

AB Small InDel Tool Documentation

1	OVERVIEW	2
2	INSTALLATION	3
2.1	Prerequisites	4
2.2	Installation procedure	4
3	AB SMALL INDEL TOOL.....	4
3.1	Usage Parameters - Required	5
3.2	Usage Parameters - Optional	5
3.3	Upstream pairing pipeline considerations (pairing_by_group.pl)	6
3.4	Usage Examples	7
3.5	List of programs/scripts included.....	8
3.6	Other scripts called by this program.....	8
3.7	Path constraints	8
3.8	System Input Files	8
3.9	Input File Versions supported.....	8
3.10	Output Files.....	8
3.11	Output Files Description.....	9
3.12	Sample Files.....	11
4	INDEL POST-PROCESSING PERFORMED IN PAIRING PIPELINE.....	12
4.1	Removal of highest insertion size	12
4.2	Sorting and adding run identifiers.....	12
4.3	Skip recreating the mates file.....	12

1.0	07/10/2009	
-----	------------	--

1 Overview

The AB Small InDel Tool processes the indel evidences found in the pairing step of the SOLiD™ 3 System Analysis Pipeline Tool ('Corona Lite'). In the pairing pipeline, these evidences are found in mate pairs by using the placement of one tag such that the other tag can be used to search for the presence of a small indel. Specifically, with the unplaced tags, the algorithm finds all possible placements of the beginning and ends of the read. Then, these will be joined together only if the resulting gap size is within the limits set in the pairing pipeline. All successful joinings form the set of gap alignments. These are referred to as the **indel evidences**.

The AB Small InDel Tool provides a front-end script that allows flexible processing of these indel evidences. It does so by taking pairing indel results and combining them (based on their proximity from each other) to form candidates. Once these combined results are formed, the tool allows for the combination of several runs together, and for filtering based on the average read position of the indel and number of non-redundant reads. The tool also allows for easy downstream analysis by producing the candidates in a text format useful for importing into databases and in a public GFF version 3 format.

Algorithm Description of Determining Evidences

Searching for indels across the human genome on individual reads is computationally intensive. The algorithm surveyed only those indels with one-end anchored (OEA) mate pairs. It does so by realigning the OEA pairs using the anchored pair which has a placement in the genome as a seed and performed a more aggressive alignment with the other tag in a several kb window (depending on the minimum and maximum insert sizes set in pairing) around the anchored mate.

Using the unanchored tag, it starts aligning both ends of the read until the maximum of number of mismatches occur (this defaults to 2 and settable in pairing by `--error_indel`).

The algorithm disallows for indels within 3 bases from either end of the read. It identifies if it is able to piece together both ends, allowing only for a single gap of up to 4 base pairs inserted (present in read but not in reference), or up to 11 base pairs deleted. (All of these values are adjustable in pairing by `--indel_parameters`.) To reduce edge effects caused by having a cut-off value in indel finding, it filters out insertions of size 4. Furthermore, it identifies it as an indel candidate if the above joining could be done with the fewest (up to a maximum of two) number of mismatches and identify the location of it by where this joining occurred. Ambiguity of this location was common, but was reduced by checking the color space compatibility of the two sequences that the gap traversed.

When an indel occurs in a sequence, and that sequence is measured using color space, the color space sequence not only has a gap of the same size of the indel, but also leaves a signature that can indicate if there is a measurement error within the gap. This is especially important in the case of insertions when you have small number of evidences and there is a disagreement in the bases of the inserted sequence. With methods that directly measure bases, there would be no indication, based on the inserted sequence alone, on which inserted bases is more trusted. In color space, this

signature can be used to see if the color that spans the gap is compatible with set of colors that go through the gap. For example, the alignment, AACG/A--G, would be 013/2-- in color space. Here the color 2 spans the gap (measuring both A and G), while 013 goes through the gap (measuring AACG). The color 2 is compatible with the sequence 013 because they both would end with a G in base space. However, an alignment of 213/2-- would not be compatible because, using the same starting base A as the above example, 213 would measure AGTA. Because the rest of the color space sequence beyond this would be aligned, 213/2-- would be indicative of a measurement error within the indel. The alignment's color space compatibility can be calculated for any sequence using color space addition. This signature for color space compatibility is used here to attempt to resolve ambiguity in the location of an indel for a given read and can also be used to better call the actual inserted sequence of the indel when a set of reads the indicate the same event are being considered.

Combining of Evidences and Filtering

This tool enables the analysis of indel evidences (`indel-evidence-list.pas`) that come from the pairing pipeline. These evidences indicate reads where a gap was detected. If there was an indication that the read came from a good mate, it is usually excluded. However using the `--indels_in_good_mates` option in the pairing pipeline allows for these to be included as well.

The evidences (even from multiple runs) are combined together and must have between `--min_num_evid` and `--max_num_evid` (inclusive) number of evidences. If evidences are within 5 base pairs between consecutive evidences, then they are combined together until there are already 5 evidences, then additional ones will only be grouped together if it is 2 or fewer base pairs from the last evidence. The result of this is the evidence file for a run.

Coverage Issue: The ability to handle high coverage situations is new for version 1.1.0 (4.2 of Corona Lite). Because only two non-redundant reads are required to make an indel call, false positives can become prevalent at higher coverage levels. By using information of non-gapped normal coverage (using the `F3_R3.mates.non-redundant` file), the software can determine a coverage ratio of normal clipped coverage over a number of non-redundant indel supports. A high ratio indicates a false positive. These are filtered at values higher than 12 by default.

Multiple runs can be combined together from several evidence files using this tool. From this combined evidence file from one or more pairing runs, the tool allows for further filtering. The first set is based on read position, basically filtering out candidates that are all found towards the end of the read which is more error prone. Note: For a 25 base-pair read, `--min-from-end-pos 9.1` and `-max-ave-read-pos 15.9` yield the same result. The default is to do this filtering on candidates that have 2 or fewer evidences to form the candidate. This can be adjusted by setting `--max-nonreds-4filt`.

The other filters are for indel sizes and outputting those where an indel size call can be made. An indel call is made if 75% of the reads or more indicate the same indel size.

2 Installation

The installation instructions apply to the AB Small InDel tool package. Small InDel tool is installed automatically with the Corona Lite Plus package.

2.1 Prerequisites

Before installation, verify that your system meets the following requirements:

- Linux CentOS 4 – The program has been built and tested on Linux CentOS 4 using GCC 3.4, Perl 5.8.5 and GNU Make 3.8.
- An existing Corona Lite installation, version 4.2 or better.

2.2 Installation procedure

1. Extract the `Small_Indel_<version>.tar.gz` package into the desired location:

```
$ tar xzvf Small_Indel_<version>.tar.gz
```

2. Enter the subdirectory created, and install into an existing CORONAROOT:

```
$ cd Small_Indel_<version>
$ make install PREFIX=$CORONAROOT
```

3. To test the installation run `make test`. Make sure that a proper Corona Lite environment is available; `PATH`, `PERL5LIB`, and `PYTHON` path should be set according to the instructions for Corona Lite.

Do not install the AB Small InDel Tool v1.0.1 or earlier version onto Corona Lite version 4.1r1.0 or later!

Do not install v. 1.1.0 onto Corona Lite version 4.0r2.0 or earlier. These can both corrupt the Corona Lite installation!

The package has only been tested on the LINUX 64-bit O/S platform. It is possible to compile for other UNIX platforms. To rebuild the LISP code, install Steel Bank Common Lisp (SBCL) for your specific platform, and enter the following:

```
$ cd $CORONAROOT/bin
$ sbcl
* (load "process-small-indels.lisp")
* (build)
```

The resulting `process-small-indels` is a new binary for your platform. A quick test is if you see the version number, build date, and build time when you do.

```
$ exec_lisp.pl -f splash
```

3 AB Small InDel Tool

Program Name: `small-indel-tool.pl`

Program Version: 1.1.0

Development Languages: Perl, Lisp

Compiled for: LINUX 64-bit O/S

PBS Is Required: No parallelization is required to run this script. However, upstream indel finding in `pairing_by_group.pl`, may require this.

Supports AB kit or protocol or sample prep method: N/A

Date: July 8, 2009

3.1 Usage Parameters - Required

Input files/directories

<code>--pairing-dir</code> <code>-pd</code>	Pairing director(ies), in which evidences (indel-evidence-list.pas) and mates file(s) (F3_R3.mates.non-redundant) can be found.
---	---

Or

<code>--evidence-fn</code> <code>-e</code>	Input indel evidence file name(s).
<code>--mates-fn</code> <code>-m</code>	Non-redundant mates file name(s). (Required for accurate calls in high-coverage situations.)

Note: For multiple files or directories, separate items with commas and no spaces.

Output file

<code>--candidate-fn</code> <code>-c</code>	Produces .gff, .txt, and .sql output files using this name.
---	---

3.2 Usage Parameters - Optional

Options for combining evidences	
<code>--min_num_evid</code>	Min. number of evidences required for an indel call. Default: 2.
<code>--max_num_evid</code>	Max. number of evidences, -1 means no max. limit. Default: -1.
<code>--consGroup</code>	Indel grouping method: <ul style="list-style-type: none"> • 1 for conservative grouping, which has 5 bp maximum between consecutive evidences (Default) • 0 for the higher of 15 or (7*indel size) maximum between evidences. • 9 for no grouping
<code>--combined-fn</code> <code>-cb</code>	Combined .sum file.
GFF output options	
<code>--sample-id</code>	Places this identifier in the header of the gff file.
<code>--include-read-seq</code>	Includes the read sequences of the individual evidences that make the indel candidate in the gff. Default: on.
<code>--noinclude-read-seq</code>	Turns off including read sequences in the gff output.
Filter settings	
<code>--filter-off</code>	Turns off all filtering.
Filtering based on Read Position	
<code>--max-nonreds-4filt</code>	Maximum number of non-redundant reads where filtering is applied. 2 is default setting and 0 is for no filtering.
<code>--min-from-end-pos</code>	Minimum number of base pairs from end of read. Default: 9.1 unless <code>--max-ave-read-pos</code> is set, then there is none.
<code>--max-ave-read-pos</code>	Maximum average read position for filtering. Default: none.
Other Filtering	
<code>--max-coverage-ratio</code>	Maximum normal read / # non-redundant indel

	support ratio to allow. Default: 12.0
--min-insertion-size	Maximum insertion size to include. Default: none.
--min-deletion-size	Maximum deletion size to include. Default: none.
--max-insertion-size	Maximum insertion size to include. Default: none.
--max-deletion-size	Maximum deletion size to include. Default: none.
--require-called-indel-size	Filters candidates where the indel size is not called. Default: on.
--norequire-called-indel-size	Turns this off.

Notes:

- Argument order is **not** important.
- Specifying “none” for an option does not currently work.
- The order of the evidence filenames (separated by commas) can have some effect in situations in which over 1000 beads are used to call an indel. This is uncommon.

3.3 Upstream pairing pipeline considerations (pairing_by_group.pl)

Changing some options for indel finding requires rerunning the pairing pipeline.

--indels_in_good_mates	Find indels in beads where a non-gapped alignment was also found. This is off by default.
--indel_parameters	Default is 'D=11,l=4,i=3,d=3'. Specify this without any spaces, where <ul style="list-style-type: none"> • “D” is the maximum deletion size to find • “l” is one higher than the maximum deletion size. • “i” and “d” are the number of base pairs from the edge required before an indel can be found for insertion and deletion, respectively.
--repeat_limit_indel	Total number of hits in the genome to consider when looking for indels. Default is 10.
--error_indel	The number of total errors allowed in the both tags for indel alignments.
--skip_mates <mates file> --skip_cov_ratio	Used if normal mates file already exists or if coverage ratio calculation is to be skipped.

However, the following can simply be redone with the tool provided here:

--min_num_evid	Default: 2.
--max_num_evid	Default: -1.

Typically for a particular read pair, gap alignments are only found when an ungapped alignment was not found to reduce false positives. This can be turned off by using `--indels_in_good_mates`. Also, different indel parameters, can be adjusted, although, although changing this value from the default may adversely effect the false positive and false negative rates. Furthermore, the placed tag can map to multiple locations, set by `--repeat_limit_indel` in `pairing_by_group.pl`. It is set to a default of 10. (This gets passed as the z parameter in pairing). This setting does not affect the case when multiple pairings from these multiple map positions occur. In this case, it is not considered an indel candidate. Finally, if the mates file already exists, then indel finding can be

significantly faster by using `--skip_mates <mates file>`. Similarly, if the coverage ratio calculation is not required, use `--skip_mates --skip_cov_ratio`.

3.4 Usage Examples

1. Combine several runs together

```
$ small-indel-tool.pl --pairing-dir \  
  /path/to/pairing1,/path/to/pairing2,/path/to/pairing3 \  
  --candidate-fn indel-candidate-newlist \  
  --sample-id "NA18507 library 1"
```

This combines indel results together from three runs and produces a single candidate list. It uses the `indel-evidence-list.pas` and `F3_R3.mates.non-redundant` files from these directories.

The three main result files are `indel-candidate-newlist.gff`, `indel-candidate-newlist.txt`, and `indel-candidate-newlist.sql`. Also, "NA18507 library 1" is reported as the sample id in the GFF file.

Note: The file names are separated by commas but should not contain any spaces.

2. Combine evidences to form consensus

Indels are combined together to form consensus calls.

```
$ small-indel-tool.pl --pairing-dir /path/to/pairing \  
  --candidate-fn indel-candidate-list \  
  --min_num_evid 10 --max_num_evid 200
```

To adjust how many evidences are required to call a candidate, these parameters can be adjusted. The default is 10 for the minimum and no maximum.

3. Making a gff file of the Evidence File

To report the `indel-evidence-list.pas` file in gff format without making a consensus:

```
$ small-indel-tool.pl --pairing-dir /path/to/pairing \  
  --filter-off --consGroup 9 --min_num_evid 1
```

4. Adjust filtering settings

Filters were implemented to decrease the false-positive rate. These examples show you how to adjust or turn off these filters.

```
$ small-indel-tool.pl --pairing-dir /path/to/pairing \  
  --candidate-fn indel-candidate-list --max-coverage-ratio 24.5
```

Maximum clipped coverage / # non-redundant indel support ratio allowed is set to 24.5 (default is 12.0). Use -1 to have no limit.

```
$ small-indel-tool.pl --pairing-dir /path/to/pairing \  
  --candidate-fn indel-candidate-list --max-nonreds-4filt 0 \  
  --norequire-called-indel-size
```

This combines the evidences together, but does no filtering on the results (`--max-nonreds-4filt 0`) on a read-position basis (set by `--max-ave-read-pos`).

Indels where the size cannot be called are also not filtered (`--norequire-called-indel-size`).

```
$ small-indel-tool.pl --pairing-dir /path/to/pairing \
  --candidate-fn indel-candidate-list --filter-off
```

The parameter `filter-off` turns off all filtering.

```
$ small-indel-tool.pl --evidence-fn indel-evidence-list.pas \
  --candidate-fn indel-candidate-list --max-insertion-size 2
  --max-deletion-size 7
```

Filters out insertions of size 3 or larger and deletions of size 8 or larger.

3.5 List of programs/scripts included

<code>small-indel-tool.pl</code>	Main program for calling parts of this package.
<code>process-small-indels</code>	Executable for filtering/gff routines for small indels.
<code>process-small-indels.lisp</code>	Source code for the above.
<code>mpindel_summ.pl</code>	Combines evidence files together to form a combined file (.sum).
<code>exec_lisp.pl</code>	Can call routines in <code>process-small-indels</code> (Used only in Corona Lite)
<code>indel-remove-sizes.pl</code>	Removes indels of a certain size (Used only in Corona Lite)

3.6 Other scripts called by this program

No other scripts are called, but it requires PERL libraries that are installed in Corona Lite v. 4.0r2.0 or later.

3.7 Path constraints

If the path is fully specified, this script will use that path. If no path is specified, then it will use the current directory.

3.8 System Input Files

`indel-evidence-list.pas` from Corona Lite pairing pipeline. Can be from multiple SOLiD™ system runs. Note, future versions may require the pairing pipeline mates file as input as well.

3.9 Input File Versions supported

Corona Lite v. 4.0 r 2.0 or later.

3.10 Output Files

The following output files are created in the directory in which the script is executing:

Primary Files

<code>[indel_candidate_filename].gff</code>	List of indel candidates in GFF v. 3 format indel.
<code>[indel_candidate_filename].sql</code>	A SQL CREATE TABLE command.
<code>[indel_candidate_filename].txt</code>	A list of indel candidates in flat file format importable into a SQL database.

Note: [indel_candidate_filename] is specified by --filename.

Other Files

comb-indel-evid-list.pas	Combined evidence file from multiple runs.
comb-indel-evid-list.pas.sum	The sum file (non-filtered candidates) of the above file.
LOG	Information on how script was called.

3.11 Output Files Description

5. GFF Format

The small Indel file is in standard GFF v3 file format with optional fields in the final column. The format is as follows:

Column 1: “seqid”

The ID of the sequence to which the start and end coordinates refer. In this case, it is the human chromosome number.

Column 2: “source”

Free-text qualifier that indicates the algorithm or method that generated the feature. This should be the name of the software that generates the output file.

Column 3: “type”

Specifies the kind of SOFA feature. This file contains the features insertion site and deletion.

Columns 4 and 5: “start” and “end”

1-based integer coordinates of the feature, relative to the sequence in column 1. For zero-length features, such as insertion sites, “start” equals “end” and the implied site is to the right of the indicated base in the direction of the landmark. For deletions, the start and end indicate the positions in the reference that are not present in the sample.

Column 6: “score”

Floating-point value representing the quality of the evidence for the feature. **Note:** This is currently set to 1.

Column 7: “strand”

“.” Means that “strand” is not relevant for this feature.

Column 8: “phase”

Translation frame; “.” because phase is relevant only for CDS features.

Column 9: “attributes”

ins_len	Insertion length
del_len	Deletion length
tight_chrom_pos	Conservative estimate of chromosome position range of the feature.
loose_chrom_pos	Estimate of maximum chromosome position range of the feature.
no_nonred_reads	Number of reads with unique start positions (non-redundant reads).
coverage_ratio	Clipped normal coverage / number of non-redundant reads. Clipped coverage is where the parts of the read that are within 5 bp at either end are not counted as a part of coverage.
bead_ids	Bead IDs that were the evidence for this indel call
no_mismatches	Number of mismatches for each read.
read_pos	Position in each non-redundant read at which the In/Del occurs.
from_end_pos	Same as above, except that the value is the number of base pairs from

	the end of the read.
strands	Strand for each read.
tags	Tags where the indel was found. Possible values are F3, R3, and FRAG.
indel_sizes	List of sizes of indel found for each evidence.
read_seqs	The read sequences of the evidences.
non_indel_ no_mismatches	Number of mismatches of the other tag that was matched without a gap. Values of NIL occur if that particular bead is from a fragment library.
non_indel_seqs	Sequences of the other tag

Attribute Values that Describe Individual Alignment Pairs

Many of the attributes above are comma separated lists that describe the alignments used to make the indel call. Only the first 1000 alignment pairs for an indel are reported for these attributes. In addition, the order is the same for each attribute. For example,

```
bead_ids=2175_1841_211,684_1070_1152,1885_805_1157;
read_pos=13,12,18; strands=+,+,-
```

Means that >684_1070_1152 matches with read_pos=12, and strand=+.

Tight and Loose Chromosome positions

The range comes from ambiguity in determining the precise indel location and from the combination of evidences. For example, in a polynucleotide tract, AAA-, AA-A, -AAA are all possible indel locations.

There are two ranges listed in this file: **tight** and **loose**. A **tight** range is one in which all the pieces of evidence contain this range. This could be the **null** set, which would be displayed as a blank column. A **loose** range is the set of all possible ranges from all the available evidence.

6. Indel Evidence Format (indel-evidence-list.pas)

Genome Position (Column 1)

The genome position given by this formula:

$$C * 2^{32} + P - 1$$

where C is the chromosome number and P is the position on that chromosome.

Indel Size (Column 2)

A negative value means a deletion, a positive value is an insertion, for example, -8 is a deletion of size 8, and 3 is an insertion of size 3.

Number of Errors (Column 3)

Number of errors in the tag where the indel was found.

Alignment Format (Column 4)

The indel evidence files contain alignment information that is represented in the following format:

```
>78_158_886_25.2_Lib1_1_25.2_CLARA_20071113_1B,1_555384.21.2(18:18_20)[T32312...2]|1_556514.0
```

- >78_158_886 is the bead ID

- 25.2_Lib1_1_25.2_CLARA_20071113_1B is the run information, where 25.2 is the read length and number of mismatches.
- Lib1_1 is the library indicator (This cannot be changed)
- CLARA_20071113_1B is the run name.

The rescued pairing (R3 matching, F3) is separated by |. The indel can be found in either the R3 or F3 tag. In this case, it is found in the R3 tag.

1_555384.21.2 means “a chromosome 1 hit at position 555384”, and 21 is the match length.

```
insertion size = read_length - match_length - 1
```

Negative values are deletions, and positive values are insertions. In this example, the read length was 25, so the indel is an insertion of size 3. The read position found was position 18, zero-based, and 18-20 is the range of other possible positions.

[T32312...2] is the read sequence.

Note: The positions above are all 0-based. This contrasts with the positions reported in the candidate files, which have been adjusted to be 1-based.

3.12 Sample Files

Samples files are found within `indel-sample-files.tar.bz2`. To uncompress the files, enter the following:

```
$ tar -jxvf indel-sample-files.tar.bz2
```

Input Files are located in `input_files/` directory:

```
indel-evidence-list-sample1.pas indel-evidence-list-  
sample2.pas indel-evidence-list-sample3.pas
```

Output Files are located in the `base` directory:

```
comb-indel-evid-list.pas, comb-indel-evid-list.pas.sum
```

`sample.output` Contains output that would normally be printed to the screen (or redirected to a file).

In `output_files/` directory:

```
indel-candidate-list-sample.gff  
indel-candidate-list-sample.sql  
indel-candidate-list-sample.txt
```

Test Command

```
$ small-indel-tool.pl --evidence-fn input_files/indel-evidence-  
list-sample1.pas,input_files/indel-evidence-list-  
sample2.pas,input_files/indel-evidencelist-sample3.pas --  
candidate-fn output_files/indel-candidate-list-mine &> my.output
```

```
$ diff indel-candidate-list-sample.gff indel-candidate-list-  
mine.gff
```

```
$ diff indel-candidate-list-sample.txt indel-candidate-list-  
mine.txt
```

The above diff's should only be different within the gff header.

4 Indel Post-processing Performed in Pairing Pipeline

These commands are here for reference only. They are generated automatically when you use `pairing_by_group.pl`. Refer to `scripts/PAIR_POST_INDEL_1.sh`.

4.1 Removal of highest insertion size

To remove artifacts caused by the cut-off constraint in the pairing algorithm, the highest insertion size is removed as part of the post processing step. The script `indel-remove-sizes.pl` performs this task:

```
for num in `seq 0 9`
do
  indel-remove-sizes.pl 0 3 < intermediates/indel.dat.$num.pas >>
  indel-sorted-list.pas.tmp
done
```

Note: If you used a different number of jobs, change the 9 value above to `#jobs-1`.

4.2 Sorting and adding run identifiers

To sort the results by reference sequence entry position, and then add identifiers for the run, and read length, enter the following:

```
sort -n intermediates/indel-sorted-list.pas.tmp | sed
"s/_R3,/_25.2_Lib1_1_25.2_CLARA_20071113_1B,/g"
> indel-evidence-list.pas
```

Here, 25.2 is the read length and number of mismatches in matching. Lib1_1 is the library indicator, and CLARA_20071113_1B is the run name. The algorithm parses using the underscore character (`_`), so if changes are required, the above format would need to be reproduced.

4.3 Skip recreating the mates file

If you would like to skip recreating the mates file, use the `--skip_mates myOldPairing/F3_R3.mates.non-redundant` option and value. The value is the mates file made from a previous run of pairing on the same tags. This significantly reduces computational time.

Licensing

This software is being licensed to you under the OSI compliant GNU Public License (GPL V3). The license can be found at the following URL: <http://www.gnu.org/licenses/gpl.html>. Please read the license in its entirety and ensure that you understand the licensing conditions for use. Your use of this software indicates your acceptance of this licensing agreement.

© 2009 Life Technologies Corporation. All rights reserved.

The trademarks mentioned herein are the property of Life Technologies Corporation or their respective owners.