De Novo Error Correction for SOLiD(TM) data
SAET v.2.2

Abstract

SOLiD Accuracy Enhancer Tool (SAET) is a spectral alignment error correction tool which applied to raw data generated by SOLiD platform reduces the color calling error rate by factor of 3-5 without having the reference genome. Decrease in error rate improves mapping, SNP calling, and de novo assembly results. Mapping becomes more accurate and the number of mapped reads increases by 40-50%. The SNP calling on error corrected data results in up to 2 times more true positive calls, and slight increase or decrease in number of false positive calls. De novo assembly on error corrected reads results in up to 3-5 times increase in average contig length. Performance of SAET was tested on various datasets including a large spectrum of genome sizes and complexities, as well as, coverages and read lengths. SAET shows similar performance on genomes of size 1Kbp - 200Mbp with coverage 10-4000x and read length 25-75bp from human sub-genomes, human transcriptomes, and bacterial genomes.

1 Installation

Download: `saet.2.2.zip` from **sourceForge**
`https://appliedbio.sfee-hosted.com/integration/viewcvs/viewcvs.cgi/?root=error_correction_spectral_alignment&system=exsy1002`
Project name: SOLiD_Tools/error_correction_spectral_alignment

Hardware: SAET executable is built and tested for `x86_64 GNU/Linux` systems.
Unzip and set running permissions.

```
> unzip saet.2.2.zip
> cd saet
> chmod 777 saet_mp
```

2 Running

```
./saet_mp <reads.csfasta> <reads.qual> <refLength> [-options]

Input:
 reads.csfasta - csfasta file with original reads (in color space).
                 Example: reads.csfasta
                 ======================
                 >469_29_17_F3
                 T2033031030123133032323113101332112233313212131032O
                 >469_29_1434_F3
                 T132113.21231311212222311021.1221112112220..2123221
                 >469_30_449_R3
                 G13033333333010203300112313002220202011010022101221
                 ======================
                 the title of each read and first two characters are irrelevant.
                 missing colors are encoded as dots.
                 header of the file may contain comments and descriptions.
```

```
 reads.qual    - filename with quality values (if available). Notice, that order
                 of reads in csfasta file should be the same as in quality value file.
                 if file is not available then input "none".
 refLength     - expected length of sequenced (or enriched) DNA region, e.g.,
                 4600000 for E.Coli 4.6Mb genome or 30,000,000 for Whole Human Transcriptome.

Output:

fixed/reads.csfasta - csfasta with corrected reads in color space
fixed/reads.qual    - quality value file where quality values of corrected positions are replaced
                      with zero (for SNP calling).

Advanced options:

 -fixdir dir         Output spectrum and fixed reads into "dir" directory (Default "fixed").
 -trustprefix len    Use only first len positions of reads to build spectrum
                     (Default len = 0.8*readLength).
 -localrounds lr     Corrects up to lr errors in a read (Default lr = round(readLength/8)).
                     Reduce if over- and increase if under- corrections are observed.
 -globalrounds gr    Repeat recursively gr times error correction procedure (Default gr = 1).
                     Reduce if over- and increase if under- corrections are observed.
 -qvupdate           Include if generation of a new quality value file is necessary.
 -qvhigh qv          Avoids correction of positions with quality value (>=qv)(Default 25).
 -nosampling         Avoids random sampling in spectrum building. If not included then for large
                     datasets (coverage > 300x) a subset of reads is used in spectrum building.
 -numcores p         If multi-threading is supported then include this option to run the code in
                     p parallel threads.

Developer options:

 -seed t             Size of seed used in spectrum construction (Default is optimal).
 -trustfreq  freq    Use this option to overwrite estimated frequency cutoff of trusted seeds.
                     All seeds with frequency < "freq" are filtered out of spectrum.
 -suppvotes vn       Require at least vn separate votes to fix any position. A vote is cast for
                     a position pos, nucleotide nuc, if a change at (pos,nuc) makes a seed t to
                     belong to spectrum (Default vn = 2, increase if overcorrection is observed).
                     Reduce if over- and increase if under- corrections are observed.
 -outspectxt         Outputs spectrum in txt format in fixed/reads.csfasta.spect.txt.
                     This file includes only seeds with trustable frequencies.
 -outspecdist        Outputs distribution of frequencies in the spectrum.
 -outspecbin         Outputs spectrum in binary format in fixed/reads.csfasta.spect.bin.
                     This file includes seeds with frequency >= 1 (or if more than two blocks are
                     merged, then frequency is >=freq where \"-trustfreq freq\" is provided . It is
                     designed to be loaded later for correction of reads. If this option is
                     included then execution of the program stops after generating the spectrum
                     file (no correction of reads is performed). This option can be used for
                     parallelization by splitting spectrum generation into multiple jobs, each
                     generating a sub-spectrum from the subset of reads.
 -inspecbin files    Uses pre-generated file(s) with spectrum (in binary format) for error
                     correction. Use "," to separate multiple files. All input spectrum files
                     are merged into one spectrum and a frequency cutoff is applied before
```

```
                    correction. All files must have the same seed size. Current reads do not
                    contribute to the spectrum, they are corrected based on input spectra.
                    This option coupled with the above option allows to use spectrum files
                    generated from higher quality set of reads, from multiple sets of reads,
                    or from reference sequences to correct current reads. It can also be used for
                    parallelization by splitting error correction into multiple jobs, each
                    correcting a subset of reads.
-maxtrim  mt        Trims erroneous tails of reads up to first trusted seed or up to "mt". If
                    remaining part of a read is shorter than seed size + 2 then read is discarded.
                    Do not use this option together with -qvupdate option.
-trimqv   tq        Trims erroneous tails of reads up to first trusted seed or up to a position
                    with quality value higher than "tq". If remaining part of a read is shorter
                    than seed size + 2 then read is discarded. Do not use with -qvupdate option.
-log   filename     Outputs execution progress into filename.
```

2.1 Usage of Advanced Options

Depending on post-error correction applications SAET can be tuned to perform more/less aggressive correction, slower but more accurate correction, fewer but more targeted correction. If you trust the quality of your reads more/less than `-trustprefix` then make corresponding changes. The runtime and aggressiveness of error correction mostly depends on `-localrounds` and `-globalrounds`. First parameter allows to correct up to `-localrounds` errors in a read by using pre-computed spectrum. Second parameter recomputes spectrum after each global round and allows to correct up to `-localrounds` errors in a read based on recomputed spectrum. SAET is designed to reduce error rate in the reads generated by SOLiD platform. This increases the number of mapped reads for resequencing projects which can lead to increase in TP and FP SNP-calls. To decrease FP calls, but slightly reduce the number of TP calls, use updated quality value file. Use `-qvhigh` parameter to restrict corrections to positions with quality value bellow `-qvhigh` threshold.

2.2 Usage of Developer Options

If globally computed cutoff for frequency of trusted seeds does not meet your purpose, e.g., it is too low and too many junk seeds are considered correct or it is too high and many correct but low frequency seeds are filtered out, then use `-trustfreq` option to overwrite estimated frequency cutoff. If you noticed that SAET makes many corruptions in the regions of reads with highly packed errors, then, you may increase `-suppvotes` that will tend to correct only isolated errors. SAET provides options for reading and writing spectrum files. That enables building of spectrum from better quality reads and using it to correct lower quality reads, or building spectrum from a reference, or building a spectrum from multiple files (e.g., data from multi-run experiments). In certain applications it is important to trim and filter out error prone reads. Trimming and filtering is enabled by using `-maxtrim` and `-trimqv` options.

2.3 Computational resources

The runtime of SAET depends on the input size and number of global/local rounds. With optimally large number of global/local rounds an expected throughput is 1Gbp per hour. Amount of used RAM should not exceed 2GB.