TAMPA User's Guide

Introduction

The Tool for Analyzing Mate Pairs in Assemblies (TAMPA) (Dew, et al. 2005) analyzes mate pairs (paired end reads of clones) in genomic assemblies to detect assembly problems. It was initially developed to process mate pairs mapped to several assemblies of the human genome but has been adapted to more general use. Specifically, TAMPA can now be run on the following.

1. assemblies produced by the Celera Assembler (Myers, et al. 2000)
2. mate pairs mapped to sequence via sim4db (Florea, et al. 1998, 2005)
3. mate pairs mapped to or assembled into sequences by other means

Compiling

TAMPA is distributed under the GNU Public License and is available on Sourceforge as part of the wgs-assembler (Celera Assembler) project at http://wgs-assembler.sourceforge.net. The software exists in the wgs-assembler/src/AS_MPA directory and builds successfully on most UNIX platforms using make. Binaries and scripts will be placed in the wgs-assembler/*platform*/bin directory, where *platform* is whatever uname returns on your system. Some of the programs require libraries (and some scripts require programs) built in other directories under src. Please contact the author at wednai@yahoo.com with any problems or questions.

Files

Input Files

There are three types of files used as input to TAMPA: one intra- and one inter-sequence mate pair file per sequence and one libraries file. All are tab-delimited ASCII text files.

The libraries file lists the mean and standard deviation of each clone library used to generate mate pairs. Each line consists of the following fields (UID refers to a unique identifying number).

Field	Value	Type
1	UID of the library	64-bit unsigned int
2	Mean length	float
3	Standard deviation	float

The intra-sequence file lists one mate pair per line for mate pairs within a single sequence (be that a scaffold, super-contig, chromosome, or whatever). Intra-sequence files must have filenames of the form *assemblyName*_([0-9]*)_intra.txt, where *assembyName* is of the user’s choosing, and ([0-9]*) is a unique number identifying the sequence. Data for each sequence is in a separate file, and each record consists of the following fields.

Field	Value	Type

1	Orientation	char:
		I = innie,
		O = outtie,
		N = normal,
		A = outtie

2	UID of left fragment	64-bit unsigned int
3	UID of right fragment	64-bit unsigned int
4	UID of library	64-bit unsigned int
5	Coordinate of 5’ end of left fragment in sequence	64-bit signed int
6	Coordinate of 5’ end of right fragment in sequence	64-bit signed int

The inter-sequence file lists one mate pair per line for mate pairs in which one read is in one sequence and the other read is in a different sequence. Inter-sequence files must have filenames of the form *assemblyName*_([0-9]*)_inter.txt, where *assembyName* is of the user’s choosing, and ([0-9]*) is a unique number identifying the sequence. Data for each sequence is in a separate file, and each record consists of the following fields.

Field	Value	Type
1	UID of fragment on this sequence	64-bit unsigned int
2	UID of this sequence	64-bit unsigned int
3	Coordinate of 5’ end of fragment on this sequence	64-but signed int

4	Orientation of fragment on this sequence	string:
		A_B = 5’ is left of 3’
		B_A = 3’ is left of 5’

5	UID of fragment on other sequence	64-bit unsigned int
6	UID of other sequence	64-bit unsigned int
7	Coordinate of 5’ end of fragment on other sequence	64-bit signed int

8	Orientation of fragment on other sequence	string:
		A_B = 5’ is left of 3’
		B_A = 3’ is left of 5’

9	UID of library	64-bit unsigned int

Each inter-chromosome mate pair may be listed in two files. However, for each inter-sequence file, TAMPA will only process inter-sequence mate pairs between that sequence and lower-numbered sequences.

Output Files

TAMPA can re-estimate clone library means and standard deviations, resulting in a libraries file of identical structure to the input libraries file with the suffix ".Reest". This is usually a good thing, but it will under-estimate library lengths if they are long relative to the sequence lengths.

When run on a set of input files, TAMPA generates four aggregate output files in addition to the several sequence-specific output files. The aggregate files are as follows.

1. intra-sequence summary file with the name *assemblyName*.intra.summary.tampa. This is a tab-delimited, spreadsheet-importable table with a self-explanatory format.
2. inter-sequence summary file (*assemblyName*.inter.summary.tampa), which is also tab-delimited, spreadsheet-importable, and self explanatory.
3. breakpoints file (*assemblyName*.intra.breakpoints.tampa), which is a concatenation (with a single header line, describing fields) of the intra-sequence breakpoints files generated for each sequence. The format is described below.
4. inter-sequence intervals file (*assemblyName*.inter.breakpoints.tampa), which is a concatenation (with a single header line, describing fields) of the inter-sequence interval files generated for each sequence. The format is described below.

For each intra-sequence input file TAMPA can generate two or more output files. These are as follows.

1. breakpoints file. The filename has the format *assemblyName*.([0-9]*).intra.breakpoints.txt. The file has a header line listing the fields. For all but deletions, the fields are interpreted as follows.

Field	Value	Type
1	UID of this sequence	64-bit unsigned int
2	Left coordinate of left breakpoint interval (i.e., interval within which the left breakpoint occurs)	64-bit signed int
3	Length of left breakpoint interval	64-bit signed int
4	Left coordinate of right breakpoint interval	64-bit signed int
5	Length of right breakpoint interval	64-bit signed int

6	Type of problem	string:
		ins = insertion,
		del = deletion,
		inv = inversion,
		trn = transposition

7	Polymorphic	char: Y = yes

		N = no
8	Number of contributing mate pairs	integer

For deletions, there is only one breakpoint interval, so fields 5 and 6 list the minimum and maximum possible deletion lengths, respectively.

2. summary file. The filename has the format *assemblyName*.([0-9]*).intra.summary.txt. This lists simple counts of raw, confirmed, polymorphic, and other types of mate pairs and problems.

3. optional ATA file for use with the Assembly-to-Assembly comparison software (Istrail, et al. 2004). The filename has the format *assembyName*.([0-9]*).*status*.*type*.ata, where *status* is either confirmed or polymorphic and *type* refers to the type of assembly problem. This file lists one summary line per problem in addition to a listing of all mate pairs contributing to each problem. The format is specific to the ATA software suite.

4. optional gnuplot file for displaying mate pairs and intersections in gnuplot. The filename has the format *assembyName*.([0-9]*).*status*.*type*.gp. This lists coordinates for plotting each problem interval and the mate pairs contributing to it. In gnuplot use the command “plot ‘filename’ with lines” to display.

For each inter-sequence input file TAMPA can generate two or more output files. These are as follows.

1. breakpoints file. The filename has the format *assemblyName*.([0-9]*).inter.breakpoints.txt. Given the current state of TAMPA implementation, this file lists only the intervals on each sequence that should be on the same sequence without additional information (e.g., which of the two sequences they should be on). The file has a header line listing the fields.

Field	Value	Type
1	UID of this sequence	64-bit unsigned int
2	Left coordinate of interval on this sequence	64-bit signed int
3	Length of interval on this sequence	64-bit signed int
4	UID of other sequence	64-bit unsigned int
5	Left coordinate of interval on other sequence	64-bit signed int
6	Length of interval on other sequence	64-bit signed int

7	Orientation of the pair of intervals	char:
		I = innie,
		O = outtie,
		N = normal,
		A = outtie

8	Number of contributing mate pairs	integer

2. summary file. The filename has the format *assemblyName*.([0-9]*).inter.summary.txt. This lists simple counts inter-sequence mate pairs and intervals.

3. optional ATA file for use with the Assembly-to-Assembly comparison software (Istrail, et al. 2004). The filename has the format *assembyName*.([0-9]*).inter.ata. This file lists one summary line per inter-sequence interval pair in addition to a listing of all contributing mate pairs. The format is specific to the ATA software suite.

4. optional gnuplot file for displaying mate pairs and intersections in gnuplot. The filename has the format *assembyName*.([0-9]*).inter.gp. This lists coordinates for plotting each inter-sequence interval pair and the mate pairs contributing to it. In gnuplot use the command “plot ‘filename’ with lines” to display.

Running TAMPA

Running on Celera Assembler assemblies

Run the script asm2TampaResults.pl. This creates a subdirectory named TAMPA, populates it with all required input files based on an assembly file, gatekeeper store, and fragment store, runs TAMPA, and copies the four aggregate files (described above) into the current directory.

Running on sim4db data

This method requires a sim4db output file, and a frag file compatible with the Celera Assembler (which contains library and pairing information) as input. To run on sim4db data, perform the following steps.

1. Create a directory in which to run TAMPA.
2. Run sim4db2tampa.pl. It will generate appropriately named intra- and inter-sequence TAMPA input files and a library file named *assemblyName*Libs.txt. Usage is as follows.

```
Usage: sim4db2tampa.pl  -f fragFilename  -s sim4dbFilename  -o outputPrefix  [-m
multiMatches]  [-h]
  -f fragFilename    name of .frg-type file
                       which contains DST and LKG messages
  -s sim4dFilename   name of sim4db output file
                       which lists fragment mappings
  -o outputPrefix    prefix of filenames for output
  -m multiMatches    what to do with fragments with
                       multiple mappings
                       omit this flag to list only uniquely
                         mapped fragments
                       c = list match with highest coverage
                       i = list match with highest identity
                       default is to omit multiply mapped
                         fragments.
```

3. Run runTampa. It has several output options. By default it will re-estimate clone library means and standard deviations. It will generate the four aggregate output files, the individual breakpoints and summary files, and optionally ATA and gnuplot files. Usage is as follows.

```
Run TAMPA (Tool for Analyzing Mate Pairs in Assemblies) on a genomic assembly.

    runTampa  [options]  <-a assembly>  <-l library>

    <-a assembly>  The prefix of the input filenames. Intra-sequence files
                   must have the form
                            assembly_([0-9]*)_intra.txt
                   Inter-sequence files must have the form
                            assembly_([0-9]*)_inter.txt
                   Refer to the user's manual for file formats.

    <-l library>  The file listing clone libraries, means, and standard
                  deviations.
                  Please refer to the user's manual for file formats.

    options:
      -h               Print help.

      -c               Print TAMPA citation.

      -v <level>       Set verbosity to level.

      -b <path>        Path to binaries.
                       Default = the current path.

      -r               Do NOT reestimate clone library means stddevs.
                       Conflicts with -i and -e.

      -i <iterations>  Iterate iterations times when reestimating clone
                       library means and stddevs.
                       Conflicts with -r.
                       DEFAULT = 4.

      -e <num_sigmas>  Exclude mate pairs beyond num_sigmas of the working
                       mean when reestimating library means and stddevs.
                       Conflicts with -r.
                       DEFAULT = 4.

      -s <num_sigmas>  To be satisfied, a mate pair must be within num_sigmas
                       of the library mean.
                       DEFAULT = 3.

      -p <pairs>       The minimum number of mate pairs that must agree
                       to confirm an assembly problem.
                       DEFAULT = 2.

      -o               Do not run TAMPA on intra-sequence mate pairs.

      -x               Do not run TAMPA on inter-sequence mate pairs.

      -g               Generate output for gnuplot.

      -t               Generate ATAC output.

      -m               Generate raw output.

 Version 1.01 (Build 1.1.2.5)
```

Running on other mapped or assembled sequence data

In this case, you will need to generate the TAMPA input files described above – one library file, and one intra- and one inter-sequence mate pair file per sequence. You can omit any number of mate pair files since the runTampa script merely processes whatever files exist that match the input filename specifications.

Once the required input files exist, you can run the runTampa script as described above.

References

I. Dew, B. Walenz, G. Sutton, A Tool for Analyzing Mate Pairs in Assemblies (TAMPA), Journal of Computational Biology 2005; 12 (5):497-513.

L. Florea, G. Hartzell, Z. Zhang, G. M. Rubin, and W. Miller. A Computer Program for Aligning a cDNA Sequence with a Genomic DNA Sequence. Genome Research 1998; 8(9):967-974.

L. Florea, V. Di Francesco, J. Miller, R. Turner, A. Yao, M. Harris, B. Walenz, C. Mobarry, G. V. Merkulov, R. Charlab, I. Dew, Z. Deng, S. Istrail, P. Li, and G. Sutton.
Gene and alternative splicing annotation with AIR Genome Research 2005; 15(1):54-66.

Eugene W. Myers, et al., A whole-genome assembly of Drosophila, Science. 2000 March 24; 287(5461):2196-2204.