Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Assembly of the highly similar sequences of the IS elements by the means of phrap and miniassembly maker perl scripts.

Markiyan Samborskyy
Department of Biochemistry, University of Cambridge
Markiyan Oliynyk
Department of Biochemistry, University of Cambridge

Method Article

Keywords: genome sequencing, missassembly resolving, repeats resolving, repeats sequencing.

Posted Date: March 22nd, 2007

DOI: https://doi.org/10.1038/nprot.2007.182

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Abstract

Introduction

The aim of this protocol is to help with resolution of highly repetitive sequences when sequencing by the whole genome shotgun sequencing strategy.

Reagents

1. ABI genome sequencing results: a. Whole genome shotgun sequencing data, b. Reads from large template library ends \(cosmids or BAC), c. Template finishing reads \(if avaiable), 2. Genome assembly \(by phrap) imported into the MySQL database. 3. Genome assembly mapping data \(contig linkage data and contig consistency data).

Equipment

Hardware: PC with linux \(64 bit strongly recommended for whole genome assembly, and at least 2 GB of RAM). Software: Phred/Phrap/Consed \(v14); NCBI BLAST; MySQL server Set of PERL scripts used for automation of the routine tasks.

Procedure

The idea is to do the miniassembly with one copy of the IS \(or any other repeat sequence), finish it \(if necessary), and to export resulting consensus back in to the main assembly as one "long read" or "scaffolding read", - consensus of repeat itself with bouth flanking non repetitive regions, at least few hundreds bp. each. 1. Get the mapping data \(contig region linkage information) and contig consistency data from the main assembly. It is retreived from large template ends reads pairs and used to locate misassemblies. 2. Repeat border localisation. Using NCBI BLASTN against the current main assembly database, locate borders of the repeat region \(where it begins and ends), by the means blasting\(N) of contig\(s) fragment\(s) with the current assembly blastN database. I recommend using master/slave aligment output mode for spotting repetitive regions. Also be avare, that real repeat borders can be different due to current assembly artifacts. Also, if it is known that, for example, the repetitive region contains known genes - transposase, than this info can be used as auxililary for repeats location finding. Also use information provided by "matchElsewhereHighQual" tags in the consed. 3. Define unique sequence "anchor regions" coordinates in the assembly - based on the repeats borders and template reads pairs information allocate coordinates of the non repetitive flanking regions, which does not contain other repeats, or other assembly problems. Usually it is from 50-100 bp from the repeat end to up to 35-40 KB from the repeat. Also note the direction to the problematic region\(repeat) - U\(Unicore, repeat after anchor region) or C\(Complement, repeat before anchor region). 4. Obtain the list of ALL templates used for sequencing from anchor regions. 5. Obtain the list of all reads which where obtained from templates anchor templates. 6. Make separate minniassembly from these reads \(obtained in step 5).

Please include all chromatograms and all corresponding PhD files, including the ones with the edits. I was making separate phredPhrap project for that. PS: \(Steps 4-6 were automated by the means of gnm_region_auto_reasm.pl) 7. Finish miniassembly by conventional methodics, using templates, which contain this region and only one copy of the repeat. Be sure to have the repeat in the good quality and error free before putting it back into the main assembly. 8. Once finished, exprort consensus of the miniassebly into the main assembly as JoiningRead_###.phd.1 file with quality vallues, where ### is the miniassembly ID, put this file into the phd_dir of the main assembly. 9. Reassemble the main assembly. 10. Check the assembly results by blasting the miniassembly consensus with the curent assembly, and from consed itself. Now this region shold be correctly assebled.

Timing

It is very dependant from the reads coverage distribution over the affected region and oligo order turnarround speed, usually, when properly set up, it can be from 1 day - to 1 month.

Troubleshooting

If you can't map oposite end \(You have "physical gap") - make new library using different digestion conditions \(or sequence more clones from the existing one \(up to 20X template coverage)), doesn't helps - refer to methods fo physical gap finishing. Problems due to repeats within miniassembly -> try making miniassembly for each repeat copy seperatelly - if impossible \(two or more 1KB 100% identical IS copies next to each other) - use other sequencing strategies \(cosmid shotgun, restriction mapping and subcloning, and than subclone sequencing for that region) Final assembly problems due to reads from within repetitive region interfearing with the assembly - try making miniassemblies for all representatives of the particular repeat family, and than taking reads which contain the repeat itself out from the main assembly. Substitute them by the miniassembly consensus backbones, also try increasing flanking region length.