

# A Better ClothoTool

Jenhan Tao

16 June 16, 2011

**Clotho is a tool for engineering synthetic biological systems and managing the data which is used to create them, providing a mechanism to begin the process of creating standardized data, algorithms, and methodologies for synthetic biology. This statement implies that Clotho provides a streamlined work flow by which users can generate useful results. However, when a new user first launches Clotho, they are greeted by a disparate set of icons with unfamiliar symbols representing apps arranged neither in alphabetical order nor any other logical fashion (Figure 1). When a user finally finds the appropriate tool, they are next tasked with retrieving the appropriate sequences and etc. Current tools such as Spreadit Parts and Spreadit Features provides access to such sequences, but requires the user to crudely copy and paste the sequences (Figure 2). Copying and pasting sequences does not leverage the full capabilities of Clotho and offers no advantages over spreadsheets and non Clotho tools. To make Clotho competitive with existing tools, Clotho tools need to use the Clotho data model to provide: easier access to data and communication between Clotho tools in an integrated work flow.**

Past discussions between the Anderson lab at UC Berkeley and the Biofab has led to the creation of a so called Biofab edition of Clotho. The Biofab edition aims to provide an IDE environment for interacting with design tools (Figure 3). The most prominent differences are a set of tabs that display the list of parts in the local database and the use of tabbed panels to display tools. These two features provide the foundation for a series of improvements to Clotho. In order to provide a consistent and more integrative Clotho user experience, the following ought to be consistently implemented in Clotho tools: 1) compatibility with the Biofab edition windowing system, 2) implementation of a menu bar, 3) drag/drop support, 4) tool window dimensions. These new specifications can all be contained within a reworked ClothoTool class within the Clotho API.



Figure 1- Existing Clotho dashboard

The use of tabbed panels offers the possibility of reducing the clutter of windows generated by various Clotho tools. The use of tabs has thoroughly been tested in the space of internet browsers, and has proven to be affective. Users should not be forced to use half a dozen windows when designing a plasmid, nor should they be

prevented from doing so. Supposing that every ClothoTool was built within a JFrame, it would be a simple matter to convert to a Netbeans tabbed TopComponent.

Every ClothoTool could have a menu option to switch between the two windowing styles. The implementation of a menu bar on all Clotho apps allows not only for a windowing option, but a unique menu for each tool. ClothoTools may need disparate menu options, and thus should not share a menu bar with the IDE window. If a developer decides that a menu bar is unnecessary the menu bar provided by the ClothoTool class could be easily disabled.

A tool nested as a tab within the IDE window right next to a set of tabs that display all available Clotho objects. Instead of copying and pasting from a spreadsheet, would it not be easier to drag an object from the inventory tabs into a tool tab? The ClothoTool class can provide the appropriate event listeners so a developer can easily provide a smooth drag and drop interface. There are already several components in the Clothocore API that supports drag and drop.

File	Edit	Selection	Vector:	Part:				
	Nickname	Short Description	Sequence	Format	Author	Status		
b			GTTAATGG...	Freeform	jenn			-1
b	BBa_B0010	BBa_B0010	CCAGGCAT...	RFC10	jenn			1
b	BBa_B0012	BBa_B0012	TCACACTGG...	RFC10	jenn			1
b	BBa_B0030	BBa_B0030	ATTAAGA...	RFC10	jenn			1
b	BBa_B0032	BBa_B0032	TCACACAG...	RFC10	jenn			1
b	BBa_J23118	BBa_J23118	TTGACGGCT...	RFC10	jenn			1
b	BBa_J45004	BSMT1	ATGGAAGT...	RFC10 - CDS	jenn			1
c	BBa_J45100	p(tetR), BBa_B0...	TCCTATCA...	RFC10	jenn			1
c	BBa_J45170	p(osmY), BBa...	GCTTATGTT...	RFC10	jenn			1
b	BBa_J45993	p(osmY)	GCTTATGTT...	RFC10	jenn			1
b	BBa_K197008	BBa_K197008	GATCTGTG...	RFC10	jenn			1
b	BBa_K197019	BBa_K197019	GATCTGTG...	RFC10	jenn			1
c	BBa_K197032	BBa_M10093.B...	CTATGCTAC...	RFC10	jenn			1
b	BBa_M10093	BBa_M10093	CTATGCTAC...	RFC10	jenn			1
b	BBa_R0040	p(tetR)	TCCTATCA...	RFC10	jenn			1
c	Bca3834	BBa_J23118.B...	TTGACGGCT...	RFC10	jenn			1
b	Bjn1392	<OmpX>	GATCTGTG...	BglBricks	jenn			1
b	Bkg9239	Listerialysin	GATCTATGA...	BglBricks	jenn			2
b	test2	test2		Freeform	jenn			-1
b	yyyyyy		GTTAATGG...	Freeform	jenn			-1

Figure 2- Spreadit Parts, which allows exchange of data between ClothoTools via copy and paste of either part name, or sequence

A tool designer should also consider the dimensions of a developing tool. Screen resolutions vary widely from 1280 by 800 pixel resolution found in popular 13 inch screen laptops to monitors that can double as a television screen. A tool with too large of a window may force the use of scroll bars (Figure 4). If not included within the Clotho tool class, it ought to be placed at least in developer documentation.

By implementing these four recommendations, each ClothoTool can deliver a better experience to the user. This experience is tied to the Clotho platform, which encourages users to utilize other ClothoTools for their synthetic biology design needs. The emerging popularity of Clotho may draw developers, which already hail from several labs across the country. Widespread utilization of the Clotho platform would truly become a nexus of new design technologies.



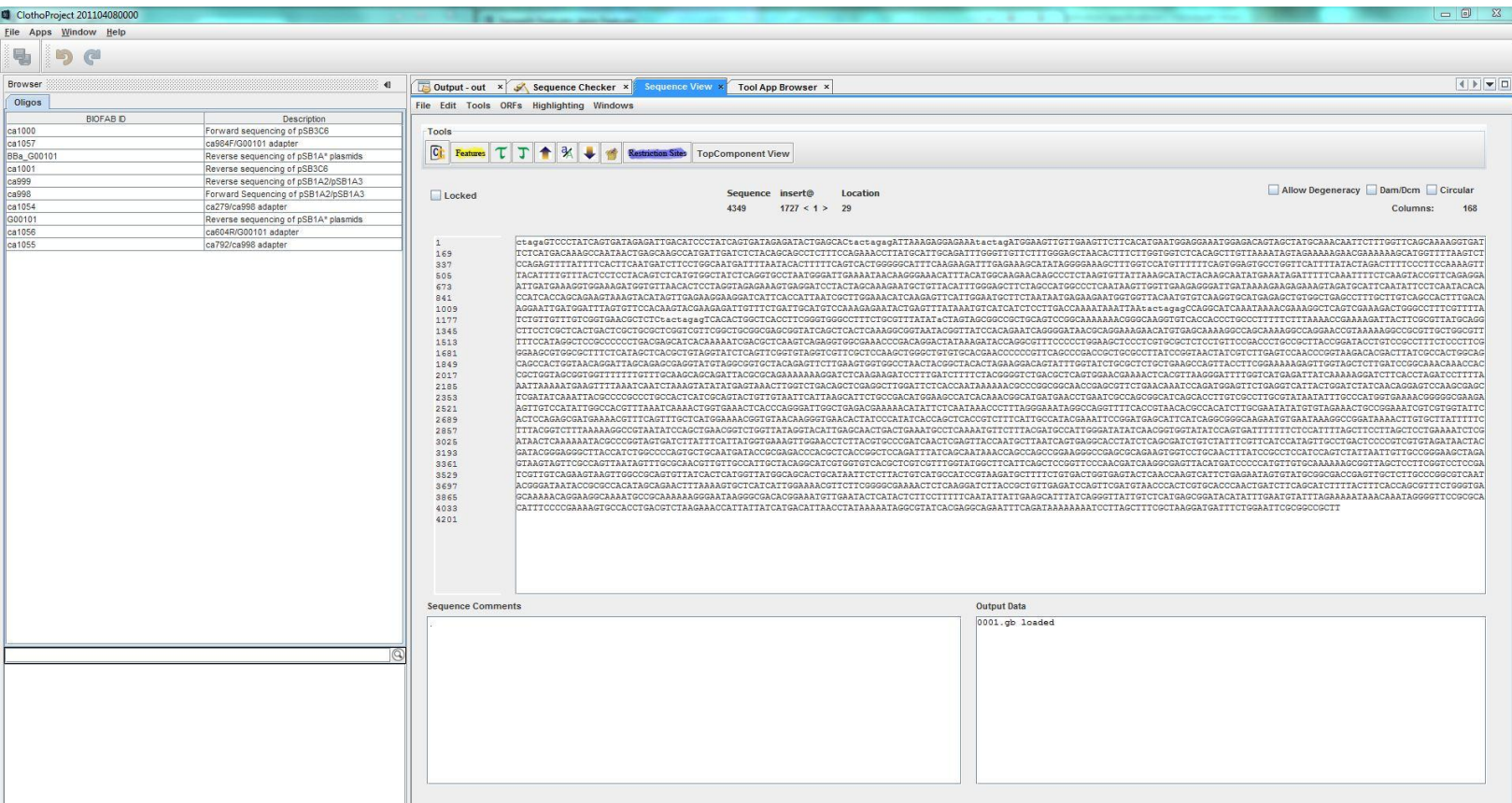


Figure 5- The ideal ClothoTool will fit perfectly into the IDE window with its own customized toolbar. Drag/drop interaction with the inventory on the left will be fully supported.