APART User's Guide

Last updated: November 15 2011

http://apart.sf.net
Author: Marek Zywicki
Innsbruck Medical University
Contact: Marek.Zywicki@i-med.ac.at

SUPPORTED SHORT READ FORMATS

APART supports all popular short reads formats derived from most popular sequencing platforms, including 454, solexa, illumina (including version1.8), SOLID, ion torrent and others. Reads should be provided as single fasta (or csfasta for SOLID) file without quality values, or fastq file with ASCII-encoded quality values. Currently APART can work with following quality notations:

- phred33 (encoded in ASCII as Phred score + 33) – used by Sanger sequencing, Illumina 1.8+, Ion Torrent, SOLID
- phred64 (encoded in ASCII as Phred score + 64) – used by Illumina 1.3+
- illumina1.5 (encoded in ASCII as Phred score + 64) – used by illumina 1.5+ - similar to phred64, but with special meaning of "B" in ASCII encoding
- solexa (encoded in ASCII as Solexa score + 64) – used by Solexa

If your sequence and quality values are in separate files or different format, we recommend to use one of format conversion utilities, often provided by sequencing platform developers, or use one of the free online tools, like suite of NGS data manipulation tools from GALAXY workbench.

SUPPORTED PLATFORMS

APART pipeline can be used on multiple operating systems, including Linux, MacOS and Windows (via Cygwin software) if prerequisite software has been successfully installed.

INSTALLING APART

The installation of the APART pipeline is done in three steps.

Prerequisites

APART make use of several auxiliary programs, which have to be installed on your system:

- Perl
- Patmatch - a pattern matching tool
- bowtie - a fast and efficient short read aligner
- bedtools - a BED files manipulation suite
- cd-hit-est from cd-hit suite - a sequence clustering software
- NCBI's blast+ - database search program
- a modified version of samtools (distributed together with apart) - software for analysis of genomic alignments in SAM format

APART program files

1. Download current APART program files.
2. Extract the archive.
3. Copy the APART perl scripts into a directory in your `$PATH`.
4. Go to `samtools-[version].apart` directory and build the samtools from source, simply by typing "`make`" in a terminal. Copy `samtools.apart` executable into a directory in your `$PATH`.

APART database

1. Create a directory for APART database, e.g. `/the_path_you_like/apartDB`
2. Download a `common` database section from download section of the APART web site.
3. Extract the `common` database section to the created directory.
4. Download the organism-specific databases you want to use. For every organism you will find three archives in the download section of the APART website, e.g.:
   `organism1.tgz` – a base directory for the organism, containing annotation, fasta-formatted genomic sequence and organism
   `organism1.genome.tgz` – a bowtie genome index for use with sequence-space reads
   `organism1.genome_cs.tgz` – a bowtie genome index for use with color-space reads
   You need to download a base directory and at least one of the genome indexes, depending which is suitable for your data.
5. Extract `organism1.tgz` to `/the_path_you_like/apartDB/` This will create directory `/the_path_you_like/apartDB/organism1/`
6. Extract bowtie genome index(es) to `/the_path_you_like/apartDB/organism1/`

The final directory structure should be as follows:

```
/the_path_you_like/apartDB
/the_path_you_like/apartDB/common
/the_path_you_like/apartDB/organism1
/the_path_you_like/apartDB/organism2 …
```

APART annotation updates

All available updates of the annotation tables will be published on APART web site. If you want to install such an update:

1. Download the file `organism1.annotation_update.yyyy-mm-dd.tgz`
2. Remove directory `/the_path_you_like/apartDB/organism1/annotation`
3. Extract `organism1.annotation_update.yyyy-mm-dd.tgz` to `/the_path_you_like/apartDB/organism1/`

This will create a new annotation directory with updated content.

RUNNING AUTOMATED ANALYSIS WITH `apart.pl`

The most straightforward way of running APART is to use the `apart.pl` command:

```
apart.pl config.apart
```

The only argument which have to be specified is the configuration file containing all necessary information and options for the analysis. The example is provided with the distribution of the APART (config.apart). Copy this file to your working directory and edit to adjust to your project.

The mandatory information which have to be present in the config file is:

START = Starting point of the analysis [`raw/mapping/assembly/custom`]. The `raw` option will perform all the steps of APART analysis, in following order: cleaning, genome mapping, assembly, repeat annotation, genomic and sequence annotation and results generation. The `mapping` option will start directly from genome mapping of the reads, so the reads should be already quality-filtered and should not contain any adaptors. The `assembly` option will start the analysis from assembly of the contigs, thus it require the genomic alignment of the reads in SAM format as input (`INPUT_FORMAT` option). The `custom` option can be used for re-running parts of the analysis with modified parameters.In this mode individual task can be enabled, however care must be taken that all necessary input files for selected modules exist in the working directory and file naming scheme is consistent (files should have common file name prefix, as stated by `BASENAME` option).

INPUT_FILE = Sequence reads file (if starting point is raw or mapping) or SAM genomic alignment file (if starting point is assembly)

INPUT_FORMAT = Input file format available choices are: `fasta`, `solexa` (for fastq files with solexa score + 64 quality values), `phred33` (for fastq files with phred score + 33 quality encoding), `illumina1.5` (for illumina 1.5+ files), `phred64` (for fastq files with phred score +

64 quality encoding), `csfasta` (for SOLID color-space fasta files), `phred33cs` (for SOLID color-space fastq files with phred score + 33 quality encoding) and sam (for genome alignment file in SAM format, it the starting point of analysis is `assembly`). All other formats can be converted to those accepted by APART with freely available tools, like GALAXY workbench.

BASENAME = File name prefix for all the files created by APART pipeline. In `custom` mode it is also the file prefix used for incorporation of necessary files for the selected modules.

ORGANISM = Name of the organism. It have to be consistent with organism directory in database (e.g. from this guide installation example: organism1).

DB = Location of the APART database. It should point to the database directory, e.g. `/the_path_you_like/apartDB` from the installation example

Optional information for accurate control of the analysis:

URL = URL address where results, including seq directory will be published. If provided, it will enable the direct linking from BED file uploaded to UCSC genome browser to detail page of the contig. If `static` is entered, the linking is disabled (default).

BEDURL = URL address where BED file will be published. If provided, it will enable automatic loading of the BED file to UCSC genome browser when UCSC link in results html pages is used. If `static` is entered, the linking is disabled (default).

MIN_READS = Minimum number of reads per contig. This option filters out contigs with low coverage (default = 10).

PROC = Number of processors to use (default = 1)

CLEAN_READS = Enables cleaning of adaptor sequences and quality filtering of the reads in `custom` mode of operation. Possible choices: 0 or 1

5_PRIME_ADAPTOR = Sequence of the 5' adaptor. If provided, APART will locate and remove this sequence from reads (default = none)

3_PRIME_ADAPTOR = Sequence of the 3' adaptor. If provided, APART will locate and remove this sequence from reads (default = none)

ADAPTOR_CHANGES = Number of allowed changes in adaptor sequence. This value correspond to number of possible mismatches, insertions or deletions in adaptor sequence during cleaning. Entered value will be used for 5' adaptor scan, and for search of 3' adaptor it will be increased by 1 (due to usual lower quality of 3' ends) (default = 2)

MIN_LENGTH = Minimum length of the reads after cleaning steps (default = 18)

3_PRIME_TRACK = Nucleotides expected in 3' poly-nucleotide track. This option allows for removal of e.g. poly-A or poly-C tracks located between insert and 3' adaptor, which are used in some library construction approaches (default = N)

QUALITY = Average quality threshold for filtering of low quality reads (default = 30)

MAP_READS = Enables mapping of the reads to reference genome in `custom` mode of operation. Possible choices: 0 or 1

SELECT_CLEANED = Selection of cleaned reads for mapping. This option allows for choosing which subset of previously cleaned reads should be used for mapping. Choice should be entered as space separated list of following: `trimmedboth, trimmed3, trimmed5, untrimmed` (default: `trimmedboth trimmed3 trimmed5 untrimmed`)

MAX_MISMATCH = Maximal number of mismatches allowed for mapping with `bowtie`. Possible value range is 0-3 (default = 1)

BOWTIE_PROC = Number of processors to use with bowtie (default = same as **PROCESSORS** parameters)

MAX_HITS = Maximal allowed number of mappings per read (default = 100)

ASSEMBLE_READS = Enables assembly of the reads into contigs in `custom` mode of operation. Possible choices: 0 or 1

CLUSTERING = Method for clustering of the contigs. The `names` option will enable the clustering based on comparison of read names between the contigs. This will result in joining of contigs with the same read composition. The `sequence` option allows for clustering of the contigs with identical or similar consensus sequence. The `none` option disables clustering (default = `names`)

CLUST_IDENTITY = Sequence identity threshold for sequence-based clustering (default = 98)

ANNOTATE_CONTIGS = Enables annotation of the contigs in `custom` mode of operation. Possible choices: 0 or 1.

BLAST_INTERGENIC = Enables sequence-based annotation of the contigs which are intergenic, intronic or antisense after genome-based annotation. The consensus sequences are blasted against functional RNA database (fRNAdb). Possible choices: 0 or 1 (deafult = 1)

BLAST_IDENTITY = Identity threshold for blastn during the sequence-based annotation (default = 80)

ANNOTATE_REPEATS = Enables genomic repeat annotation in `custom` mode of operation. Possible choices: 0 or 1

GENERATE_HTML = Enables HTML results generation in `custom` mode of operation. Possible choices: 0 or 1

GENERATE_STATISTICS = Enables generation of joined statistics text file in `custom` mode of operation. Possible choices: 0 or 1

RUNNING INDIVIDUAL APART MODULES

The APART pipeline consist of six perl scripts which perform a subsequent steps of the analysis. In addition to automated script `apart.pl`, they can also be called directly.

cleaning.apart.pl

This script performs initial preprocessing and cleaning of the sequence reads. It is able to remove the 5' and 3' adaptor sequences and polynucleotide tracks from 3'ends. It include also quality filtering and trimming. After theses steps, reads longer than chosen minimum are returned. The color-space encoded reads can not be submitted for cleaning. If there is such need, they have to be converted to sequence space prior to APART analysis (we recommend the use of GALAXY workbench).

Input:

Sequence read file in fasta or fastq format. Quality values in fastq file should be ASCII-encoded.

Output:

A series of files containing trimmed and untrimmed sequence reads in same format as input, with extensions:

- *.trimmedboth – reads trimmed on 5' and 3' ends
- *.trimmed3 – reads trimmed on 3' end only
- *.trimmed5 – reads trimmed on 5' end only
- *.untrimmed – reads not trimmed
- *.rejected – reds rejected due to quality or length limits

*.clean.statistics – a text file containing different read statistics, like number of cleaned reads etc.

Usage:

```
cleaning.apart.pl [options] -i input_file -I input_format
```

Mandatory options:

-i input file name

-I input file format – available choices are: `fasta`, `solexa` (for fastq files with solexa score + 64 quality values), `phred33` (for fastq files with phred score + 33 quality encoding), `illumina1.5` (for illumina 1.5+ files) and `phred64` (for fastq files with phred score + 64 quality encoding)

Other options:

-a sequence of the 5' adaptor. If provided, the 5' adaptor trimming will be performed

-b sequence of the 3' adaptor. The sequence should be provided in reverse-complement orientation to oligonucleotide that has been used for library construction - as expected to appear in the sequence reads. If provided, the 3' adaptor trimming will be performed

-c number of allowed changes in adaptor sequences (default = 2). This number correspond to number of mismatches, insertions or deletions which will be allowed for detection of 5' adaptor. The number of changes in 3' adaptor is always increased by 1 (due to lower quality of 3' ends of the reads)

-o output file name prefix (default = same as input file)

-O output format (default = same as input). It is possible to convert the fastq files to fasta files only

-l minimal length of cleaned sequence (default = 18). This value should be adjusted to the reference genome size and reads content. For small genomes it is reasonable to use 16, for complex eukaryote genomes the minimum should be 18

-t letters expected in polinucleotide tail (default = N). Usually single letter should be provided, for example “A” for poly-A tail removal

-n remove sequences containing Ns (default = 0)

-T perform quality trimming of 3' ends for reads without 3' adaptor (default = 1)

-Q average read quality threshold for sequences (default = 30). This value should be adjusted to match the expected sequencing technology quality.

mapping.apart.pl

This script uses the `bowtie` aligner to map the sequence reads to reference genome.

Input:
Sequence read file in fasta or fastq format. Quality values in fastq file should be ASCII-encoded.

Output:
Alignment file in SAM format with removed unaligned reads

Usage:

```
mapping.apart.pl -i input_file -I input_format -d db_location
```

Mandatory options:

-i input file name

-I input format – available choices are: are: `fasta`, `solexa` (for fastq files with solexa score + 64 quality values), `phred33` (for fastq files with phred score + 33 quality encoding), `illumina1.5` (for illumina 1.5+ files), `phred64` (for fastq files with phred score + 64 quality encoding), `csfasta` (for SOLID color-space fasta files) and `phred33cs` (for SOLID color-space fastq files with phred score + 33 quality encoding)

-d Location of the organism-specific database – should point to the organism directory, e.g. `/the_path_you_like/apartDB/organism1` from the installation example

Other options:

-m number of allowed mismatches in alignment in range 0-3 (default = 1)

-M Maximum number of mappings per read (default = 100)

-o Prefix for output files (default: as in annotation file)

assembly.apart.pl

This script performs the assembly of the reads into contigs which is based on their overlapping positions on reference genome. Then, for every contig it performs a number of tasks:

– calls a read-based consensus sequence and quality values for consensus letters
– calculates read number, normalized read number and maximum coverage value
– detects putative processing products and calculates their abundance

Next, it performs the clustering of the non-unique contigs based either on read composition or consensus sequence.

Input:

– reads alignment file in SAM format

– optionally, file containing reads used for mapping, if sequence files containing unmapped and over-mapped reads should be written

Output:

*.multimapped – a file containing sequence reads that has been mapped to reference genome more times than provided limit (file is created if reads used for mapping are provided)

*.unmapped – a file containing reads that have not been mapped to the reference genome (file is created if reads used for mapping are provided)

*.assembly.statistics – a text file containing different statistics about reads, contigs and clustering

*.plus.wig; *.minus.wig – a WIG files containing coverage plots of plus and minus strand, respectively, for display in genome browsers

*.bed – a BED file containig the positions of the all contigs for display in genome browsers

*clust.bed – a BED file containig the positions of the representative contigs only for display in genome browsers. This file is created only if contig clustering is enabled.

*.cons – a file containing a consensus sequences of the contigs in fasta format

*.align – a file containing the corresponding genomic sequences of contigs and consensus sequence quality values. Letters identical to consensus are annotated as dot, differences as letters. Consensus quality correspond to a frequency of the displayed consensus letter rounded to a full integer (values range 2-9).

*.expresssion – a tab-separated file containing the normalized read number (RPM), relative positions of predicted processing products (in format start..stop_abundance), genomic uniqueness and read number for every contig.

*.clusters – a file containing the clustering results. The representative contig is followed by space-separated list of clustered contigs.

"seq" directory – for every contig two files are written with names:

contigID – fasta file containing the contig's reads

contigID.sam – SAM read alignment of the reads within the contig.

Usage:

```
assembly.apart.pl [options] -i input_file -d db_location
```

Mandatory options:

-i input file name in SAM format

-d database localization – should point to the organism directory, e.g. `/the_path_you_like/apartDB/organism1` from the installation example

Other options:

-o output file name prefix (default: as input file)

-a file containing all reads used for mapping. If provided, the output files containing unmapped and over-mapped sequences will be written

-r minimum read number per contig (default = 10). This option allows to filter a low abundant contigs speeding up the subsequent steps of the analysis

-m maximum number of mappings per read (default = 100)

-c clustering method (default = names). The possible choices are `names` for read composition based clustering of the non-unique contigs or `sequence` for sequence-based clustering.

-h clustering threshold for sequence based clustering method (default = 0.98)

-u URL address for linking from genome browser. This option allows to introduce the URL to a web site where APART results will be published. If provided, the link will be added to the header of the BED file allowing for direct access to the detailed contig information from UCSC genome browser (default = `static`)

repeats.apart.pl

This script checks for contig overlap with the genome-annotated repeats.

Input:

– BED file containing contigs assembled by `assembly.apart.pl`

Output:

*.genrep – a file containing repeat annotation. Only repeat annotated contigs are listed.

Usage:

```
repeats.apart.pl [options] -b BED_input_file -d db_location
```

Mandatory options:

-b input file name in BED format

-d database localization – should point to the organism directory, e.g. `/the_path_you_like/apartDB/organism1` from the installation example

Other options:

-v Minimal contig overlap to annotate repeat (default = 0.1)

-o output file name prefix (default: as input BED file)

annotate.apart.pl

This script perform the annotation of the contigs based on reference genome annotation. Next, optionally it extracts the sequences of intergenic, intronic and antisense contigs and comapres them against fRNAdb sequence collection. It combines also information gathered in previous APART steps into final annotation table.

Input:

– BED file containing contigs assembled by `assembly.apart.pl`

– *.expression file produced by `assembly.apart.pl`

– *.cons – consensus sequence file produced by `assembly.apart.pl` (optionally, if sequence-based annotation is enabled)

– *.genrep – file containing repeat annotation (optionally, if repeat annotation was performed with `repeats.apart.pl`)

Output:

*.annotation – a file containing contig annotation table in tab-separated format. For single contig multiple entries are possible if contig overlap with more than one annotation feature.

Usage:

```
annotate.apart.pl [options] -b BED_input_file -e expression_file -d
db_location
```

Mandatory options:

-b input file name in BED format

-e *.expression file produced by `assembly.apart.pl`

-d database localization – should point to the organism directory, e.g. `/the_path_you_like/apartDB/organism1` from the installation example

Other options:

-c contigs consensus sequence file. If provided, sequence-based annotation will be performed

-p number of processor for use with blastn during sequence-based annotation

-s blast similarity threshold for sequence-based annotation (default = 80)

-r repeat annotation file

-o output file name prefix (default: as input BED file)

results.apart.pl

This scripts combine all the gathered information into a series of HTML tables and pages.

Input:

– *.annotation file containing contigs annotation produced by `annotate.apart.pl`

– *.align – genomic alignment file produced by `assembly.apart.pl`

– *.cons – consensus sequence file produced by `assembly.apart.pl` (optionally, if sequence-based annotation is enabled)

– *.clusters – file containing contig clustering information produced by `assembly.apart.pl`

Output:

*.html – a series of HTML files containing a tables with contigs annotation. Only representative contigs are presented. Contigs clustered to representative, can be seen in contig detail page

seq/*.html – a HTML files containing detailed description of every representative contig from the result tables.

*.annotataion.statistics – a file containing detailed statistics about annotated features gathered for representative contigs

Usage:

```
results.apart.pl [options] -a annotation_file -g genomic_alignment_file -i cluster_file -c consensus_file -d db_location
```

Mandatory options:

-a *.annotation file containing contigs annotation produced by `annotate.apart.pl`

-g *.align file containing genomic alignments for the contigs produced by `assembly.apart.pl`

-i clustering information file produced by `assembly.apart.pl`

-c contigs consensus sequence file produced by `assembly.apart.pl`

-d database localization – should point to the organism directory, e.g. `/the_path_you_like/apartDB/organism1` from the installation example

Other options:

-u URL address of the on-line published BED file for automated display in UCSC genome browser (default = `static`)

-o output file name prefix (default: as input BED file)