This collection of software comprises a single-nucleotide polymorphism detection pipeline, as described in Tan et al. 2014. A microscopy-based screen employing multiplex genome sequencing identifies cargo-specific requirements for dynein velocity. Mol. Biol. Cell.

DEPENDENCIES

We require:

* Preliminaries
  - A UNIX-like environment (such as Mac OS or any standard flavor of Linux, but not Windows)
  - Python (we have tested our code with Python versions 2.6.6 and 2.7.3)

* Biological software
  - Samtools: http://samtools.sourceforge.net/
  - BWA: http://bio-bwa.sourceforge.net/
  - GATK: http://www.broadinstitute.org/gatk/
  - Biopython: http://biopython.org/

CONTENTS AND USAGE

We label our raw sequence files as:

[chars][M/WT][further text].txt

In our case, 'chars' is a three-digit identifying number, although it can be any set of characters. M indicates that the file contains data from mutants; WT indicates that it contains WT data from parent or sibling strains. M and WT designations are essential, and – while they may be placed anywhere before the .txt -- neither the characters 'M' nor 'WT' may appear anywhere in the file-name segment 'chars' or 'further text'. This restriction may be relaxed by substituting different strings in lines 18 and 19 of synonymous.py to distinguish mutant from wild-type data.

* assemble.sh
  - depends on assemble-multiple.py (see below)

This script calls BWA, Samtools, and the GATK in order to assemble genome data. The first few lines of the script define the locations of reference data and BWA, Samtools, and the GATK GenomeAnalysis.jar file, and they must be set for your particular environment before beginning the analysis.

We have tested it with BWA 0.6.2, Samtools 0.1.18, and the Genome Analysis Toolkit version 2.4-7. We have encountered problems with other versions (both older and newer) of the GATK and BWA. The script can in principle be modified to accomodate paired-end reads or color space data, but as written is configured for single-end DNA reads.

It should be run in a directory that contains all the files that are desired to be included in an analysis. The output are files [chars][M/WT][further text].vcf. It takes no command-line arguments. It will attempt to assemble all .txt files for which there does not exist a corresponding .vcf file in the directory in which it is run, but will fail with an error if the .txt file is not a well-formed sequence file.

* synonymous.py
  - depends on assign_to_ORFs.py

This script should be run after assemble.sh. It performs SNP subtraction and removal of synonymous mutations, and sorts but does not filter on the basis of read depth quality. It outputs [chars]M[further text]_subtr_all, the final product of the analysis.

It also writes archive files ending in .pk; these are much faster to read and process than .vcf files. If you assemble a set of WT sequences for some analysis and wish to subtract them from a new set mutant sequences, you may simply move the .pk or .vcf files to a directory with the new data (or add the new mutants to the original directory) without reassembling the original data.

* assemble-multiple.py FILE PROPORTION

This script automatically recognizes common barcode sequences appended to the start of reads in a FASTQ file 'FILE'. It removes these sequences and prints to STDOUT, which must be redirected with >. The optional argument 'PROPORTION', if between 0 and 1, causes assemble-multiple to randomly select that proportion of reads from the file, without replacement, rather than writing out all reads.

SUPPORT

These scripts were written by Mark Chonofsky while working in the Reck-Peterson Lab at Harvard Medical School. We cannot offer detailed support, but please contact Mark with any queries or bugs by emailing him at:

first initial last name at gmail

LICENSE