"Comparative Genome Assembly" by Pop et al.

Seminar CS2431

Vladimir Yanovsky

March 5, 2008

De-Novo Assembly

- Two types of assemblers: hierarchical shotgun (BAC-to-BAC) is more expensive than WGS.
- WGS is more error prone than "wet lab - assisted" BAC-to-BAC assembly.
- All follow *overlap-layout-consensus* paradigm.

Overlap-Layout-Consensus

All previous algorithms perform roughly the following steps:

1. Find pairwise *overlaps* of all reads – can take $O(n^2)$ or better.
2. Build a graph with vertices representing the reads and edges representing the overlaps.
3. *Layout* – find a "good" path or set of paths in the graph building *contigs* – sequences longer than reads but way shorter than the size of the genome.
4. *Consensus* – make contigs agree.
5. *Scaffolding* – using matepairs info.
6. New de-novo assemblers use information from the *overlap* and *layout* stages in the *scaffolding* stage and perform iteratively.

Resequencing

Genome assemblers sometimes help answering such basic questions as how many chromosomes the organism has. If not, having a similar organism already assembled is likely to help:

- May want to sequence several strains of similar bacteria.
- Or sequence another organism of the same species
- Or sequencing another patient in medical settings - must be fast and cheap.
- Arachne not suitable for NGS - discards reads which are 50 bases after trimming. Likely other assemblers fare as bad.
- WGS for NGS $\Longrightarrow$ resequencing can be the only way to go.
- How can we use this obvious idea in an automated way?

Amos-Cmp Bird's eye view

Overlap stage takes the most time of the three stages.
AMOS-Cmp goes to extreme - no overlap stage at all.

1. *Read alignment* - use *MUMmer*. Ambiguous placements – repeats – resolved later.
2. Repeat resolution – use mate pairs (their existence or distance between). If still not decided – choose randomly.
3. *Layout* – takes care of indels and rearrangements.
4. *Consensus Generation* – find consensus of group of reads covering a subsequence of the reference genome. Use iterative multiple alignment.
5. *Scaffolding* – same as before but now we don't have access to the alignment information.

Insertions in the target

Two contigs will be created. **B** will only be mapped at the *scaffolding* stage.

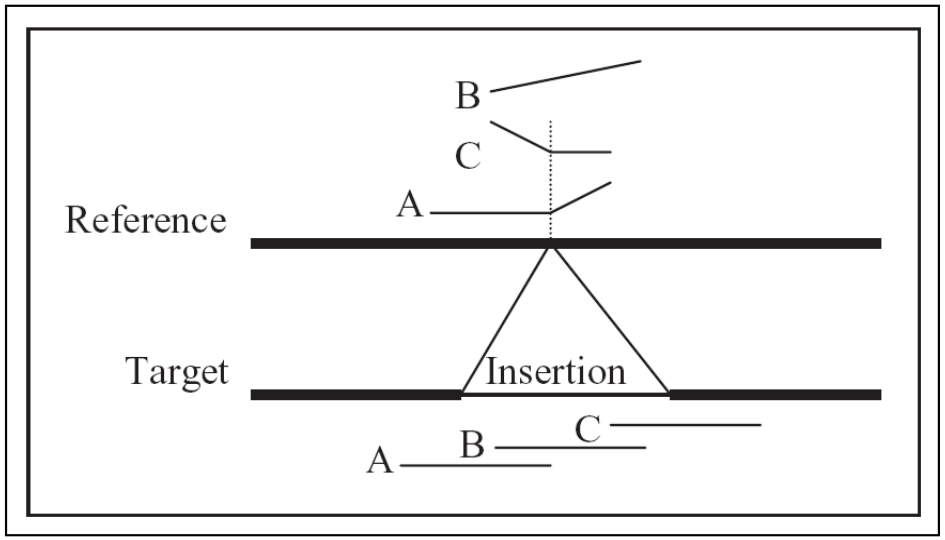

Figure: Mapping reads to the reference genome when the target genome contains an insertion. Slanted lines depict no match.

Insertions in the target - shorter than a read

Another easy case for AMOS-Cmp:

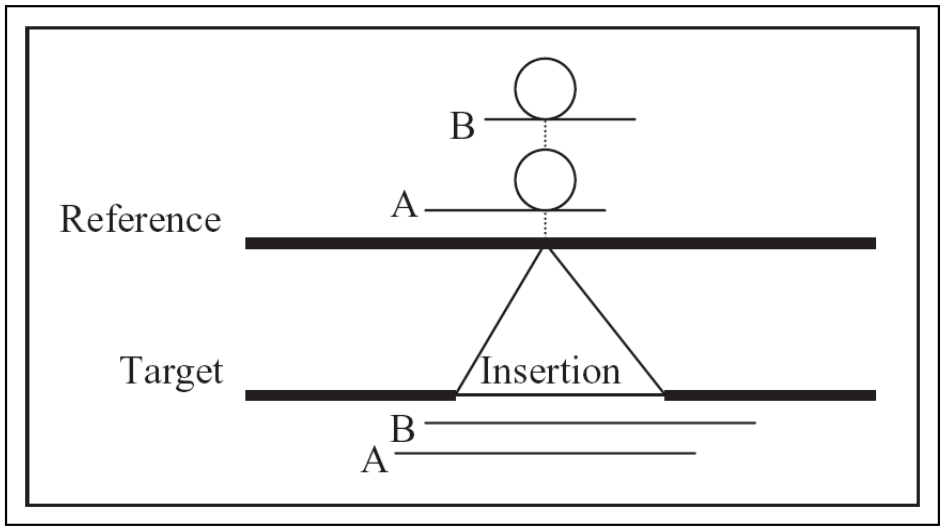

Figure: The insertion in the target shorter than a single read. The "bubbles" identify the portions of the two reads that do not align to the reference.

Insertions in the reference

We have a clear "signature" here as well:

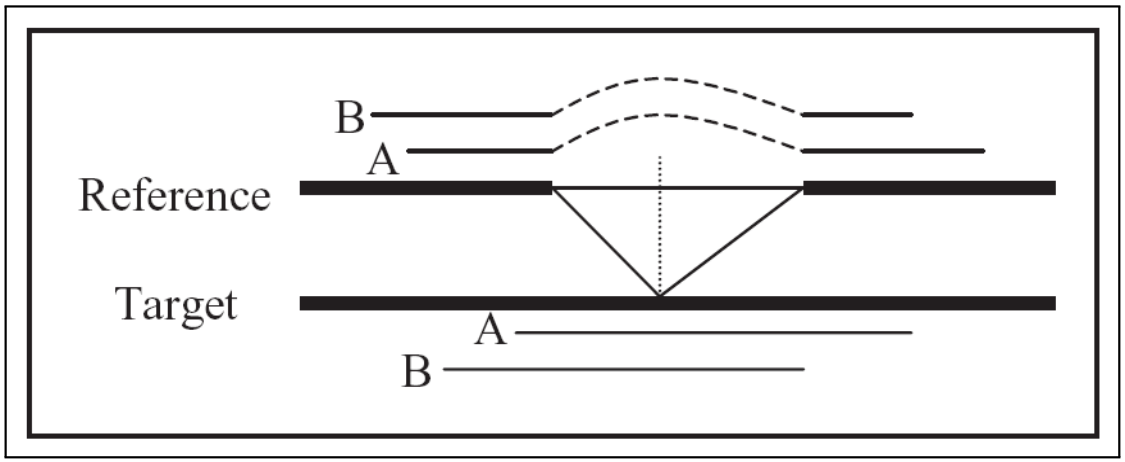

Figure: Insertion into the reference. Such an alignment of reads to the reference indicates the presence of the insertion. Dashed lines indicate the 'stretch' of the reads needed to align to the reference.

Rearrangement

Scaffolding will (hopefully) help the assembler: We have a clear "signature" here as well:

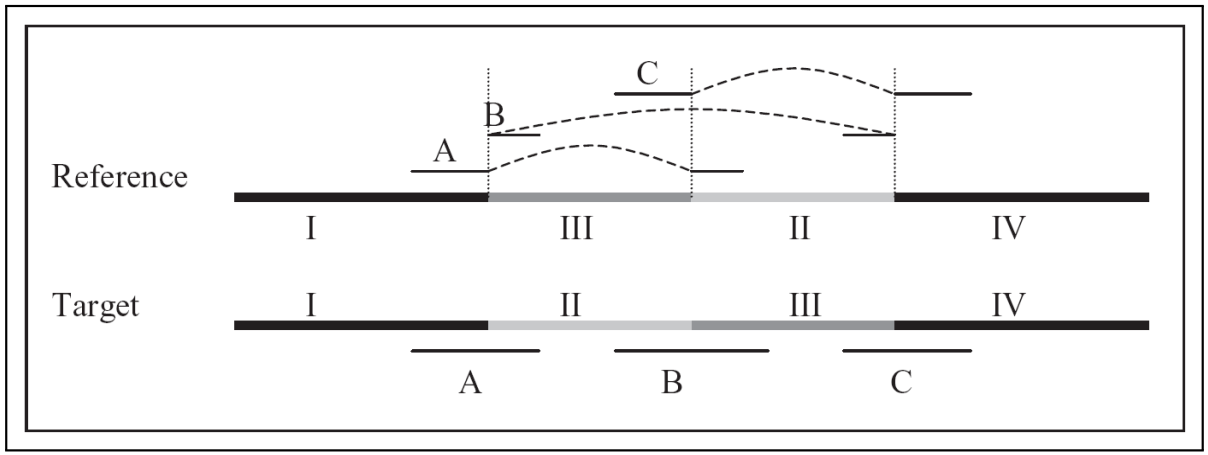

Figure: Signature of rearrangement – insertion into reference. AMOS-Cmp creates a single contig spanning sections 1 and 2 and another contig from sections 3 and 4

Divergent DNA

Looks a little bit similar to insertion into target. But not "identical" as the authors claim.

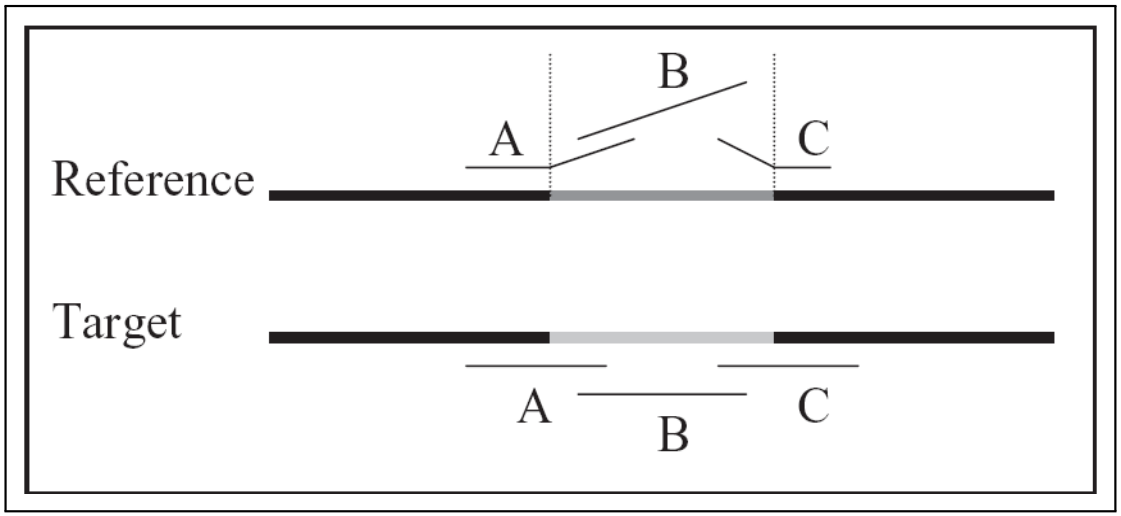

Figure: Divergent DNA. Two contigs are created by the assembler. Again, rely on scaffolder.

Distinguishing between sequencing error and true polymorphisms

- Trim reads using *lucy* to remove regions likely to have errors.
- Breakpoint – a problematic point. Must decide if it is an error.
- Decide by voting using that the errors are independent and can happen everywhere.

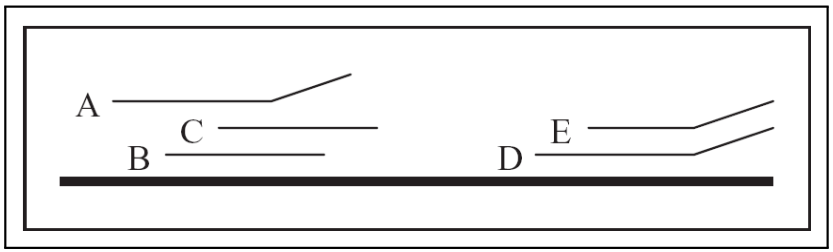

Figure: Detecting errors. Read A is probably incorrect while reads D and E indicate polymorphism.

Flanking Ends

Allow overlap between adjacent alignments to the reference.

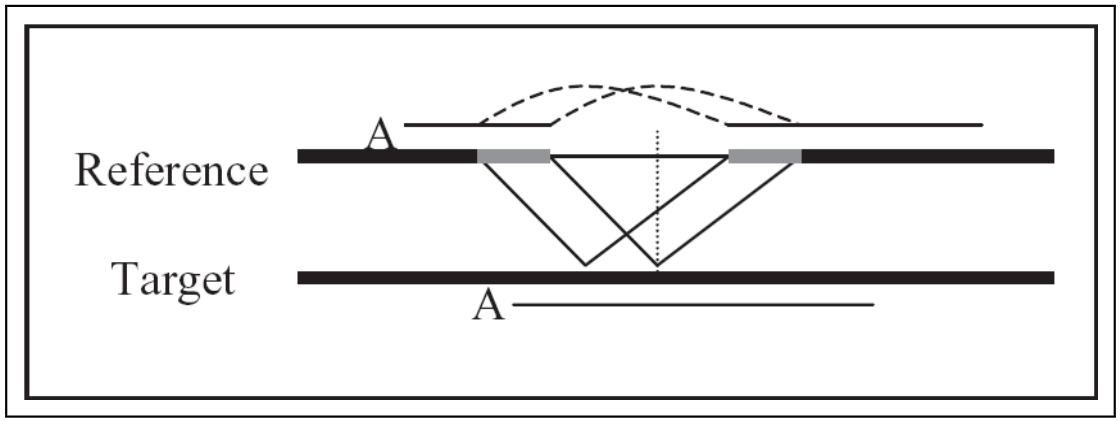

Figure: Insertion into the reference – short flanking repeats. Dashed lines connect sections occurring twice.

Two strains of Streptococcus

Note the repeats in the first 500k bases region.

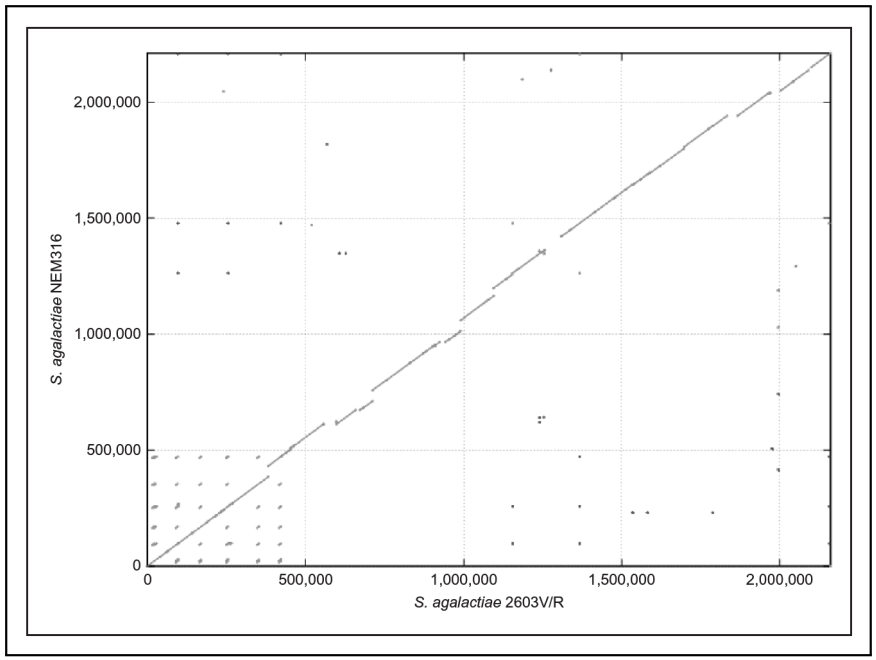

Figure: Two similar strains of streptococcus

Overall statistics of the assemblies

$v.2603 < CelAsm < v.NEM\ 316$

×	v. 2603			v. NEM 316			CelAsm		
	N	Total contig size	N50	N	Total contig size	N50	N	Total contig size	N50
1	604	1,001,743	0	527	839,315	0	585	903,184	0
2	619	1,593,364	2,294	586	1,393,287	1,479	657	1,488,287	1,595
3	443	1,856,394	5,707	450	1,640,231	4,179	506	1,812,266	4,981
5	243	2,043,842	14,915	277	1,829,976	10,395	293	2,046,730	12,458
7	144	2,100,541	27,364	198	1,891,527	18,142	189	2,110,396	21,926
9	86	2,119,579	42,679	155	1,919,237	24,239	130	2,132,490	33,953

Figure: Assembling strain Streptococcus agalactiae 2603: "autoassembly", NEM 316 strain as the reference, baseline – Celera Assembler

Alignment to the original

- Contigs with less than 90% similarity were discarded $\Rightarrow$ N gaps $<$ N from the previous slide
- Autoassembly outperformed Celera Assembler.
- Insertions in the 2603 with respect to the NEM 316 strain are not fair to AMOS. Remove them from Celera and win.

×	*v.* 2603			*v.* NEM 316			CelAsm			LW
	N gaps	Total ga size	% genome covered	*N* gaps	Total gap size	% genome covered	*N* gaps	Total gap size	% genome covered	% genome covered
1	588	1,168, 208	45.92	511	1,329,996	38.43	562	1,261,419	41.61	39.31
2	596	577,987	73.24	552	778,491	63.96	601	679,386	68.55	74.10
3	430	301,899	86.02	415	530,417	75.45	455	365,736	83.07	89.88
5	232	119,917	94.45	240	347,697	83.90	257	153,824	92.88	98.56
7	132	62,410	97.11	155	292,068	86.48	146	81,406	96.23	99.79
9	80	43,408	97.99	110	270,210	87.49	97	61,544	97.15	99.97

Figure: Assembling strain Streptococcus agalactiae 2603: "autoassembly", NEM 316 strain as the reference, baseline – Celera Assembler

The first megabase

- AMOS-Cmp was able to assemble the leading 17k contig.
- Celera contigs end at repeats. AMOS-Cmp does better.
- NEM 2603-based assembly does not cover dissimilarities.

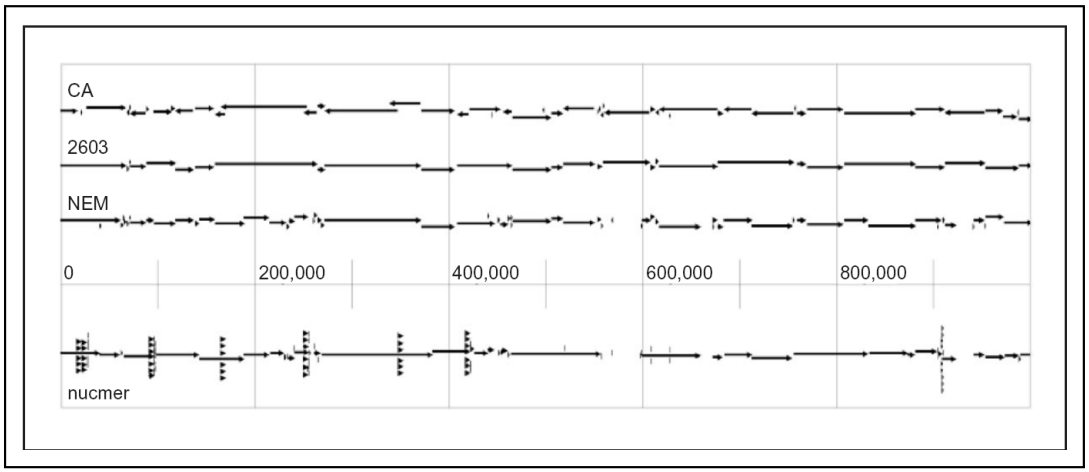

Figure: Assemblies of the first megabase of 2603 with 9x coverage. nucmer – the alignment of NEM 316 to 2603. Arrows mean repeats.

Applicability

Works poorly for dissimilar genomes:

Reference genome (# bases)	*Staphylococcus epidermidis* RP62A [unpublished] (2,616,530)	*Streptococcus pyogenes* MGAS315[26] (1,900,521)	*Streptococcus pyogenes* MGAS8232[21] (1,895,017)	*Streptococcus agalactiae* 2603 V/R[9] (2,160,267)
Target genome (# bases)	*Staphylococcus epidermidis ATCC* 12228[22] (2,499,279)	*Streptococcus pyogenes* SF370 serotype M1[23] (1,852,441)	*Streptococcus pyogenes* SF370 serotype M1[23] (1,852,441)	*Streptococcus pyogenes* SF370 serotype M1[23] (1,852,441)
Region that cannot be assembled (# bases, %)	143,007 (5.72%)	148,192 (7.99%)	142,495 (7.69%)	1,640,396 (88.55%)

Figure: Portion of the genome that cannot assembled for four pairs of similar organisms. The number of bases that cannot be assembled as well as the fraction of the target genome is given.

Discussion & Conclusion

- Outperforms a standard assember such as Celera Assembler in computing resources.
- Relatively high quality of the assembly.
- Works well when the overlap between reads is 10 base pairs or fewer since the overlap is decided by the more significant overlap with the reference. NGS!
- Standard assembler cannot make use of singletons. AMOS-Cmp – can.
- Drawbacks – cannot handle inserts into target, difficulties with divergent sequences.