., Giegerich, R. **REPuter: the manifold applications of repeat analysis on a genomic scale**. *Nucleic Acids Research* 2001, 29:4633-4642
- Volfovsky, N., Haas, B. J., Salzberg, S. L. **A clustering method for repeat analysis in DNA sequences**. *Genome Biology* 2001, 2:1-11
- Lefebvre, A., Lecroq, T., Dauchel, H., Alexandre, J. **FORRepeats: detects repeats on entire chromosomes and between genomes**. *Bioinformatics* 2003, 19:319-326
- Bao, Z., and Eddy, S. **Automated *de novo* identification of repeat sequence families in sequenced genomes**. *Genome Research* 2002, 12:1269-1276