VecFinder: Automated de novo identification and removal of vector and adapter sequence from genomic datasets

Michael Yu Zhang [1], Hank Tu [1], Harris Shapiro [1], Darren Platt [2 1]

[1] Lawrence Berkeley National Laboratory/JGI, [2] Lawrence Livermore National Laboratory/JGI

May 4, 2007

ACKNOWLEDGMENTS:

The work conducted by the US Department of Energy (DOE) Joint Genome Institute is supported by the Office of Science of the DOE under Contract Number DE-AC02-05CH11231. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government, or any agency thereof, or the Regents of the University of California.

DISCLAIMER:

[**LBNL**] This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California.

[**LLNL**] This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

VecFinder: Automated de novo identification and removal of vector and adapter sequence from genomic datasets

Michael Yu Zhang, Hank Tu, Harris Shapiro, Darren Platt

Introduction

High-throughput Sanger sequencing requires DNA to be inserted into bacterial vectors for biological amplification. Adapter or linker oligonucleotides may also be attached to target DNA fragments to facilitate insertion into the vector. These vector and adapter sequences are sequenced concomitantly with the target, or insert, sequence and represent contamination which must be removed from the dataset prior to analysis. Removal of such contamination can be accomplished by screening the dataset against the known sequence of the vector and adapter used to generate the data. However, often in the case of public or collaborator datasets, information regarding these contaminant sequences may be incorrect or absent, resulting in an incomplete screening.

We've created a piece of software, **VecFinder**, which is able to identify the sequence of the vector and adapter from the read sequences alone and subsequently remove it. This alleviates the dependence on the library creators to provide the vector and adapter sequences used for the library. It also automates the previously manual task of identifying and screening the adapter or linker sequence, which can be tedious and time-consuming

Accomplishments

- Reduce reliance on knowledge of vector and adapter sequences used to generate a dataset
- Eliminate need for manual identification of contaminant sequences
- Increase automation and robustness of genome assembly and EST/cDNA pipelines
- Improve genome assembly quality by enabling cleaner datasets

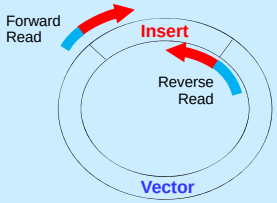
Algorithm Overview

K-mer frequency analysis

The VecFinder algorithm is based on the analysis of subsequence, or k-mer, frequencies within a sequencing dataset.

The same vector/adapter sequence is present in nearly every read, while the insert sequence is relatively unique. Accordingly, the comers from the vector sequence are found at much higher frequencies in the dataset than the k-mers from the inserts.

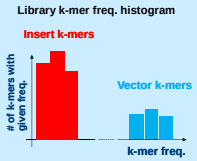
We take advantage of the contrast in k-mer frequency between the vector and the insert to pinpoint the vector/insert junction for each read.



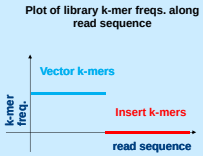
– Common vector sequence
– Unique insert sequence

– **Subsequence (8bp k-mer) frequency signature**
  – **Vector/adapter k-mers: higher frequencies**
  – **Insert k-mers: lower frequencies (~1-100)**

Dataset-based



Read-based

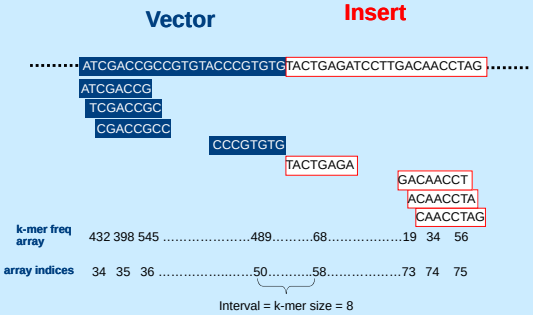


Calculate vector/insert boundaries

Select *n* reads to represent the entire dataset for identifying the vector sequence, where *n* is chosen to limit the runtime of the program. For each read in the subset, identify the suspected vector/adapter sequence for the read using the following procedure.

1. K-mer Frequency Array

   Generate a k-mer frequency array using the dataset frequencies



2. Suppress low complexity k-mers

   – Homo-polymer runs (AAAAAAAA) have elevated frequencies which mimic that of the vector signature.

   Set the frequency of every k-mer which contains a base of the homopolymer to zero

3. Interval Differential Array

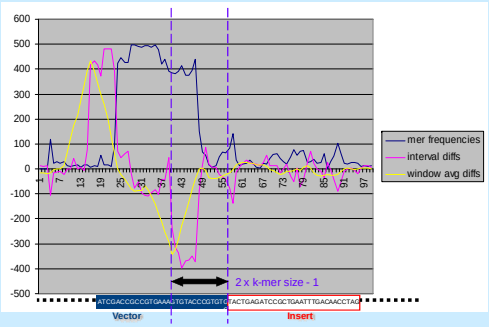
   Create an array containing the difference between every pair of k-mers separated by a given interval in the K-mer Frequency Array

   Use interval equal to the k-mer size to maximize difference at the vector/insert boundary

4. Moving Window Average Array

   Perform a moving window average on the Interval Differential Array.

   In order to turn a plateau of high interval differentials into a single peak, a moving window average is performed. The window size is determined by the length of the plateau, which is approximately the k-mer size (8).



5. Calculate the index of the read corresponding to the beginning or end of the insert

   Add 2 * k-mer size – 1 (2 * 8 – 1 = 17) to:

   – Local minima in the 5' half of the read to find the 5' insert start coordinate
   – Local maxima in the 5' half of the read to find the 3' insert end coordinate (if applicable)

Identify and remove contaminant sequence

1. Generate a list of candidate vector sequences from the selected list of *n* reads, one for each sequencing direction
2. Find a consensus vector/adapter sequence using multiple alignment
   – "Multiple sequence alignment with hierarchical clustering" F. CORPET, 1988, Nucl. Acids Res., 16 (22), 10881-10890
   – Generate a forward and a reverse vector consensus
3. Use megablast to find the insert boundaries for every read
4. Remove the sequence from the ends of the read to the insert boundaries

The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231