Protocol:

Connor: An Open Source Tool for Processing ThruPLEX® Tag-seq Data with Unique Molecular Tags

Background

Researchers are increasingly searching for low frequency variants in diverse genomic populations using deep coverage, next-generation sequencing data. Errors in PCR amplification and instrument base calling confound attempts to reduce false positives and discover true biological variants. In response, a number of researchers have introduced DNA tagging methods. By adding a molecule-specific DNA sequence prior to steps involving amplification, PCR duplicates can be identified and utilized to reconstruct the sequence of the original biological molecule. This allows the partitioning of PCR and sequencing errors from true, low-frequency variants.

The bioinformatic analysis tool, Connor, de-duplicates a tagged BAM file and produces a BAM file with consensus alignment pairs that represent the original biological molecules. In terms of data workflow, Connor is similar to position-based deduplication (e.g. Picard MarkDuplicates[1]) with changes to address two major challenges. These challenges are:

(1) original molecules found in small target regions with ultra-deep coverage are disproportionally discarded and (2) consensus sequences that match the reference genome are typically chosen over the sequences containing variants. To address these challenges, Connor combines sequences where the alignment structure and molecular tags match, creating consensus sequences that model the original molecules.

Workflow

Figure 1: Bioinformatics analysis workflow.

Connor assumes the input BAM files are tagged using the Rubicon Genomics' ThruPLEX® Tag-seq Kit. In particular, Connor assumes each query sequence begins with a 6 nucleotide unique molecular tag (UMT), followed by an 8 to 11 nucleotide non-random stem sequence and then the target sequence region. The aligner should preserve the leading UMT and stem sequences and mark those areas as "soft clipped" in the BAM CIGAR field to indicate they did not match the reference; this is default behavior for some aligners (e.g. Burrows-Wheeler Aligner, BWA[2]).

Region	UMT	Stem	Target Sequence
Sequence	ACTGTT	GTAGCTCA	GTTGAGACACAT...
CIGAR	Soft Clipped		Matched Ref

Table 1: BAM file requirements. *The BAM CIGAR field should mark the UMT and Stem as "soft clipped" and the target sequence should be marked as "Matched Reference".*

Because a correct UMT and consistent alignment structure are integral to Connor's ability to accurately deduplicate and avoid any manipulations of the raw FASTQ or aligned BAM files. Examples of problematic manipulations include:

- **End Trimming:** Removes base calls from the front of the sequence (the location of the UMT Tag) which would prevent affected reads from being correctly grouped.
- **Pre-alignment Quality Trimming:** Removes reads from the end of the sequence creating distinct CIGAR values obscuring the original alignment structure.

Connor has been tested with the following aligners (using default parameters except where noted):

- BWA v. 0.7.12
- Bowtie2 v. 2.2.4 (local mode)
- DNASTAR SeqMan NGen v.13.0.2.2 (disable clipping and deduplication)
- Hisat2 v.2.0.4
- Novoalign v.3.04.06

Installation Requirements

Connor requires python 2.7 or later and has been tested with:

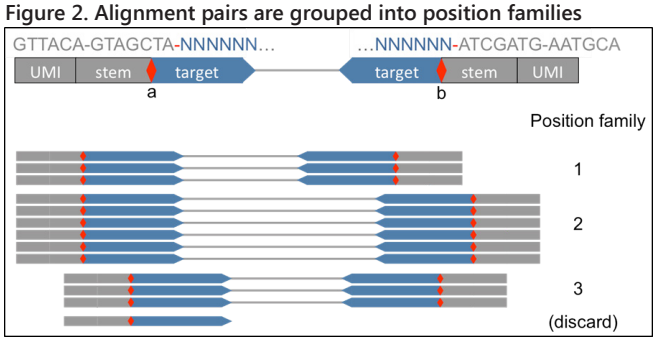
- Python 2.7 and 3.4
- pysam 0.8.4 and 0.9.0
- OSX
- nix RHEL6/7

Connor does not work in Windows OS because it depends on the python library pysam, which is not supported on Windows.

Overview of Connor's UMT Deduplication Method

1. Discard alignments that could not be mapped, are not properly paired, have low mapping quality (<1), or are missing their CIGAR value
2. Discard alignments whose pair partner is missing
3. Group together alignment pairs which share the same stem-template edge coordinates into "position families" (Figure 2)

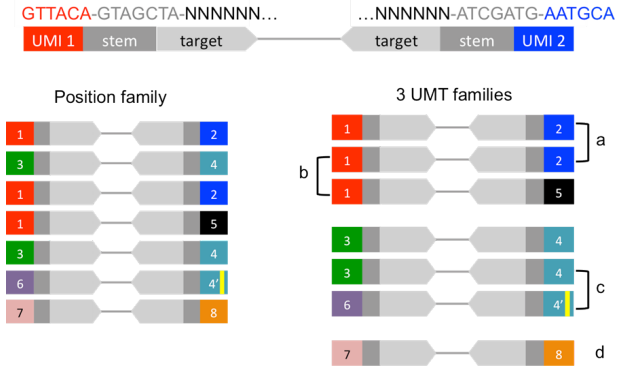
Figure 2. Alignment pairs are grouped into position families



Reads are grouped into position families based on their alignment to the reference. Each bar represents a paired alignment positioned on the reference. The stem-target boundaries of the target sequences define the groups (beginning of left target, end of the right target) [a,b]. Unmapped, unpaired or low-quality reads are removed.

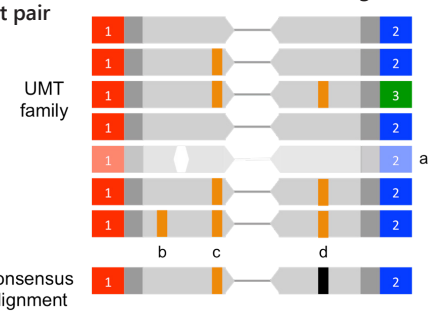
4. Based on left and right UMT, subdivide each "position family" into "UMT families" (Figure 3)
   a. Extract left + right (combined 12-mer) UMT and sort by frequency into candidate families
   b. For each alignment pair, loop over candidate family (in descending popularity) comparing the alignment UMT with the candidate family
      - Exact match across 12-mers is considered a match
      - Exact match of left or right UMT (6-mer) is considered a match
      - Inexact match (within a user-defined Hamming distance, default=1) of left or right UMT is considered a match
   c. Each alignment pair will either match an existing family or create its own family
5. Within each UMT family, establish the majority CIGAR across alignments; discard alignment pairs with non-majority CIGARs
6. Discard UMT families with fewer than a user-defined number of original pairs (default=3 pairs)
7. For each UMT family, collapse the set of original alignment pairs into a single consensus alignment pair. (Figure 4)
   - Consensus sequence is determined by majority vote at each position in the base call sequence
   - Any position with less than a user-defined percent majority (default=60%) results in an N at that position
   - Consensus quality is determined by majority vote at each position in the quality sequence
   - Consensus CIGAR is the majority cigar for that UMT family

Figure 3: Position families are subdivided into UMT families



Each bar represents a paired alignment; distinct UMT tags are numbered and shaded. The collection of the left is the alignments from the position family; on the right those same alignments are partitioned into three UMT families. Rare anomalies in tagging and PCR processes can erroneously split families. To prevent spurious family splits, an alignment will be admitted to a UMT family if either left or right UMT tag is similar (within specified hamming distance). Several match types are noted: [a] are exact left-right matches, [b] is exact one-side match, [c] is inexact one-side match. Small families (<3 alignment pairs) are discarded [d].

Figure 4: UMT families are combined into a single consensus alignment pair



The set of alignment pairs in a UMT family (top) are combined into a single consensus pair (bottom) by majority vote. Each bar represents a paired alignment; distinct UMT are numbered and shaded; mismatches (candidate variants) in the target sequence region are highlighted. Within a UMT family, only structurally identical alignments (i.e. matching CIGAR values) can be combined; alignment pairs with minority CIGAR values are discarded [a]. Consensus alignment preserves the majority UMT left and right tag and stem. In the target sequence region, the base calls are tallied for each position and the majority base call becomes the consensus base call [b,c]. If the majority base call is less than the consensus sequence threshold (60% by default), the base call is replaced by "N" indicating ambiguity [d].

References

1. Picard - A set of tools (in Java) for working with next generation sequencing data in the BAM format. http:// broadinstitute.github.io/picard.

2. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics. 2009, 25, 1754-1760.

Trademarks

Any questions regarding this tool should be sent to:
bfx-connor@umich.edu.

The tool was co-developed with the University of Michigan Bioinformatics Core.