A non-programmatic description of methods

Tim Millar

2017-07-31

Contents

Overview

TEFingerprint is a Python 3 library and command line tool for producing transposon based fingerprints from paired end reads. TEFingerprint seeks to characterise data sets for further downstream analysis.

The TEFingerprint pipeline is composed of the following steps:

1. Mapping paired end reads to a database of known transposons
2. Identification of *informative* reads
3. Mapping informative reads to a reference genome
4. Fingerprinting (density based clustering) of mapped read positions
5. Computing the union of the fingerprints from multiple samples
6. Comparing read counts among samples within the combined fingerprint
7. Downstream filtering and analysis of results

Selecting Informative Reads

The initial stage of the TEFingerprint process is to identify *inforamtive* reads. Reads are informative if they relate some information about the location of transposon insertions when compared to a reference genome.

Paired end reads are mapped to a library of known transposon sequences. Pairs are then identified in which one read has mapped to a transposon and the other is unmapped. Assuming that our library of known transposon sequences is largely complete, unmapped reads are likely to (primarily) contain non-transposon-genomic DNA. Read pairs in which both reads have mapped to a transposon sequence can be used as an additional source of information if one read has a significant soft-clipped region at its 5-prime (outer) end. In these cases, the soft-clipped section can be extracted and included as (short) informative read. Any pairs in which neither read is mapped are uninformative and ignored.

The (unmapped) informative reads are tagged with the transposon that their pair has mapped to and mapped to a reference genome. These reads will tend to map in stranded clusters either end of a location at which a transposon is resent in the sample.

Fingerprinting

Once a set of informative reads has been identified, labeled and aligned to a reference genome, they may be used to generate a *fingerprint*. To this end reads are grouped by transposon category, typically at super-family or family level, and the position of their 3-prime end is extracted. A density-based clustering algorithm is then used to identify regions of the reference genome which are dense with informative read 'tips'.

Density based clustering

When identifying clusters of read tips to identify potential transposon insertions, the ideal clustering algorithm would meet the following conditions:

1. The number of clusters does not need to be specified a priori
2. Points (read tips) may be classified as 'noise' points if they do not form a suitably dense cluster
3. Parameters can be tuned to detect informative signal at/around a specific density level
4. The algorithm has some flexibility to distinguish between (split up) proximate clusters

The first three conditions point to DBSCAN(*) (Ester et al. 1996, Campello et al. 2015) is a suitable method for cluster identification. However DBSCAN* is limited to identifying clusters at a single density. This can be problematic if two or more insertions (of the same variety) occur close to one another as DBSCAN* is likely to treat them as a single larger cluster (i.e. does not meet condition four).

HDBSCAN* (Hierarchical DBSCAN*) described by (Campello et al. 2015) can identify clusters at any density (based on a mesure of cluster stability) but is too flexible for our purpose and unable (in its basic form) to target a specific density level (i.e. does not meet condition three). Condition three is particularly important for the purpose of TEFingerprint as the targeted clusters (ends of insertions sites) will generally have a consistent density/size which can be estimated from the insert size. Furthermore their may be *real* clusters within the data that are uninformative for our purpose. For example Gypsy elements tend to concentrate around centromeres leading to a large clusters (composed of the reads for many insertions) within the scale of the entire genome.

Two clustering algorithms available in TEFingerprint. A univariate implementation of DBSCAN* (non-hierarchical) is made available but not used by default. The second (default) hierarchical univariate method which is derived from HDBSCAN* but differing in the following ways:

1. Based on the notation established by Campello et al. 2015, **Cluster support** is calculated as:

$$S(\mathbf{C}_i) = \sum_{\mathbf{x}_j \in \mathbf{C}_i} \varepsilon_{\max}(\mathbf{C}_i) - \varepsilon_{\min}(\mathbf{x}_j, \mathbf{C}_i)$$

   where $\varepsilon_{\min}(\mathbf{x}_j, \mathbf{C}_i)$ is calculated as $\max\{d_{\text{core}}(\mathbf{x}_j), \varepsilon_{\min}(\mathbf{C}_i)\}$

2. **Cluster selection** is then performed in a top down manner where a cluster is selected if its support is greater than the combined support of its child clusters.

3. The search space of the cluster tree may be constrained by optional global maximum and minimum values of ε. If included then cluster support will calculated values of epsilon between the global minimum and maximum values. By default cluster support is calculated for all values of ε from below the root node of the cluster hierarchy to 0 (as in HDBSCAN*).

In practice the global minimum value ofε is left at 0 and the global maximum value of ε is set to the approximate insert size of paired end reads. This produces clusters with relatively constant density, similar to that of DBSCAN*, but with improved identification and separation of proximate clusters.

Comparing Multiple Samples

Fingerprinting produces a binary (i.e. presence absence) pattern of loci across a reference genome indicating the boundaries of transposon insertions within a samples genome. However the binary pattern is extracted from non-binary data (read positions/counts) and the absence of a cluster in one sample does not

guarantee an absence of signal (reads) within that location. Therefore a direct comparison of fingerprints from multiple samples may be misleading.

A better approach is to compare read counts within the combined (union of) fingerprints. Mathematically each cluster in a fingerprint can be expresed as a closed interval. For example a cluster spanning the region between points a and b (inclusive) can be expressed as the closed interval $[a, b]$. The fingerprint of sample i can then be expresed as the union of every interval (cluster) found within that sample U_i. Thus the union of fingerprints for a set of n samples is calculated:

$$\bigcup_{i=1}^{n} U_i$$

The new union of fingerprints represents the boundaries of potential transposon insertions across all samples. We then use each interval within the union of fingerprints as a potential insertion site for *all* of the samples. A samples read count within a given interval is recorded as evidence for the presence or absence of an insertion at the genomic location represented by that interval.

In this manner, TEFingerprint identifies comparative characters (potential insertion sites) for a group of samples and summarises each samples support (read counts) for the presence/absence of a character.