

iAssembler (current version: v1.2.1 -12/02/10)

Introduction

iAssembler is a standalone package to assemble ESTs generated using Sanger and/or 454 pyrosequencing technologies into unigenes. The pipeline gives much higher accuracy in EST assembly by employing an iterative assembly strategy and automatic error corrections of mis-assemblies. iAssembler first performs iterative assemblies using [MIRA](#) and [CAP3](#) (default: four cycles of MIRA assemblies followed by one CAP3 assembly) to correct assembly errors (mostly two similar sequences are not assembled) which occur frequently in just one round of assembly. The program then performs post-assembly quality checking by 1) aligning each cDNA sequence to its corresponding unigene sequence to identify mis-assemblies; and 2) comparing unigene sequences against themselves to identify sequences from same genes that were not assembled together. The identified mis-assemblies are then corrected by the program automatically.

Check the [short presentation on iAssembler](#) for more information.

System requirement and dependencies

- Linux (required)
- Perl version 5.10.0 or higher (required). [Perl](#) was installed by default on most Linux systems
- BioPerl version 1.006 or higher (required). Please check <http://www.bioperl.org> and [wiki/Installing_BioPerl](#) for more details on installation of BioPerl.
- [NCBI BLAST package](#) (required). Provided in iAssembler.
- [MIRA assembly program](#) (required). Provided in iAssembler.
- [CAP3 assembly program](#) (required). Provided in iAssembler.

Release notes

- iAssembler v1.2.2 - 03/28/11. Changes from previous version:
 1. Replaced MIRA with a newer version (V2.9.43 -> V3.2.0).
 2. Fixed several other small bugs
- iAssembler v1.2.1 - 12/02/10. Changes from previous version:
 1. Fixed a small bug - [-e can't be less than 6]
- iAssembler v1.2 - 08/02/10. Changes from previous version:
 1. Compatible with MIRA version 3.x
- iAssembler v1.1 - 06/23/10. Changes from previous version:
 1. Fixed the error that caused EST clustering to fail for datasets containing highly redundant sequences
 2. Fixed several other small bugs
- iAssembler v1.0 - 05/21/10. Changes from previous version:

1. Added an output file in [SAM format](#). The file contains the alignment information of each sequence read to its corresponding unigene and can be views by several visualization programs such as [Tablet](#) and [IGV](#).
 2. Combined percent identity cutoff for clustering (-x) and assembly (-p) into a single parameter (-p). Parameter -x is disabled
 3. Disabled clustering using blastn. Currently only megablast is used for clustering. Parameter -b now has different meaning (see below)
 4. Added -b parameter which specifies the number of threads used for MIRA assembly program
 5. Added -d parameter to control whether to generate program log files
- iAssembler v1.0 (beta) - 04/13/10

Installation

Installation of iAssembler is straightforward. Just [download](#) the appropriate version of iAssembler for your system and uncompress the downloaded file.

```
shell$ tar -xzf iAssembler-1.0.x32.tar.gz
```

This will generate a folder named "iAssembler-1.0.x32" on a 32-bit machine or "iAssembler-1.0.x64" on a 64-bit machine (we call this folder "iAssembler home folder"). iAssembler home folder includes two subfolders, a "bin" folder which contains all executables and a "doc" folder which contains the program documentation and the example configure file (see below). The home folder also contains a perl script, iAssembler.pl, which is the core script to run the whole iAssembler pipeline.

Run iAssembler

Quick Start

1. Put the EST sequence file in FASTA format (assuming the file name is **input_EST_seq**) into iAssembler home folder
2. Go to iAssembler home folder and run iAssembler with the following command

```
shell$ perl iAssembler.pl -i input_EST_seq
```

3. The program will generate an output folder named **input_EST_seq_output** which contains all the output files. See [below](#) for the description of the output files.

Input files

iAssembler takes a sequence file in FASTA format, and optionally the corresponding sequence quality file, as its input. The sequences must be processed and cleaned by removing low quality regions and sequences derived from adapters, vectors, rRNAs, tRNAs, as well as

sequences from other organelles such as chloroplast and mitochondrion. iAssembler itself does not provide functions to clean and trim raw sequences. Two such programs are [lucy](#) and [seqclean](#).

Parameters

(Note: Based on our experiences, the default settings of iAssembler program can achieve very high quality assemblies for most Sanger and/or 454 ESTs.)

Section 1: Input parameters

- i **[String]** Name of the input sequence file in FASTA format (*required*)
- q **[String]** Name of the quality file in FASTA format (default: none)
- z **[String]** Name of the parameter configuration file (default: none)
iAssembler allows users to put their preferred assembly parameters in a file so users do not need to type the same parameters every time they run the program. (see below for detail description of [iAssembler assembly parameters](#)). Here is how to create the parameter configuration file.

1. Copy the example parameter configuration file (config.manually) from the "doc" folder to iAssembler home folder and change the parameters in the file accordingly.
2. The command running iAssembler with the parameter configuration file will be:

```
shell$ perl iAssembler.pl -i input_EST_seq -z config.manually
```

Section 2: Assembly parameters

- a **[Integer]** number of CPUs used for megablast clustering (default = 1)
- b **[String]** number of CPUs used for MIRA assembly program (default = 1)
- e **[Integer]** maximum length of end clips (0~100; default = 30)
- h **[Integer]** minimum overlap length (≥ 30 ; default = 30)
- p **[Integer]** minimum percent identify for sequence clustering and assembly (95~100; default = 97)

Section 3 : Output parameters

- u **[String]** prefix used for IDs of the assembled unigenes (default = UN)

iAssembler names the resulted unigenes with a prefix and trailing numbers, e.g., UN00001

- l **[Integer]** length of the trailing numbers in unigene IDs (>= default; default = number characters of the maximum number assigned to unigenes)
For example, if the maximum trailing number assigned to the resulted unigenes is 5000, then the default of -l is 4 ('5000' has 4 characters). In this case users can set a number greater than or equal to 4.
- s **[Integer]** start number of unigene ID trailing number (>= 1; default = 1)
- o **[String]** Name of the output directory (default = "input file name" + "_output")
- d Produce log files. With this parameter will produce log files in the output folder

Output files

iAssembler generates four files and a "log" folder (if -d is supplied) in the output directory.

1. unigene_seq.fasta

Unigene sequences (FASTA format) generated from the EST assembly process.

2. unigene.sam

A [SAM format](#) file containing the alignment information of each sequence read to its corresponding unigene. The file can be viewed by [Tablet](#), [IGV](#), and many others...

3. contig_member

A tab-delimited txt file containing unigenes and their corresponding EST members.

4. unigene_mp

A tab-delimited txt file containing the mapping details of EST members to their corresponding unigenes

EST ID	EST Length	Unigene ID	Unigene length	Query Start	Query End	Hit Start	Hit End	Strand	% Identity
EST0001	116	UN0001	1195	11	108	650	747	1	100.00

5. member_position_stat

A tab-delimited file containing the summary statistics of aligning ESTs to their corresponding unigenes.

Len/%ID	100-99	99-98	98-97	97-96	96-95	95-94	94-93	93-92	92-91	91-90	<90
000-100	0	0	0	0	0	0	0	0	0	0	0
100-200	1046	136	64	0	0	0	0	0	0	0	0
200-300	2044	343	112	0	0	0	0	0	0	0	0
300-400	3510	573	139	0	0	0	0	0	0	0	0
400-500	967	131	36	0	0	0	0	0	0	0	0
>500	9	1	0	0	0	0	0	0	0	0	0

6. log folder (if parameter -d is supplied)

A folder containing all the log files from the program

Download

Current version of iAssembler is v1.2.2. It's available for both 32 and 64-bit linux systems.

[Download iAssembler from the ftp server](#)

Note: For large dataset, 32-bit CAP3 can run into the "out of memory" problem. In this case please use the 64-bit version of iAssembler.

Contact

For questions and suggestions, please contact us at bioinfo@cornell.edu