GAM: Genomic Assemblies Merger

A. Policriti[1,2], S. Scalabrin[1], F. Vezzi[3], R. Vicedomini[2] ✉

[1]Applied Genomics Institute, Udine, Italy
[2]DIMI, University of Udine, Udine, Italy
[3]KTH: Royal Institute of Technology, SciLife Lab Stockholm, Sweden

Motivations

In the last 3 years more than 20 assemblers have been proposed to tackle the hard task of assembling. Recent evaluation efforts (Assemblathon 1 and GAGE) demonstrated that none of these tools clearly outperforms the others. However, results clearly show that some assemblers performs better than others on specific regions and statistics while poorly performing on other regions and evaluation measures.

With this picture in mind we developed GAM (Genomic Assemblies Merger) whose primary goal is to merge two or more assemblies in order to obtain a more contiguous one. Moreover, as a by-product of the merging step, GAM is able to correct mis-assemblies.

GAM does not need global alignment between contigs, making it unique among others Assembly Reconciliation tools. In this way a computationally expensive alignment is avoided, and paralog sequences (likely to create false connection among contigs) do not represent a problem.

GAM procedure is based only on the information coming from reads used in the assembling phases, and it can be used even on assemblies obtained with different datasets.

Methods

Let us concentrate on the the merging of two assemblies, dubbed M and S. As a preprocessing step, that is an almost mandatory analysis, reads (or a subset of them) used in the assembling phase are aligned against M and S using a SAM-compatible aligner (e.g., BWA, rNA).

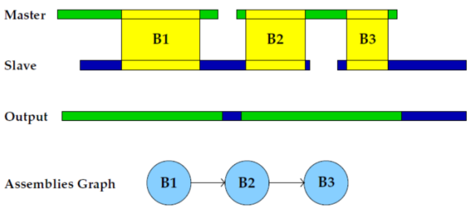
GAM takes as input M, S and the two SAM files produced in the preprocessing step. The main idea is to identify fragments belonging to M and S having high similarity. For this purpose, GAM identifies regions, named blocks, belonging to M and S that share an high enough amount of reads (i.e. regions sharing the same aligned reads).

After all blocks are identified the Assembly Graph (AG) is built: each node corresponds to a block and a directed edge connects block A to block B if the first precedes the second in either M or S (see Fig.1).

Once AG is available, the merging phase can start. As a first step GAM identifies genomic regions in which assemblies contradict each other (loops, bifurcations, etc.). These areas represent potential inconsistencies between the two sequences. We chose to be as much conservative as possible electing (for example) M to be the Master assembly: all its contigs are supposed to be correct and cannot be contradicted. S becomes the Slave and everywhere an inconsistency is found, M is preferred to S.

After the identification and the resolution of problematic regions, GAM visits the simplified graph, merges contigs accordingly to blocks and edges in AG (each merging phase is performed using a Smith-Waterman algorithm variant) and finally outputs the new improved assembly.

GAM is not only limited to contigs, it can also work with scaffolds, filling the N's inserted by an assembler and not by the other.



Genome	Tool	Length	Contigs	L50 (bp)
Olea (chloroplast)	CLC-Ill	127,942 bp	10	16,215
	CLC-454	128,572 bp	9	15,993
	GAM	130,101 bp	3	112,156
Populus trichocarpa	CLC	339,551 Kbp	104,432	6,130
	ABySS	526,633 Kbp	88,193	11,768
	GAM	441,133 Kbp	83,978	14,407
Boa constrictor	CLC	1,363 Mbp	373,909	7,716
	ABySS	1,730 Mbp	7,042,239	2,348
	GAM	1,372 Mbp	367,060	8,031

Figure 1