

# Abstract

Trinity is one of the most novel methods of assembling short sequencing reads into large transcripts. Inchworm, one of the modules of Trinity software extracts the k-mer information from the FASTA file generated by Jellyfish to assemble overlapping k-mers into larger linear transcripts. Since the original implementation purely written in C++ has several performance bottlenecks, it can be tuned to obtain a significant performance boost. This thesis uncovers some of important performance bottlenecks and presents various techniques to obtain a significant performance boost of the application. I have developed an optimized parallel version of Inchworm and the thesis provides the comparison of this version and the original implementation using two standard datasets. It also describes some techniques to improve the quality of finally obtained transcripts by manipulating the present implemented greedy algorithm of assembling overlapping k-mers. This thesis also suggests and discusses some ideas that may help developers to optimize Inchworm to further extent.