AnnoGraph Usage Notes

Aaron Halpern

Saul A. Kravitz

Karin Remington

Qing Zhang

1. Overview

AnnoGraph is a tool for viewing and annotating the graph of relationships between BACs that have been processed by the overlay assembly tool. The relationships reflect shared fragment hits between BACs.

2. AnnoGraph Release Area

An AnnoGraph release area is a directory consisting of the following files and directories:

AnnoGraph_Release_Notes	-- a text file with README type info
bin/	A directory with executables and scripts
data*/	Data directories of data compatible with this version of AnnoGraph

The latest AnnoGraph can be found /prod/overlay/meta_assembly/AnnoGraph_3_0.

3. Setting up AnnoGraph for Use

3.1. For Users

Set the environment variable ANNOGRAPH_ENV to point to the release area.

Add $ANNOGRAPH_env/bin/ to your path.

Create a data directory, and link all files from $ANNOGRAPH_env/data* to your directory. This will allow you to use the latest data files, but generate your output locally.

3.2. For Developers of Node/Edge commands

Set the environment variable ANNOGRAPH_ENV to point to a local directory.

Add $ANNOGRAPH_env/bin/ to your path.

Copy the bin/ subdirectory of the AnnoGraph release area to a local directory and add it to your path. Edit the node and edge commands files appropriately.

Create a data directory, and link all files $ANNOGRAPH_env/data* to your directory. This will allow you to use the latest data files, but generate your output locally.

4. Command Line

AnnoGraph supports the following command line options:

-as string Aux.Store

The Auxiliary store contains all of the data used to annotate the graph nodes.

-gs string Graph.Store/gs

The Graph store is the persistent representation of the graph that AnnoGraph is displaying.

-gds string GraphDataStore/gds

The Graph Data Store is a persistent representation of the raw data that was used to construct the graph.

-nc string $ANNOGRAPH_ENV/node_commands.txt

The commands listed in this file are incorporated in the Node Commands menu of AnnoGraph.

-ec string $ANNOGRAPH_ENV/node_commands.txt

The commands listed in this file are incorporated in the Edge Commands menu of AnnoGraph.

-e string ExcludedEdges.txt

The edges (UID pairs) read from this file are excluded from the graph.

-b string IncludedUIDs.txt

If this options is specified, the graph that is displayed will be the subgraph induced by the set of UIDs in this file.

-lt int 5

This option sets the minimum weight of an edge that is to be considered part of the graph. Specifying this value will **change** the component structure of the graph. A lower value includes more edges, while a higher value excludes some edges.

-ht int 100000

This option sets the maximum weight of an edge that is to be considered part of the graph. Specifying this value will **change** the component structure of the graph. A lower value may exclude some edges, while a higher value includes more edges.

-c int -1

This option specifies the component number for AnnoGraph to display. The division of the graph into components will change as a function of –lt and –ht.

-sts string

Dump a table of graph nodes and annotations. See section of sts file.

-n string 0

Load the subgraph centered at the node with the specified UID.

-d int -1

Load the subgraph centered at the node specified by –n with diameter specified here. If not specified, loads an entire connected component.

-ex string exclude_list.txt

The user can mark edges in the graph for exclusion on subsequent invocations of AnnoGraph. The set of all edges marked in a given session is output to the exclusion list file when AnnoGraph exits.

-el edge list file (output) string

The list of edges in the graph (component?) can be output????

5. Menus

5.1. Window Menu

This menu includes useful zooming options.

5.2. Show Menu

This entries on this menu toggle annotation of the graph nodes with different types of information:

- BAC UID
  Celera UID
- BAC Genbank Accession
- BAC IID
  Celera IID
- BAC Chromosome
  Best guess chromosome.

- Center

  Center where the BAC was sequenced
- STS UID

  List of STS hits by UID
- STS Accession

  List of STS hits by Accession
- STS chrom:bin

  List of STS hits by chromosome and bin

5.3. Save Menu

- Save selected edges
- Save selected nodes

5.4. Node Commands Menu

The following commands can be invoked on a node in the graph:

- Celamy

  Display the overlay assembly of this BAC (if available)
- GEX

  Display the graph explorer view of the overlay assembly of this BAC (if available)
- Genbank Entry

  Invoke a netscape browser to view the genbank entry for this BAC
- Stats

5.5. Edge Commands Menu

The following commands can be invoked on a edge in the graph:

- GenesView

  Displays the overlay assembly views of each of the two nodes (as per Celamy node command) with the shared fragments (see Shared Fragments) highlighted, and the relevant sections of contigs the pairwise align to each other highlighted. In the window where AnnoGraph was invoked, a useful report on the plausibility of these pairwise alignments constituting a tiling is listed.
- Overlap (see ContigOverlapPlot.doc)

  A 'dot plot' display of the alignment of the contigs in the assemblies of the two BACs.
- Shared Fragments

  A pair of overlay assembly views are shown with all fragments that are shared, or involved in bridges, highlighted. Using the mouse to right-click on a highlighted fragment displays a report of the form:

  Contig <m> Frag <n> (Type:[RB]) # hits:<h> highHit:[01] bacEnd:[01]

  Where m is the contig IID, n is the frag IID, type R is a celera read, type B is a bac end (external), h is the number of BACs that recruit this particular fragment, highHit of 1 is a fragment that was **not** included in the calculation of the edge weights, and bacEnd of 1 indicates fragment with a high-variance mate (50k library or Bac end).

- Sorted Overlap (see ContigOverlapPlot.doc)

  Similar to overlap, but the display order of the contigs is changed to try to make tiling detection easier for humans.

5.6. Find Menu

Used to highlight a node in the currently displayed graph in bright green. Prompts for a node iid.

6. Buttons

6.1. Refocus

6.2. NextComponent

6.3. PrevComponent

6.4. Mark For Exclusion

6.5. Done

7. Mouse

- Shift-clicking on an edge produces a report in the top left corner of the display as follows:

  W:%f [swm:%f] [swbm:%f] [s:%f] [b:%f] [lbr:%f]

  This displays the total weight (w) of the edge, along with the components of the weight, as discussed in the following section on computing the graph. Only non-zero components are displayed.

8. Generation of the Graph

Public BACs were split into bactigs according to contig coordinates recorded in GenBank, and/or vector/contamination screening

Celera fragments were recruited to bactigs (>40bp >98% identity match) -- all fragments recruited by a BAC were recorded for each BAC. Fragments that are recuited to > 5 BACs are ignored. We also ignore fragments that are recruited to Bactigs of size < 2kb.

Weighted edges were generated by examining shared fragments or shared mate pairs between BACs. Each fragment contributes to at most one of the following weighted sums. In the following N_f represents the number of Bacs that recruited fragment f.

- swm: shared fragment ( ) with mate

  A single fragment is shared between the two BACs, and its mate hits only in one of the two BACs. Each such pair contributes contributes 1/(N_{sf} - 1) * 1/(N_{f2})

BAC1

BAC2

- swbm: shared fragment ( ) with bad mate

  A single fragment is shared between the two BACs, and its mate does NOT hit in either of the two BACs. This is a negative indication, and each such pair contributes 1/(N_{sf} - 1) * 1/(N_{f2}).

- s: shared fragment ( ) and shared fragment pairs

  A fragment that is shared between the two BACs, and it either has no mate, or its mate is also shared by both BACs. A shared fragment contributes a weight of 1/(N_{f-1}) and a shared fragment pair contributes 1/(N_{f1}-1) * 1/(N_{f2-}-1).

BAC1

BAC2

- b: bridge

  This is a pair of 2k/10k fragments, not shared between the two BACs, whose mate relationship bridges the two gaps. That is, one fragment hits the first bac and the other fragment hits the second bac. Each such pair contributes contributes

1/(N_{f1}) * 1/(N_{f2}).

BAC1

BAC2

- lb: long bridge

  This is exactly the same as a bridge, but with high variance (50k or Bac End) mates.

The weight w of an edge is computed as:

$$w = s + swm - 2 * swbm + 3 * (b + lb)$$

9. Glossary:

node:	BAC
edge:	connecting two BAC nodes when they share Celera Fragments or Fragments in one BAC have mates in the other BAC.
edge weight:	An indication of the strength of the relationship between two BACs.
component:	a collection of all BACs connected by edges. Components are numbered, starting from 0 being the largest component (with most nodes)
graph:	a collection of components (for example, whole genome, or chromosome specific)
uid:	Celera unique identifier >= 13 digits long
iid:	internal identifier used by IR (BACs have iids in the range 1 to the number of BACs)
Overlay Assembly:	An assembly of a phase 0 or phase 1 BAC that incorporates all of the Bactigs (HTGs) in the BAC and all recruited Celera reads. (Questions about the overlay assembler can be directed to Daniel Huson or Knut Reinert).
Celamy:	IR's in-house tool to view how fragments and bactigs are assembled into unitigs, contigs and scaffolds. In the context of AnnoGraph it is used to view overlay assemblies.