Assembling Puzzles from Preassembled Blocks

Pavel A. Pevzner

Genome Res. 2001 11: 1461-1462
Access the most recent version at doi:10.1101/gr.206301

Assembling Puzzles from Preassembled Blocks

Pavel A. Pevzner

Department of Computer Science and Engineering, University of California at San Diego, La Jolla, California 92093-0114, USA

Assembling large jigsaw puzzles is difficult, and most of us haven't even seen a 10,000 piece puzzle on sale in a toy store. Such puzzles require an enormous dedication, and most children (not to mention adults) are not willing to put the time and effort into their assembly. Moreover, it is only feasible for multi-feature compositions like "Garden of Pleasures" by Hieronymus Bosch (one of the best-selling large puzzles) with hundreds of people and animals. When Celera assembled their first million-piece puzzle (Myers et al. 2000), the *Drosophila melanogaster* genome, the Public Human Genome Project did not have a program that would reliably assemble even thousand-piece puzzles without errors. Surprisingly enough, there is still no such program in the public domain today.

Not to worry: Kent and Haussler (2001) "saved" the Human Genome Project with their `GigAssembler`. `GigAssembler` is very different from the Celera assembler: It assembles a million-piece puzzle (genome) from thousands of preassembled blocks (BAC contigs). Each such preassembled block may be composed from thousands of the original pieces (reads). The idea is simple: If you see a blue eye in one preassembled block from the "Garden of Pleasures", then you are likely to find one more blue eye in another preassembled block. These two blocks should go together and help in the puzzle assembly. There are plenty of "pairs of eyes" in the genome: paired plasmid ends, BAC end pairs, parts of mRNAs or ESTs, and others. The difficulty, however, is (once again!) in repeats: What if there are many blue-eyed people (or animals) in the puzzle? Another problem is that assembly errors in preassembled blocks lead to complications. Unless such incorrectly assembled blocks are broken into correct parts, they may lead to errors in the final assembly.

The simple idea behind `GigAssembler` would not work for the traditional shotgun assembly because the "blue eyes" are not seen in the original small pieces (reads): Every blue eye is broken into many pieces. However, thanks to the hard work of `Phrap`, `Consed` (Gordon et al. 1998), and the army of finishers in many sequencing centers worldwide, the blocks have already been preassembled. As a result, instead of assembling millions of short 600-bp reads, Kent and Haussler assemble thousands of much larger (10,000 bp on average) blocks. This approach, of course, assumes that all (or most of) these blocks (BACs) are assembled correctly. Celera apparently was skeptical about the quality of BAC assemblies. Instead of using preassembled blocks, the Celera assembler shred them into small pieces, mimicking the original sequencing reads (Venter et al. 2001). Although Celera reported that a significant number of BAC assemblies conflict with their shotgun data, `GigAssembler` seems to be able to successfully handle most such misassemblies.

There are two surprising things about the `GigAssembler` algorithm: It is simple and it works. `GigAssembler` is a greedy algorithm that first assembles the pieces that best fit together and continues with less and less well-fitting pieces. For example, if there are 10 blocks with blue eyes in the puzzle, it is not clear how to combine them into five pairs (not to mention that there are a number of one-eyed creatures in the "Garden of Pleasures"). However, if two of these 10 blocks also come with diamond earrings there is strong evidence that these two pieces should go together. `GigAssembler` uses mRNA, ESTs, and other types of data as additional pieces of information about block pairing (diamond earrings), and estimates the fitness score for every two blocks of the puzzle. Afterwards, the blocks are assembled in a greedy fashion starting from the best-fit pairs.

Greedy algorithms, although simple, face many problems in the conventional shotgun fragment assembly: The shotgun reads may be too short to take advantage of the blue-eye principle and the repeats may be too numerous. Knowing that the greedy strategy does not work well for shotgun assembly, Kent and Haussler made a bet on the fact that preassembled blocks are long enough to make the blue-eye principle work. After implementing `GigAssembler` and assembling the public working draft of the human genome, they proved that it is indeed the case (International Human Genome Sequencing Consortium 2001).

Although `GigAssembler` is based on a simple greedy principle, the implementation has to deal with many challenges. The distinguished and unique feature of `GigAssembler` is the variety of information it uses for assembly. To produce the working draft of the human genome, it had to deal with 400,000 sequence contigs from 30,000 large insert clones, process billions of bases of ESTs, etc. Instead of masking repeats, as done bythe Celera assembler, `GigAssembler` tries to utilize them and faces a difficult problem of dealing with both accurate contig data and rather inaccurate EST and BAC end single reads data. Another problem is how to deal with the conflicts and how to position the contigs that do not overlap: What if the blue-eye principle suggests that a block *A* should go after a block *B*, while the brown-eye principle suggests that *A* should go after another block *C*? Fortunately, a similar problem has been addressed in the past for physical mapping (Thayer et al. 1999). In computer science it is known as constraint satisfaction, and Kent and Haussler take advantage of the classical Bellman-Ford algorithm to resolve conflicts in `GigAssembler`.

The success of `GigAssembler` may influence the future decisions on how to proceed with other genomic sequencing projects. The question of whether the future belongs to the shotgun approach or to the BAC-by-BAC approach or to a hybrid approach is still being debated. In the end it will boil down to the economics: The least expensive and the most accurate approach will prevail. Unfortunately, it is not easy to predict the cost for each of these approaches and one of the unknown variables is that we don't know yet the limits of the next generation of assembly algorithms. The debates about the optimal way to sequence genomes are not new. They started four years ago with two famous back-to-back *Genome Research* papers: "Human whole-genome shotgun sequencing" by James Weber and Gene Myers (Weber and Myers 1997) and "Against a whole-genome shotgun" by Phil Green (Green 1997). After Myers and colleagues published the algorithm behind the Celera assembler and assembled the *Drosophila melanogaster* ge-

E-MAIL ppevzner@cs.ucsd.edu; FAX (858) 534–7029.

Article and publication are at http://www.genome.org/cgi/doi/10.1101/gr.206301.

nome, many of us started to think that Phil Green had lost this debate. After re-reading "Against a whole-genome shotgun", I have a different opinion: It is a very convincing proof that the whole-genome shotgun would fail with Phrap or any other fragment assembler circa 1997. The paper provided many insightful biological arguments in favor of the clone-by-clone approach that still apply today and put forth some strong arguments in favor of the hybrid approach. In particular, it is still unclear whether some "difficult-to-assemble" parts of the genome (i.e., many highly repetitive regions from the Y chromosome) can be resolved with a pure shotgun approach. Some BACs from such regions are so repeat-rich that their assembly presents a combinatorial conundrum even with 10+ coverage. Because the Celera assembler masks the repeats it is not clear yet whether the quality of their assembly deteriorates in such regions. It may turn out in the end that Phil Green was right and that for these "difficult-to-assemble" parts the clone-by-clone approach is the best strategy to infer the complicated repeat structure.

The only flaw with Phil Green's paper is the failure to make a bet on a new generation of algorithms that made the whole-genome shotgun assembly feasible. Gene Myers made this bet and transformed fragment assembly from a sleepy corner of bioinformatics into one of the most active research directions. When Kent and Haussler started GigAssembler last year, it was not obvious whether the problem was solvable at all and whether there would be enough information in BAC overlaps, mRNAs, ESTs, etc., to make their algorithm work. They made their bet and they won. Neither Myers nor Kent and Haussler were 100% sure that they would come up with a correct assembly until the very moment the programs were completed and the first assemblies were generated. An important lesson for the ongoing discussions on how to proceed with future genomic projects: Never underestimate the power of algorithms and make your bets even when the future results are not 100% guaranteed. Otherwise, we still would be waiting for the first draft of the human genome to appear.

REFERENCES

Gordon, D., Abajian, C., AND Green, P. 1998. *Genome Res.* **8:** 195–202.
Green, P. 1997. *Genome Res.* **7:** 410–417.
International Human Genome Sequencing Consortium. 2001. *Nature* **409:** 860–921.
Kent, W.J. and Haussler, D. 2001. *Genome Res.* **11:** 1541–1549.
Myers, E.W., Sutton, G.G., Delcher, A.L., Dew, I.M., Fasulo, D.P., Flanigan, M.J., Kravitz, S.A., Mobarry, C.M., Reinert, K.H., Remington, K.A., et al. 2000. *Science* **287:** 2196–2204.
Thayer, E.C., Olson, M.V., and Karp, R.M. 1999. *Genome Res.* **9:** 79–90.
Venter, J.C., et al. *Science* **291:** 1304–1352.
Weber, J. and Meyers, G. 1997. *Genome Res.* **7:** 401–409.