Amplicon Indel Hunter (AIH/AID)

aiHunter v1.1.0

Sabah Kadri*[1,2], Chao Jie Zhen[1], Michelle N Wurst[1], Bradley C Long[1], Zifeng F Jiang[1], Y. Lynn Wang[1], Larissa V Furtado[1], and Jeremy P Segal†[1]

[1]Division of Genomic and Molecular Pathology, Department of Pathology, The University of Chicago, Chicago, Illinois
[2]Center for Research Informatics, The University of Chicago, Chicago, Illinois

Accurate detection of large insertions and deletions (indels) via amplicon-based targeted NGS assays remains a challenge when depending on alignment-based methods. Sequencing reads that cover these indels are, by definition, different from the reference sequence, and lead to variable performance of alignment algorithms. Amplicon Indel Hunter (AIH) is a large (>5-bp) indel detection method that is reference genome independent and highly sensitive for the identification of somatic indels in amplicon-based, paired-end, NGS data. The software (aiHunter) takes as input paired end fastq files and information about the amplicons in the assay, and detects amplicons with potential large indel. The output from AIH is given as input to a helper tool, Amplicon Indel Diagnoser (AID) which tries to annotate the exact indel sequence and returns output in VCF format.

Publication (please cite if using software):

Sabah Kadri, Chao J. Zhen, Michelle N. Wurst, Bradley C. Long, Zi-Feng Jiang, Y. Lynn Wang, Larissa V. Furtado, Jeremy P. Segal, **Amplicon Indel Hunter Is a Novel Bioinformatics Tool to Detect Large Somatic Insertion/Deletion Mutations in Amplicon-Based Next-Generation Sequencing Data**, The Journal of Molecular Diagnostics, Volume 17, Issue 6, November 2015, Pages 635-643, ISSN 1525-1578,
http://dx.doi.org/10.1016/j.jmoldx.2015.06.005.
(http://www.sciencedirect.com/science/article/pii/S1525157815001506)

*skadri@bsd.uchicago.edu
†jsegal5@bsd.uchicago.edu

Contents

1 Release Notes

v1.1.0 has the following changes over v1.0.0

1. **Bug fix in VCF file generation**
If AID fails to decode the exact indel, due to bad quality or the size of the indel, it returns a `Failed indel identification - manual review required for <amplicon name>` line in the <read1 filename>.R2fastq.finalindelstats file. In v1.0.0, the VCF file generation was failing if there was a decoding failure by AID. This is now fixed. The VCF file will only contain entries for the indels which are successfully decoded by AID. However, we encourage all users to also investigate the <read1 filename>.R2fastq.finalindelstats file to manually check entries which failed with AID.

2. **Example data**
Example data, and its associated documentation is now included in the aiHunter folder. Please see the folder `example_data_v1.1.0`.

3. **New FAQ section**
We have added a new FAQ section to this documentation, based on common issues faced by some of our users. Please refer to the FAQ section for better understanding of any errors encounters. Please feel free to reach out **Sabah Kadri (skadri@bsd.uchicago.edu)** for any additional troubleshooting.

2 Quickstart

A quick way to start using aiHunter is to have paired end Fastq files, an infosheet with details of the assay, and a fasta file with the insert sequences. Please see "Input Data" below for more details.

```
python aiHunter.py --read1 <Read1 Fastq> --read2 <Read2 Fastq> --amp
<Amplicon information sheet> --inserts <Inserts Fasta file> [options]
```

Please see Section 5 for more information about the correct formatting of the amplicon info sheet and inserts fasta file.

3 Installation

`aiHunter` has been tested with python v2.7.6. It requires the following packages in the python PATH to run it: (1) argparse (2) os (3) sys (4) time (5) Bio (biopython) (6) math

An executable binary is also provided with its own documentation in the binary_distribution folder. This can only be run on the Linux OS.

4 Overview of AIH/AID

The aiHunter software accepts as input a pair of fastq files from the amplicon NGS assay and a primer information sheet that includes the sequences of the assay primer pairs and the expected length of each amplicon product (See Input data below). The software first runs AIH to identify amplicons with potential large indels and then runs the annotator AID too to identify the indel, if possible.

For each pair of on-target reads, the correct primer pair for the amplicon is identified from the beginning of the forward and reverse read sequences, and the expected read overlap offset is calculated. The read mates are compared against each other using the expected offset, with the expectation of high-quality matching. If an amplicon insert includes any length affecting mutation, paired sequence overlap at the expected offset will be severely affected, decreasing from perfect or near-perfect matching to matching at the level of random chance. A matching threshold of 90% is used.

AIH is built with a custom cushion parameter to increase specificity. This is also the lower bound on the size of indels that AIH can detect. Thus, matching is attempted at the expected offset, and also at any offset that differs from the expected offset by less than the cushion value, with any successful match deemed acceptable. The default cushion value is 5, but it can be input by the user (See Optional Parameters below). On completing the analysis of every read pair in the files, summary statistics are produced for each amplicon, indicating the number of read pairs identified, and the fraction of reads that failed to successfully overlap at the expected offset, which is the proportion of reads likely to harbor a larger indel. Any amplicons with >5% MAF (by default) are flagged as potentially mutant amplicons. The MAF threshold can also be input by the user.

Once the potential mutant amplicons are identified by AIH using the user-defined MAF threshold, the helper tool, AID, is used to annotate the mutant indel and determine its genomic location. AID selects the read pairs belonging to each potential indel-containing amplicon with a 90% match threshold and calculates for each read pair, the expected overlap offset $\text{Offset}_{Expected}$ and the $\text{Offset}_{Mutated}$. The $\text{Offset}_{Mutated}$ is calculated by sliding the read mates across each other to find the best overlap between the sequences, using only bases with quality greater than Q25. In the case of an amplicon with a large indel,

$$\text{Indelsize}=\text{Offset}_{Mutated} - \text{Offset}_{Expected}$$

The most abundant indel size that is outside the cushion buffer is selected, and the consensus mutant amplicon sequence is calculated for read pairs with this offset. AID processes the consensus mutant indel-containing amplicon sequence, comparing it with the amplicon reference sequence, which is calculated from the Inserts Fasta and Primer sequences, to establish the details of the indel. Chromosomal locations are calculated from this result, using the known genomic coordinates of the amplicon.

5 License

See License.txt

6 Input Data

aiHunter requires the following input files in order to run:

`--read1`	Read1 Fastq file.
`--read2`	Read2 Fastq file.
`--amp`	Tab-delimited amplicon info file. See below.
`--inserts`	Fasta file with insert sequences. See below.

6.1 Amplicon Info file

Please make sure that the coordinates listed in this file are accurate. These will be used by the Amplicon Indel Diagnoser program to identify the genomic coordinates of the indel.
Note: Please make sure that the file has a header line.
Tab delimited file with following columns:

Column name in header	Description
`AmpliconName`	Name of the amplicon. Depends on assay.
`Primer1Sequence`	Primer1 sequence. Should be + strand genomic sequence.
`Primer2Sequence`	Primer2 sequence. Should be "-" strand genomic sequence, that is, reverse complement of the sense sequence.
`AmpliconLength`	Length of the amplicon. Can be calculated as (AmpliconStop - AmpliconStart + 1) or (Primer2End - Primer1Start + 1)
`Chr`	Reference Chromosome
`GenomicAmpliconStart`	Genomic Amplicon Start, which is the same as Primer1 Start
`GenomicAmpliconEnd`	Genomic Amplicon End, which is the same as Primer2 End

6.2 Inserts Fasta file

Amplicon Indel Diagnoser part of aiHunter requires a fasta file with the insert sequences of the assay in order to determine the exact sequence of the indel,

wherever possible. This file can be generated using by making a BED file of the inserts and then using UCSC genome browser in order to download the Fasta file.
Note: Please make sure your inserts bed file has
Chr Primer1End and (Primer2Start - 1) in order to get the correct sequence from UCSC
Note: Please make sure that the amplicon names in the inserts fasta file match the amplicon names in the amplicon info file

7 Optional parameters

`--help` or `-h`	Outputs a quick help manual for running aiHunter.
`--version` or `-v`	Provides version information.
`--info`	Quick information about the Amplicon Info file
`--out OUTDIR`	Output directory for the results. Default = Current working directory.
`--cushion CUSHION`	Cushion Value (default: 5). Please see Section 3 or publication for more details.
`--maf SIGNIFICANCE`	Mutant allele frequency threshold (default: 0.05)

8 Outputs from AIH and AID

8.1 Output files and formats

aiHunter outputs the following four files for each pair of fastq files. The main output file is <**read1 filename**>**.R2fastq.finalindelstats.vcf** :

1. <**read1 filename**>**.R2fastq.indelcalls.txt:**
   This is the output of the AIH module of the software. It is a table for one row for each amplicon. The four columns and their descriptions are given below.

`Amplicon`	Name of amplicon as provided in the input info file
`Reads_w_indel>5bp`	Number of Reads with potential indel of size greater than the cushion size. The 5bp represents the default cushion size, and even if a different cushion is provided, this column header remains the same.
`#readpairspassingfilter`	Number of reads which match the primer pair for the specific amplicon with minimum identity of 90%
`%indel`	Column#2 / Column#3

2. **<read1 filename>.R2fastq.indelcalls.significant.txt:**
   A list of amplicon names that pass the MAF threshold (See `--maf SIGNIFICANCE`). Each of these are investigated by the AID module of the software.

3. **<read1 filename>.R2fastq.finalindelstats:**
   Output of the AID module of the software. The file has no header. It is a 8-column tab-delimited text file with the following columns.
   ```
   chrom  position  depth  depth  ref_allele  ref_freq  indel  indel_freq
   ```

4. **<read1 filename>.R2fastq.finalindelstats.vcf:**
   Output of the AID module of the software. A VCF formatted output of the annotated (if possible) indels.

8.2 Indel Annotation Failure

In cases where the insertions are so large that there is little or no longer overlap between the read mates, AID might fail in determining the exact indel sequence and output:

```
Failed indel identification - manual review required for <amplicon
name>
```

In such cases, we usually use a combination of the detailed logs from AID and inspection of the misaligned read alignments in IGV to manually determine the indel sequence. This is not a frequent occurrence and only affects insertions.

8.3 Bad quality Problems

Please note a bad sequencing run with deteriorating base qualities in a large portion of the read may affect the sensitivity of detection by this method, depending on the amplicon and read lengths.

9 FAQs

Q. The finalindelstats file returns `Failed indel identification - manual review required for <amplicon name>`

A. This can happen when the insertion is so large that there is little or no overlap between the read mates. In this case you might want to do a manual review. In such cases, we usually use a combination of the detailed logs from AID and inspection of the misaligned read alignments in IGV to manually determine the indel sequence.
This might also be due to a bad sequencing run with deteriorating base qualities in a large portion of the read may affect the sensitivity of detection by this method, depending on the amplicon and read lengths.

Q. The indelcalls.txt shows 0 counts for all amplicons.

A. There might be a problem with the input amplicon info file. Please recheck the format of the file, making sure that the Primer 1 sequence is in the genomic sense "+" direction and Primer 2 sequence is in the anti-sense "-" direction, that is, the reverse complement of the genomic sequence. *Hint: You can use UCSC Insilico PCR tool to check the orientations of the primer sequences*

Q. The log file returns an error of "No sequence found for <amplicon name>" and the finalindelstats file is empty even though the indelcalls.txt has high MAF for the amplicon.

A. Please make sure the insert sequences for each amplicon is present in the Fasta file, and that the names of the amplicons are the same between the fasta file and the amplicon info file.

Q. I am getting various string-related errors when I try to run the program.

A. Please make sure:

1. The inserts.fa file has the Inserts - sequence of the amplicon between the primers and does not include the primer sequences.
2. Sequences for each amplicon are present in the Fasta file, and that the names of the amplicons are the same between the fasta file and the amplicon info file.
3. The Fastq files are not zipped. The current version of aiHunter only works with unzipped fastq files.
4. All reads are of the same length, that is, the adapters are not trimmed off, either by the software on the instrument or other data pre-processing.