GEAN manual

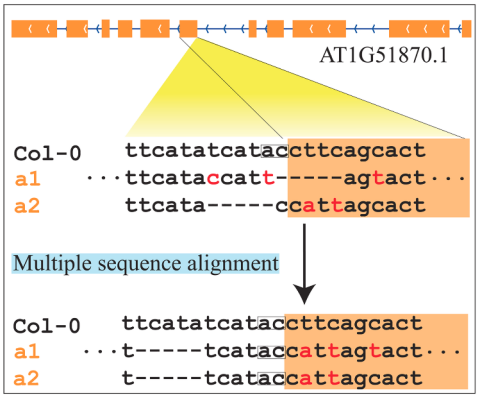
Baoxing Song

2019-09-18

Contents

1 GEAN

Here we provide a solution for INDEL inconsistent alignment problem which could lead to false positive splice sites disturb or ORF-shift predication. And a whole genome MSA pipeline has also been implemented basing on the genic features.



By soving this problem, GEAN could be used to transform the well established genome annotation of model species to the genome of other natural variation individuls or phylogenetically nearby species with whole genome avaliable.

The inconsistent alignment problem could affect the function impact annotation of INDELs (non-coding INDEL V.S. ORF-shift INDEL), SNP (non-coding region SNP, coding region SNP), by re-alignment, those variants could be moved to non-coding regions.

1.1 Install

1.1.1 Dependencies

CPU support avx2 GNU GCC >=6.0 Cmake >= 3.0 Due to the high computational density of weighted sequence alignment algorithm, GEAN only fully works on hadware platform supporting AVX2 CPU constructions. As long as you are not using a very old machine, AVX2 should be valiable.

1.1.2 Installation

```
git clone https://github.com/baoxingsong/GEAN.git
cd GEAN
cmake CMakeLists.txt
make
```

If you would like to move the generated `gean` excusable file to a different path, please put `configure` file and `SpliceSites` under the same folder with the `gean` excusable file.

1.2 Usage

1.2.1 Examples

Examples on different purpose and using genome with different complexity could be found on our github: https://github.com/baoxingsong/GEAN/tree/master/example

```
/home/bs674/software/bin/gean
## Program gean
## Usage: gean <command> [options]
## Commands:
##  -- variant calling:
##     pseudogeno  create pseudo genome sequence
##     lift        transform coordinate to another accession
##     revlift     transform coordinate of another accession to reference
##     liftgff     transform all the GFF/GTF coordinates
##     revliftgff  transform all the GFF/GTF coordinates back to reference
##     reanva      update variants records for functional annotation
##     gff2seq     get the protein/CDS/gene sequence of GFF/GTF file
##     annowgr     annotate re-sequenced genome
##     randomVar   assign a random position for each variant
##
##  -- whole genome wide MSA:
##     premsa      cut the whole genome sequence into fragments
##     msatosdi    generate sdi files from MSA results
##
##  -- de novo assembly genome:
##     transgff    trans reference gff/gtf to de novo assembly genome
##     spltogff    trans reference gff/gtf to de novo assembly genome using sam file
##     purifygff   purify the result from transgff
##     sinsyn      keep syntenic genes priorly and single copy genes (for inner species)
##     sinsyn2     keep syntenic genes priorly and single copy genes (for inter species)
##     quotasyn    quota syntenic blocks
##     orf         keep only ORF conserved genes
##     varcall     variant calling for de novo genome sequence
```

1.2.2 Working on variant calling result

Those functions are designed for whole-genome resequencing variant calling data. It works well for sdi file, which is a very simple file format. For VCF format, you should make sure there is no heterozygous variant calling result. If you are working heterozygous line, you could do phasing and separate your variant calling result into two or more VCF files. Please make sure you have only those variants records pass quality control in the input file. Since the VCF file format diverse from different variant calling software, we could not make our software work with all the VCF files, it is highly recommended to reform your vcf file into sdi format.

Assume you have a vcf download from 1001 Arabidopsis thaliana website (http://1001genomes.org/data/GMI-MPI/releases/v3.1/intersection_snp_short_indel_vcf/intersection_10001.vcf.gz). After gzip, you could run this command to get sdi file.

```
cat intersection_10001.vcf | grep -v "#" | awk '{print "Chr"$1"\t"$2"\t"length($5)-length($4)"\t"$4"\t"$5}' > 10001.sdi
```

1.2.2.1 The sdi (Snps, Deletions and Insertions) file format

Each line of an sdi file consists of the columns chromosome, position, length, reference base, consensus base. The quality value [*0-9] (* or numeric value from 0-255) is optional for GEAN, and will not be used.

- **chromosome** The same name as the chromosome id in reference fasta file
- **position** 1-based leftmost position
- **length** the length difference of the changed sequence against reference (0 for SNPs, negative for deletions, positive for insertions)
- **reference base** [-A-Z]+ (regular expression range)
- **consensus base** [-A-Z]+ ((regular expression range), IUPAC code is used for heterozygous sites

- **quality value** * means no value available; [0-9]+ shows the quality of this variant. It is not necessary Phred quality. Some examples are: Chr1 723 0 C T 2 Chr1 2719 -4 TGCA - 1 Chr1 6786 1 - T 1 Chr1 16786 -4 AGGCA T 1 The columns after the 6th are optional. In the IMR/DENOM output, 7-9th column means: HMQ coverage (the coverage from reads with mapq >=30), SNP Phred score, HMQ consensus base(the consensus base when considering reads with mapq>=30),

1.2.2.2 Get pseudo genome sequence using reference genome sequence and variant calling result

Generate a pseudo genome by substitute the reference allele in reference genome sequence with alternative allele.

```
/home/bs674/software/bin/gean pseudogeno
## Usage: gean pseudogeno -r reference -v variants -o output
## Options
##  -h          produce help message
##  -r FILE     reference genome in fasta format
##  -v FILE     variant calling result in vcf/sdi format
##  -p STRING   prefix for vcf records
##  -o FILE     output pseudo genome in fasta format
```

** -prefix is the prefix of chromosome name for vcf/sdi variant records. Like the chromosome in TAIR10 reference genome is Chr1, Chr2, Chr3, Chr4 and Chr5. While the chromosomes in vcf files from the 1001 genomes project were indicated with 1, 2, 3, 4 and 5. So `-prefix Chr` should be set to make the software work properly. If this parameter is not configured correctly, the program will act as no variant records in the input vcf/sdi file.

1.2.2.3 Liftover reference coordinate to pseudo-genome-sequence

Project/liftover a certain reference genome-sequence coordinate to re-sequencing accession/line with pseudo-genome-sequence avaliable aquired by the above command (pseudogeno). The lift over of the reference genomic coordinates to the pseudo-genome sequences of re-sequenced individuals is performed by counting the number of base pairs shifted by the upstream variants. Use case: I was interested in the haplotype sequence of RDO5 gene in resequenced Arabidopsis population. I got the pseudo-genome-sequence for each individual, and liftover the start codon and stop codon coordinates of Col-0 RDO5 to all other individuals. And extract the haplotype sequence from each pseudo-genome-sequence, then I could perform a multiple sequence alignment.

```
/home/bs674/software/bin/gean lift
## Usage: gean lift -v variants -c chromosome -p position
## Options
##  -h          produce help message
##  -r FILE     reference genome in fasta format
##  -v FILE     variant calling result in vcf/sdi format
##  -f STRING   prefix for vcf records
##  -c STRING   chromosome
##  -p INT      the position/coordinate in reference genome
```

1.2.2.4 Liftover pseudo-genome-sequence coordinate to reference genome sequence

Project/liftover a certain coordinate of re-sequencing accession/line pseudo-genome-sequence to reference genome-sequence by counting the number of base pairs shifted by the upstream variants.

```
/home/bs674/software/bin/gean revlift
## Usage: gean revlift -v variants -c chromosome -p position
## Options
##  -h          produce help message
##  -r FILE     reference genome in fasta format
##  -v FILE     variant calling result in vcf/sdi format
##  -f STRING   prefix for vcf records
##  -c STRING   chromosome, should be consistent with the chromosome information in sdi file (The coordinate starts from 1)
##  -p INT      the position/coordinate in re-sequenced genome
```

1.2.2.5 Liftover reference gff/gtf/gff3 annotation to pseudo-genome-sequence

Lift over the reference genome annotation (gtf/gff file) to a re-sequencing accession/line pseudo-genome-sequence by purely coordinate liftover of each boundary coordinate.

```
/home/bs674/software/bin/gean liftgff
## Usage: gean liftgff -v variants -i inputGffFile -o outputGffFile
## Options
##  -h        produce help message
##  -r FILE   reference genome in fasta format
##  -v FILE   variant calling result in vcf/sdi format
##  -i FILE   the input GFF/GTF file of reference line/accession
##  -f STRING prefix for vcf records
##  -o        the output GFF/GTF file of target line/accession
```

1.2.2.6 Liftover pseudo-genome-sequence gff/gtf/gff3 annotation to reference genome sequence

Project/liftover the gene structure (gtf/gff file) annotation of re-sequencing accession/line to reference genome-sequence by purely coordinate liftover.

```
/home/bs674/software/bin/gean revliftgff
## Usage: gean revliftgff -v variants -i inputGffFile -o outputGffFile
## Options
##  -h        produce help message
##  -r FILE   reference genome in fasta format
##  -v FILE   variant calling result in vcf/sdi format
##  -i FILE   the input GFF/GTF file of non-reference line/accession
##  -f STRING prefix for vcf records
##  -o        the output GFF/GTF file of reference line/accession
```

1.2.2.7 Recall variants by align the genic region sequencing and keep the completeness of ORF

Realign the pseudo-genome sequence using ZDP algorithm to solve the inconsistent INDEL alignment problem and recall all variants to avoid false positive ORF-state shift predication. This function assummes all the coordinates are 1-based. Except that point, it works well with bed files

```
/home/bs674/software/bin/gean reanva
## Usage: gean reanva -i inputGffFile -r inputGenome -a similar segments -s new genome sequence -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   GFF/GTF file
##  -r FILE   reference genome sequence
##  -v FILE   variant calling result in vcf/sdi format
##  -f STRING prefix for vcf records
##  -o FILE   output file
##  -m INT    minimum intron size
```

* By ORF-states, this software has following criteria:

- 1) Splicing sites is one of motif in "SpliceSites" file, which is included in the release
- 2) The minimum length of intron is larger than a certain value
- 3) CDS sequence length is larger than a certain value
- 4) The length of CDS sequence is divisible by 3
- 5) No premature stop codon
- 6) End with end codon
- 7) Start with start codon

The IUPAC Codes of DNA sequence could be well dealt with. The result of ORF-states are included in the CDS sequence

1.2.2.8 Extract sequece using genome sequence and annotation file

Extract CDS sequence, C-DNA sequence and protein sequence for each protein-coding transcript. And predict the protein coding potential (termed as ORF-state). In the output CDS sequence fasta file, could tell the ORF-state, and why predicte it as ORF-state conserved/shift.

```
/home/bs674/software/bin/gean gff2seq
## Usage: gean gff2seq -i inputGffFile -r inputGenome -p outputProteinSequences -c outputCdsSequences -g outputGenomeSequences
## Options
##  -h        produce help message
##  -i FILE   reference genome in GFF/GTF format
##  -r FILE   genome sequence in fasta format
##  -m INT    minimum intron size for ORF stats checking
##  -p FILE   output file of protein sequence in fasta format
##  -c FILE   output file of CDS (without intron) in fasta format
##  -g FILE   output file of CDS (with intron) in fasta frormat
```

1.2.2.9 Annotate the pseudo-genome-sequence

Transform the reference gene structure annotation to re-sequencing accession/lines with several complementary methods. The approaches embed in this function are:

- 1) standard coordinate coordinate liftover.
- 2) ZDP approach.
- 3) exonerate CDS alignment
- 4) exonerate protein alignment
- 5) other genome annotation (you could obtain is using ab initio annotation like Augustus, or RNA-seq guided genome annotation)

For the first 4 approach, the gene structure predicated by the upstream module would be adapted firstly, and the region predicted as ORF-state shift would be handled by the below modules. The genes mode in 5th module located in region that could not find a ORF-state conserved region in the first 4 modules will be ingegreted into the finall output.

```
/home/bs674/software/bin/gean annowgr
## Usage: gean annowgr -i inputGffFile -r referenceGenomeSequence -v variants -o outputGffFile
## Options
##  -h        produce help message
##  -i FILE   reference GFF/GTF file
##  -n FILE   the de novo annotation GFF of the target accession
##  -r FILE   reference genome in fasta format
##  -v FILE   variant calling result in vcf/sdi format
##  -o FILE   the output GFF/GTF file
##  -m INT    minimum intron size
##  -d        remove reference ORF shift transcripts (default false)
##  -f STRING prefix for vcf records
##  -t INT    number of threads, default: 4
##  -l INT    longest transcript to align. default(50000)
```

1.2.2.10 Simulate random variants

Assign a random position for each variant in a variant calling result file, which was used to compare the different between observed variant calling and random variants.

```
/home/bs674/software/bin/gean randomVar
## Usage: gean generateRandomSdi -v variants
## Options
##    -h        produce help message
##    -r        (string) reference genome in fasta format
##    -v        variant calling result in vcf/sdi format
##    -o        prefix of output file
```

1.2.3 Whole genome wide multiple sequence alignment pipeline

The function implemented here is an updated version of the functions implemented in Irisas. The functions implemented are faster and consider genome annotations.

- Fistly, cut the whole genome sequence into fragments.
- Secnodly, Perform MSA on each fragments.
- Finally, GEAN merge all the MSA alignment fragments into whole genome level, perform variant calling and output sdi files.

Please reference Iriasa document for the whole pipeline.

1.2.3.1 Cut the (pseudo-)genome sequence of a population of individuals into fragments to perform multiple sequence alignment for each fragment.

```
/home/bs674/software/bin/gean premsa
## Usage: gean premsa -i inputGffFile -r referenceGenomeSequence -v variants
## Options
##  -h         produce help message
##  -i FILE    the input GFF/GTF file of reference line/accession
##  -r FILE    reference genome
##  -v FILE    list of variant calling results files
##  -f STRING  prefix for vcf records
##  -m INT     minimum intron size
##  -t INT     number of threads, default: 4
##  -w INT     window size, default: 10000
##  -s INT     window overlap size, default: 500
##  -p INT     output catch size (default 100)
##  -l INT     longest transcript to align. default(50000)
```

1.2.3.2 Variant calling using multiple sequence alignment

Perform variant calling from the multiple sequence alignment of sequence fragments of a population of genome sequences

```
/home/bs674/software/bin/gean msatosdi
## Usage: gean msatosdi -a accessionList -c chromosomeLi -m MSAresultFolder -o outputFolder  -r referenceGenomeSequence -v variants
## Options
##  -h         produce help message
##  -c FILE    chromosome list
##  -m FOLDER  folder of MSA result
##  -o FOLDER  output folder
##  -r FILE    reference genome in fasta format
##  -t INT     number of threads, default: 4
##  -v FILE    list of variant calling results files
##  -f STRING  prefix for vcf records
```

1.2.4 Project reference annotation to de novo assemly genome sequence

Pipeline to project the reference gene structure annotation to a de novo assembly genome sequence highly similar with the reference genome sequence

1.2.4.1 Liftover reference genome annotation to a de novo assembly genome using whole genome alignment result

Liftover reference genome annotation to a de novo assembly genome sequence using whole genome sequence alignment. This function perform standard sequence alignmetn to lift over genome annotation firstly, and then complement using ZDP approach.

The result file contains duplication gene annotations records, which might do not compile with other software and could be purified with the following function.

```
/home/bs674/software/bin/gean transgff
## Usage: gean transgff -i inputGffFile -r inputGenome -a similar segments -s new genome sequence -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   reference GFF/GTF file
##  -r FILE   reference genome sequence
##  -a FILE   similar segments
##  -s FILE   target genome sequence
##  -o FILE   output GFF/GTF file
##  -w INT    sequence alignment window width (default: 60)
##  -sl       run in slow model (default false)
##  -l INT    longest transcript to align. default(50000)
##  -m INT    minimum intron size
```

1.2.4.2 Liftover reference genome annotation to a de novo assembly genome using CDS sequence mapping

This function implemented similar function as the above on, but it takes the minimap2 sam output as input. Usage exmple is avaliable.

```
/home/bs674/software/bin/gean spltogff
## Usage: gean spltogff -i inputGffFile -r inputGenome -a similar segments -s new genome sequence -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   reference GFF/GTF file
##  -r FILE   reference genome sequence
```

```
##  -a FILE   sam file
##  -s FILE   target genome sequence
##  -o FILE   output GFF/GTF file
##  -g INT    output tag, should be 0, 1 or 2 (default 1)
##            0 prefer to output the ZDP realignment result
##            1 prefer output the standard alignment result
##            2 prefer output the longer result
##  -w INT    sequence alignment window width (default: 60)
##  -l INT    longest transcript to align. default(50000)
##  -m INT    minimum intron size
```

1.2.4.3 Liftover reference genome annotation to a de novo assembly genome using CDS sequence mapping

This function implemented similar function as the above on, but it takes the minimap2 sam output as input. Usage exmple is avaliable.

```
/home/bs674/software/bin/gean spltogff
## Usage: gean spltogff -i inputGffFile -r inputGenome -a similar segments -s new genome sequence -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   reference GFF/GTF file
##  -r FILE   reference genome sequence
##  -a FILE   sam file
##  -s FILE   target genome sequence
##  -o FILE   output GFF/GTF file
##  -g INT    output tag, should be 0, 1 or 2 (default 1)
##            0 prefer to output the ZDP realignment result
##            1 prefer output the standard alignment result
##            2 prefer output the longer result
##  -w INT    sequence alignment window width (default: 60)
##  -l INT    longest transcript to align. default(50000)
##  -m INT    minimum intron size
```

1.2.4.4 Remove duplication genome annotation records

Remove those duplication gene structure annotations generated from the transff function

```
/home/bs674/software/bin/gean purifygff
## Usage: gean purifygff -i inputGffFile -s inputGenome -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   GFF/GTF file
##  -s FILE   target genome sequence
##  -o FILE   output GFF/GTF file
##  -x INT    minimum gene length
##  -m INT    minimum intron size
```

1.2.4.5 Syntenic protein coding gene analysis for the whole chromosome

This function perform syntenic analysis for the whole chromosome for the whole genome using the longest path approach. It tried to align all the genes on the whole chromosome and good for the analysis that assume no large scale genome re-arrangement. The algorith works very well for a few of reversions. The algorithm implemented here is a modified version of longest path. The same algorithm has been implemented in the Practical Haplotype Graph by us. And it was used for the syntenic Arabidopsis col-0 and ler-0 in our manuscript.

```
/home/bs674/software/bin/gean sinsyn
## Usage: gean sinsyn -i referenceGffFile -s inputGenome -a inputGffFile -o output GFF/GTF file
## Options
##  -h          produce help message
##  -i FILE     reference GFF/GTF file
##  -s FILE     target genome sequence
##  -a FILE     target GFF/GTF file
##  -o FILE     output GFF file
##
##  ADVANCED PARAMETERS
##  -m INT      minimum intron size
##  -d          keep tandem duplication (default false)
##  -on         only output syntenic bolck genes (default false)
##  -rt DOUBLE  reverse syntenic block threads hold (12.0)
##  -rs DOUBLE  reverse syntenic match score (3.0)
##  -rp DOUBLE  reverse syntenic mismatch penalty (-4.0)
##  -ss DOUBLE  score for gene located in syntenic region (1.0)
##  -so DOUBLE  score for gene with ORF conserved (1.5)
##  -dl DOUBLE  length ratio to drop a gene transformation (0.2)
```

1.2.4.6 Syntenic protein coding gene analysis for local regions

Perform syntenic analysis for the local regions. It assumes large scale genome re-arrangements. And the output syntenic are more fragmented. This function is an re-implementation of DAGchainer This fucntion was used to the syntenic analysis of Arabidopsis and Cardamine hirsuta.

```
/home/bs674/software/bin/gean sinsyn
## Usage: gean sinsyn -i referenceGffFile -s inputGenome -a inputGffFile -o output GFF/GTF file
## Options
##  -h          produce help message
##  -i FILE     reference GFF/GTF file
##  -s FILE     target genome sequence
##  -a FILE     target GFF/GTF file
##  -o FILE     output GFF file
##
##  ADVANCED PARAMETERS
##  -m INT      minimum intron size
##  -d          keep tandem duplication (default false)
##  -on         only output syntenic bolck genes (default false)
##  -rt DOUBLE  reverse syntenic block threads hold (12.0)
##  -rs DOUBLE  reverse syntenic match score (3.0)
##  -rp DOUBLE  reverse syntenic mismatch penalty (-4.0)
##  -ss DOUBLE  score for gene located in syntenic region (1.0)
##  -so DOUBLE  score for gene with ORF conserved (1.5)
##  -dl DOUBLE  length ratio to drop a gene transformation (0.2)
```

1.2.4.7 Syntenic protein coding gene for genomes with different whole genome duplication history

This function could be used for two genome with different whole genome duplication and parameters could be tuned for homologous genes have different copies. Perform syntenic analysis using local regions. It assumes large scale genome re-arrangements.

```
/home/bs674/software/bin/gean quotasyn
## Usage: gean quotasyn -i referenceGffFile -s inputGenome -a inputGffFile -o output GFF/GTF file
## Options
##  -h          produce help message
##  -i  FILE    reference GFF/GTF file
##  -s  FILE    target genome sequence
##  -a  FILE    target GFF/GTF file
##  -o  FILE    output GFF file
##
##  ADVANCED PARAMETERS
##  -m  INT     minimum intron size
##  -mq INT     maximum gene copies in the query
##  -d          keep tandem duplication (default false)
##  -on         only output syntenic bolck genes (default false)
##  -r          whether sort output with coordinate (default: false, only make sense when -on set as true)
##  -md INT     maximum distance of syntenic block genes (default 20)
##  -ms DOUBLE  minimum syntenic block score (default 6.0)
##  -op DOUBLE  open gap penalty (default -0.1)
##  -ep DOUBLE  extend gap penalty (default -0.05)
##  -ss DOUBLE  score for gene located in syntenic region (1.0)
##  -so DOUBLE  score for gene with ORF conserved (1.5)
##  -dl DOUBLE  length ratio to drop a gene transformation (0.2)
```

1.2.4.8 Keep only ORF conserved genes from GFF file

Remove all the ORF shifted genomoe annotation records from the input gff file

```
/home/bs674/software/bin/gean orf
## Usage: gean orf -i inputGffFile -s inputGenome -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   GFF/GTF file
##  -s FILE   genome sequence
##  -o FILE   output GFF/GTF file
##  -m INT    minimum intron size
```

1.2.4.9 Base pair level variant callnig for de novo genome sequence

Perform base-pair level variant for de novo genome sequence and out put a sdi file. The de novo genome should be in fasta file, and in chromosome level. And the homologous chromosome should share the same entry ID with the reference genome fasta file. It assumes there is no re-arrangement. By base-pair level variant calling, we mean given the reference genome and the variant calling result, GEAN could infer the query genome sequence without any error.

```
/home/bs674/software/bin/gean varcall
## Usage: gean varcall -i refGffFile -r refGenome  -t targetGff -s targetGenome -o output GFF/GTF file
## Options
##  -h        produce help message
##  -i FILE   reference GFF/GTF file
##  -r FILE   reference genome sequence
```

```
##  -t FILE   target GFF/GTF file
##  -s FILE   target genome sequence
##  -o FILE   output file
##  -x INT    minimum gene length
##  -w INT    sequence alignment window width (default: 60)
##  -m INT    minimum intron size
```

1.2.5 Acknowledgements

The GEAN development team would like to thank all our enthusiastic users who have contacted us with suggestions to improve the codebase, request new functions, point out bugs, and beta-test the initial versions of GEAN. Many thanks also to everyone who has used GEAN and cited the publication – we are glad it has proven useful in your research, and a good citation record will help us to obtain fundings to keep developing GEAN.

1.2.6 Citation

If you use GEAN, please cite: `Baoxing Song, Qing Sang, Hai Wang, Huimin Pei, Fen Wang and Xiangchao Gan. (2019) A weighted sequence alignment strategy for gene structure annotation lift over from reference genome to a newly sequenced individual. bioRxiv. doi:10.1101/615476`