**The large amount of short read data that has to be assembled in future applications, such as in metagenomics or cancer genomics, strongly motivates the investigation of disk-based approaches to index next-generation sequencing (NGS) data. Positive results in this direction stimulate the investigation of efficient external memory algorithms for de novo assembly from NGS data. Our article is also motivated by the open problem of designing a space-efficient algorithm to compute a string graph using an indexing procedure based on the Burrows–Wheeler transform (BWT). We have developed a disk-based algorithm for computing string graphs in external memory: the light string graph (LSG). LSG relies on a new representation of the FM-index that is exploited to use an amount of main memory requirement that is independent from the size of the data set. Moreover, we have developed a pipeline for genome assembly from NGS data that integrates LSG with the assembly step of SGA (Simpson and Durbin,** [**2012**](javascript:popRef2('B23'))**), a state-of-the-art string graph-based assembler, and uses BEETL for indexing the input data. LSG is open source software and is available online. We have analyzed our implementation on a 875-million read whole-genome dataset, on which LSG has built the string graph using only 1GB of main memory (reducing the memory occupation by a factor of 50 with respect to SGA), while requiring slightly more than twice the time than SGA. The analysis of the entire pipeline shows an important decrease in memory usage, while managing to have only a moderate increase in the running time.**