Gambit

Version 0.4.145
(Beta Release)

Derek Barnett

Marth Lab, Department of Biology

Boston College

Gambit Documentation

Table of Contents

1. Introduction

Gambit is a new cross-platform GUI (graphical user interface) application for sequence visualization and analysis.

Our software takes advantage of the indexing features of the (fairly) recently standardized BAM sequence alignment format that allows rapid access to genomic data, with minimal startup time and re-rendering delay. Gambit also supports a variety of annotation formats (BED, GFF, GFF3, VCF) “out-of-the-box” for displaying gene/region annotations as well as SNP entries.

Gambit is currently “plugin-aware” with respect to format support, meaning support for new formats can be added without needing to modify/upgrade the main program itself.

Bioinformatics analysis of sequence data currently requires specialized expertise and is rapidly becoming a bottleneck. Analysis needs of smaller biology laboratories can be served if the visualization software integrates essential analytical functionality. Such functionality includes PCR primer design to support candidate SNP validation experiments; connecting to common databases; exporting data from specific regions of the chromosome for detailed and focused analysis.

Gambit will soon be “plugin-aware” with respect to analytical tools to provide such integrated functions. One benefit from implementing the analysis system via plugins is that anyone with programming skills can readily include their own custom features: support for new file formats, additional metrics, or even custom views. Another benefit is that users need not add every plugin. You will be able customize Gambit to be as lightweight or fully-functional as your needs require.

1.1 Release Note

The current beta version (0.4.145) provides basic viewer capabilities; further analysis tools are coming soon.

1.2 License Info

The current beta version(0.4.145) is available free under the GPL 2.0 license – both “free as in speech” as well as “free as in beer”. Our lab is committed to always providing our software free of charge for academic use. In the future, Gambit may be released under a dual-license setup, where commercial users pay for a license. But for now, especially at this early beta stage in development, Gambit is free to all.

1.3 Contact Info

Feel free to contact me if you have any comments, suggestions, questions, complaints, whatever.

Derek Barnett
Marth Laboratory
Boston College – Higgins Hall
140 Commonwealth Ave
Chestnut Hill, MA 02467

Email: barnetde@bc.edu
Lab Website: http://bioinformatics.bc.edu/marthlab/Main_Page

1.4 Reporting Bugs

As a beta user, you're getting first crack at the software, but that means there will inevitably be some minor annoyances and (hopefully few) major issues. I'm counting on you to let me know if and when you notice something. If something is broken, annoying, or just feels awkward – let me know. I'll do my best to fix it. This isn't a MS product - we don't have to "just deal with it". ☺ You can play a major part in the direction that Gambit takes, if you so choose.

If you do happen to notice a bug, please submit to the bug tracker system on the project website:
http://code.google.com/p/gambit-viewer/issues/list

2. Getting Started

All necessary files (pre-compiled executables, source code, a test data set, as well as this PDF) are available from the Gambit project homepage: http://code.google.com/p/gambit-viewer/downloads/list

2.1 Running Gambit

The easiest way to get started is to download and run the pre-compiled executables.

Linux 32– /64–bit

For Linux users, download the proper version (Gambit_linux_x86.tar.gz or Gambit_linux_x64.tar.gz) for your system.

Extract the files in the archive:

```
tar –xzvf Gambit_linux_<version>.tar.gz
```

To run Gambit, use the supplied shell script. This is used for now to insure that Gambit finds the proper Qt libraries, if others are present on your system:

```
sh ./Gambit.sh
```

> *** Note** – If you are not running Gambit locally (i.e. via SSH to a server), be sure to start an X server on your local machine. Enable X11 forwarding and use one of the following techniques, depending on your local machine's OS. Be aware, however, that graphics performance will suffer when using X11 forwarding.
>
> Linux/Mac users – connect via SSH using the '-X' option.
> Mac users may have to obtain this X11 implementation for MacOSX:
> http://www.apple.com/downloads/macosx/apple/macosx_updates/x11formacosx.html
>
> Windows users – start up an X server (Xming) before executing the shell script on the server
> Xming is available free: https://sourceforge.net/projects/xming

Mac

For Mac users, simply download the Gambit_macosx_<version>.dmg package file.
Double-click the .dmg file to 'mount' the disk image on your desktop.
Once mounted, click the drive icon (or open in Finder) to open the Gambit application bundle.
Double-click Gambit (Application) to start.

Windows

Unfortunately, **no Windows 64-bit version is available** at the time of writing. Once it is available, it will be accessible in exactly the same way as the 32-bit version.

For Windows 32-bit users, download the Gambit_win_<version>.zip archive file.
Extract the contents of the .zip file (2 directories – one labeled ‘plugins’, the other ‘release’).
In the folder ‘release’, you will find Gambit.exe. Double-click to start.

2.2 Open Source

In the spirit of openness and freedom, I’ve made the source code available to those who want to tinker and/or contribute. In doing so though, please realize that Gambit is to date unpublished. I would like to publish on this someday, so please don’t scoop this right out from under me. ☺ I welcome feedback and future contributors. In fact, I believe the long-term success of Gambit will lie in a community of plugin developer/hackers and end users. But that’s down the road. In the meantime, I welcome any meaningful feedback.

Gambit is pretty straightforward to compile once you have Qt installed. An explanation on obtaining Qt and compiling Gambit can be found at the end of this document (**Appendix A**).

2.3 Performance Notes

The major performance bottleneck in Gambit occurs, when data sets become very large, with the task of graphical rendering, updating, etc. For most users, this may never become apparent. This is also an active area of my work, trying to optimize this part of the process. In the meantime, here are some tips and guidelines to limit any growing pains with super-huge data sets.

Mounting drives

The performance of X11 over a network connection is ridiculously slow. A decision was made to keep Gambit running locally on the user’s PC. However many large data sets are ridiculously big for a desktop/laptop. To get around this issue, you can mount your network shared drive locally – Gambit will detect it just like a local hard drive and load data just fine. If you need help, consult your OS documentation and/or talk to your IT guys about setting this up.

File size limitations/expectations

Due to current performance limitations, I advise against trying to open extremely large datasets (>> 100GB BAM files), or smaller data sets that have extremely large coverage (>> 200X). These files **will** actually load up if your computer has the memory available to do so, but it will take some time to show initially and update when you zoom or scroll. This will hopefully not be a relevant point to researchers working with non-mammals. However, researchers working with large mammalian data sets – such as those on the 1000 Genomes Project, for example – may want to keep this in mind and limit viewing data to single individuals or trios, as opposed to, say an entire Pilot <X> dataset.

Once again, these limitations are simply the state of the software as of writing. Optimizing these critical paths is a very high priority to me. Future updates to Gambit will deal with these extremely large data sets in a much more sophisticated manner, so that the software can keep up with such high data throughput.

3. Using Gambit

3.1 Understanding Gambit file types

When Gambit opens a data file, it considers this data to be one of 4 basic “types”: Alignment, Gene, Reference, or Snp.

Alignment – files hold the results of, well, an alignment.
Reference – a reference genome, typically FASTA
Gene – any multi-base annotation
Snp – any single-base annotation

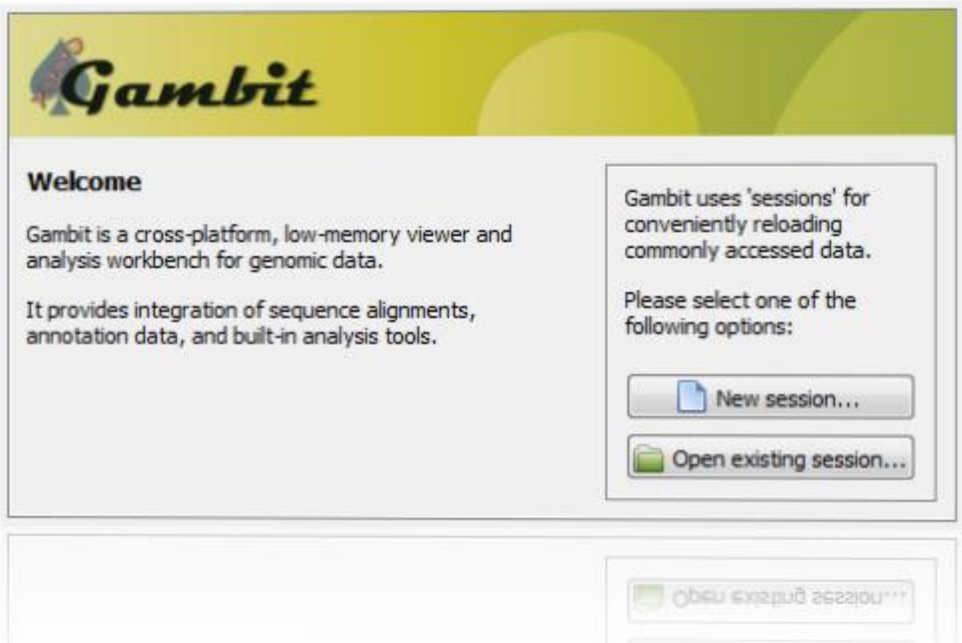
The only data required for a proper assembly view is an Alignment file along with a Reference file. Gene and Snp files are optional.

The Gene data type is currently **not limited** to only gene annotations, but can be used for any multiple-base annotations – multi-base indels, exon capture target regions, transcript annotations, etc.

This nomenclature is definitely a reasonable target for change – specifically, combining the idea of Gene and Snp data types into a more general ‘Annotation’ data type.

3.2 Using sessions

Gambit uses the concept of project sessions to keep track of currently open files. Upon starting up Gambit, you are presented with the Home screen, which contains this welcome box:



Welcome

Gambit is a cross-platform, low-memory viewer and analysis workbench for genomic data.

It provides integration of sequence alignments, annotation data, and built-in analysis tools.

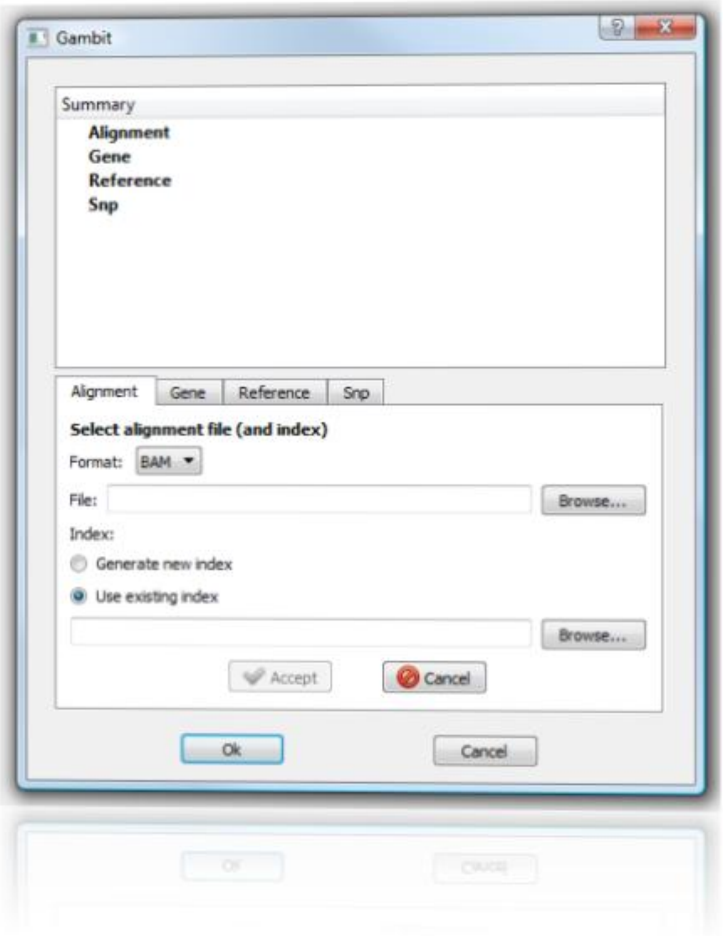
Gambit uses 'sessions' for conveniently reloading commonly accessed data.

Please select one of the following options:

This gives you the option of creating a new session or opening an existing one.

Let’s walk through creating a new session:

After clicking "New session…" this dialog will pop up:



Select alignment file (and index)

Format: BAM

File:

Index:

○ Generate new index

● Use existing index

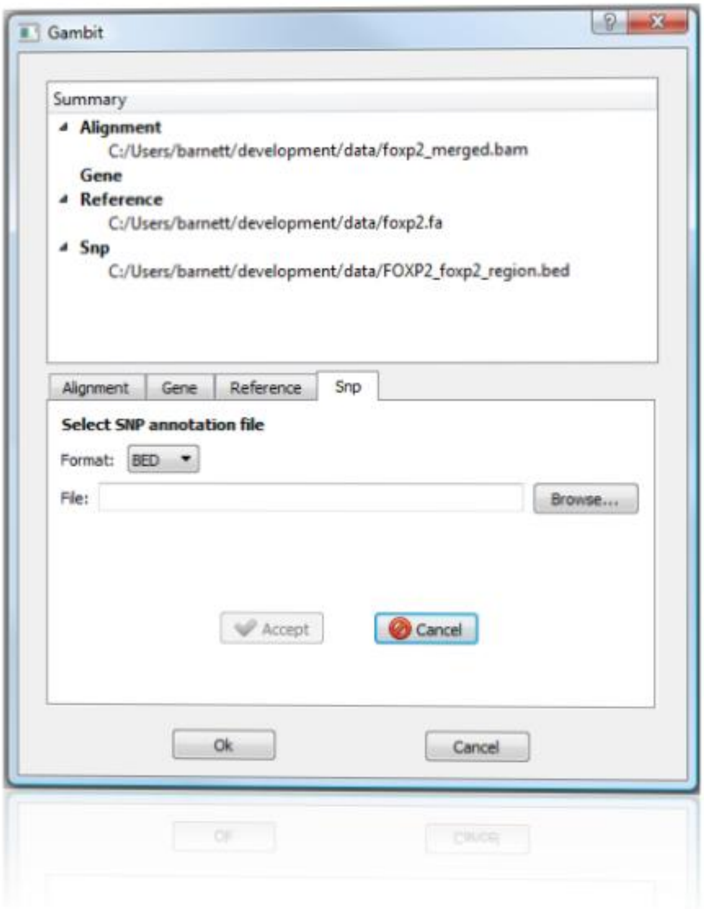
As you can see, there is a summary at the top to show all files selected so far. Beneath is the selection area. Each tab corresponds to one of the four data types.

Each selection tab has a few main components:

1 – The format drop-down box. This box will reflect all the formats known by Gambit to supply the current data type. These formats are written as plugins, so **you can add support for a new format, without requiring a new version of Gambit.**

2 – The filename entry box. Type in your alignment file here, or more likely, click the "Browse…" button to browse your file system for the alignment data you want.

3 – If the file format selected uses an index file, the next sections on the form are visible. You may choose to generate a new index or use an existing one. If an index exists in the same directory as the main data file, Gambit will automatically detect this and populate the index filename entry line. This assumes the filenames are the same, except for the expected index file suffix (i.e. the index file "dummy.bam.bai" is found if you select "dummy.bam" as your main data file).



- Alignment
  - C:/Users/barnett/development/data/foxp2_merged.bam
- Gene
- Reference
  - C:/Users/barnett/development/data/foxp2.fa
- Snp
  - C:/Users/barnett/development/data/FOXP2_foxp2_region.bed

Select SNP annotation file

Format: BED

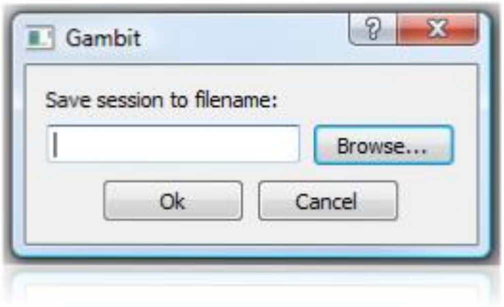
File:

4 – Once all your data for this file is selected, click "Accept" to save this file entry. You will see it added to the Summary display at the top of the dialog.

Select a different tab and continue until all your files are selected.

Click Ok to finish file selection.

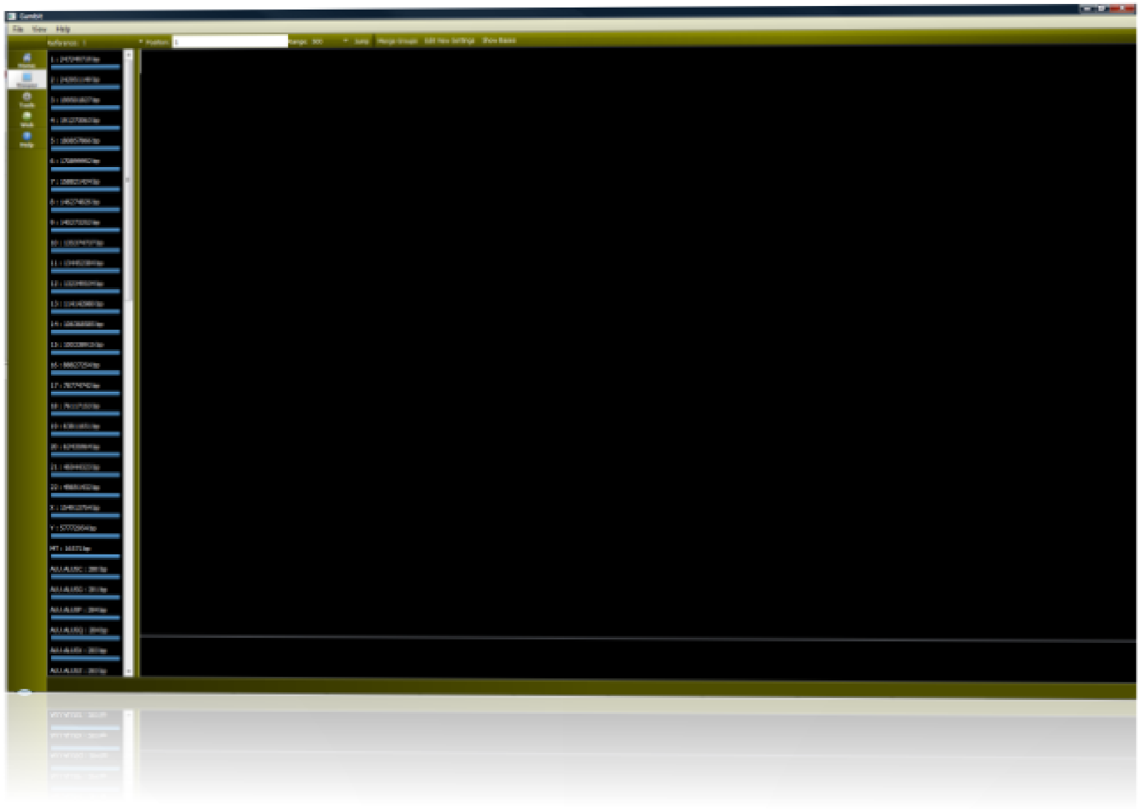
To save your session, type in a name to call your session. Please note, by default this will save your session into the Gambit application's home directory. If this is not what you want (and I would advise against it), click "Browse…" to select the folder where you want to save.



Gambit sessions are given the extension ".gss".

In the future, this session can be re-loaded by clicking the "Open existing session…" button from the Home tab shown above.

After saving your session (or opening an existing one), you will be shown a screen something like this:



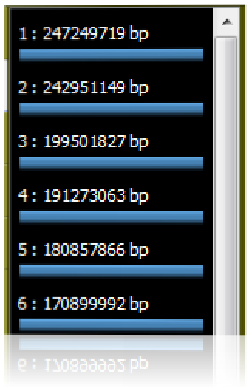
Now let's look at how to bring up some data.

3.3 Jumping to a region of interest

Gambit currently handles data in chunks which I'll refer to as regions. For example a region might consist of bases 1-500 on chromosome 21, or bases 1010001-1015000 on chromosome X. You get the idea.

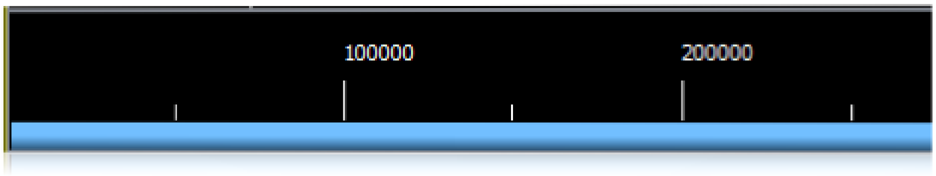
So how do we select a region? In Gambit, there are a few ways of doing this.

When you load a data session, a window near the left is populated with all the chromosomes (or other 'reference' sequences) present in the data set. The chromosome name and length are given.



1 : 247249719 bp
2 : 242951149 bp
3 : 199501827 bp
4 : 191273063 bp
5 : 180857866 bp
6 : 170899992 bp

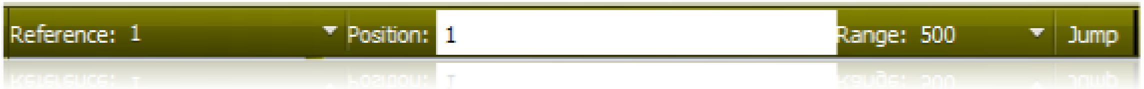
Clicking one of these bars jumps to the beginning of the chromosome.
Selecting a reference also creates a slider at the bottom of the screen, shown below.



If you click any location along this bar, Gambit will jump to the region around the coordinate you click.

Ok, that's fine for generalities, but what if you want to get a little more specific with your region?

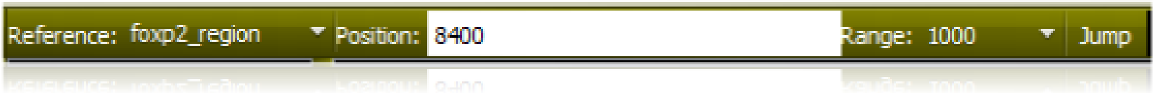
At the top of the assembly view area, you'll notice what I call the assembly toolbar. It has various actions related to the view area. On the left side of this toolbar, you'll see some controls, shown below.



Reference: 1 Position: 1 Range: 500

The far left control is a drop-down box that will allow you to select any reference in the data set.
The position edit box lets you type in any coordinate (**1-based**).
The range drop down box lets you select from a few preset ranges.
Click the "Jump" button (or press Enter from the position edit box) to jump to the desired region.

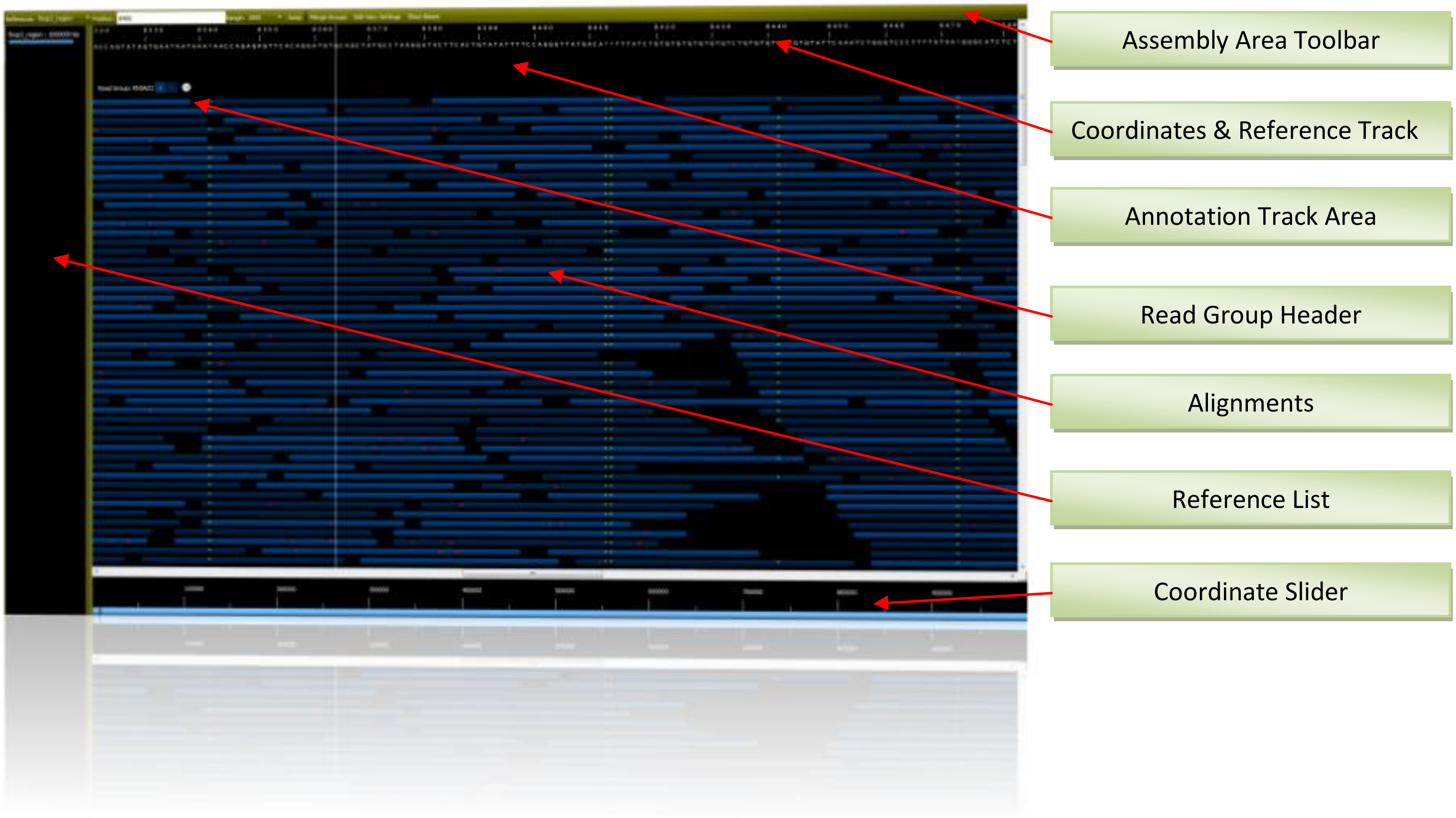
Here is an example jump that we will execute - the 1000bp surrounding position 8400 of a target region 'reference sequence'.



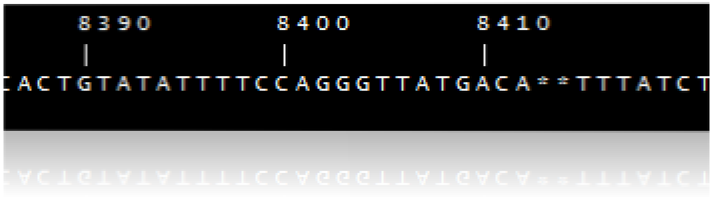
Reference: foxp2_region Position: 8400 Range: 1000

3.4 Assembly view components

Below is an example of what you might see after jumping to this region.



Let's explain what you see here in more detail.

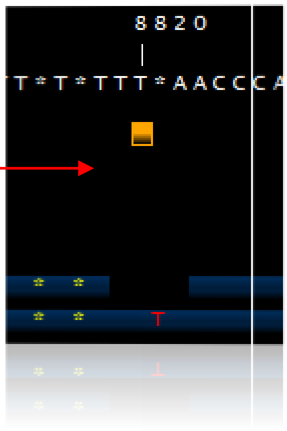


```
  8390      8400      8410
   |         |         |
ACTGTATATTTTCCAGGGTTATGACA**TTTATCT
```

Near the top of the assembly view, you will notice the reference sequence and coordinates. This is pretty self-explanatory, but what is worth noting are the '*' characters. These occur in places where at least one of the alignments in this region had an inserted base. To ensure that all coordinates line up visually – we insert these 'padding' bases to the reference, as well all other alignments that do not have this insertion.

Below this reference track lies the annotation track.

In this example, we have a BED file containing Snp-type annotations. These are shown by an orange square.

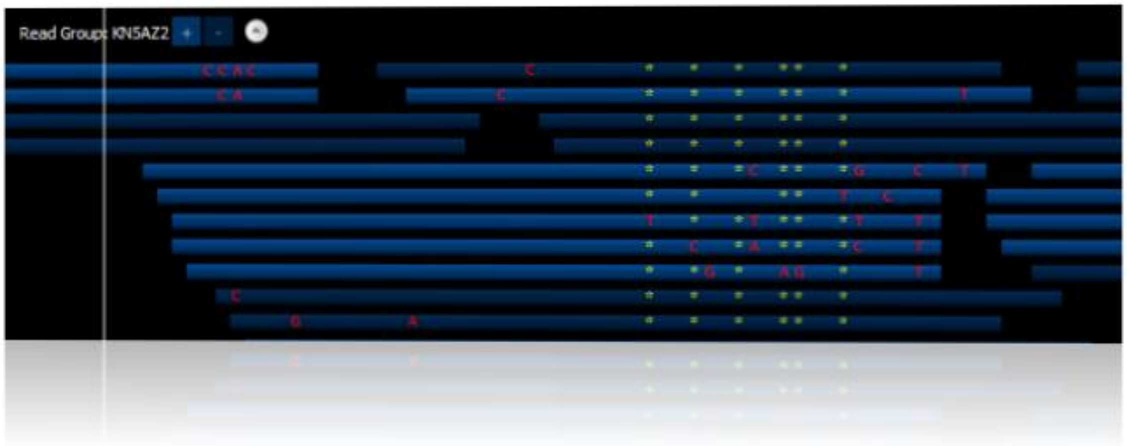


```
   8820
    |
T*T*TTT*AACCCA
```

Any Gene-type multi-base annotations are shown in the same area, only with red rectangles.

So there you have the tracks at the top of the screen. As you navigate around the assembly (explained below) you'll notice that these elements stayed fixed at the top of the view area. Thus, if you are inspecting a region of deep alignment coverage, you will always be able to see 'where you are' with respect to the reference and annotations.

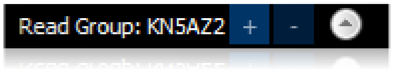
Now let's look at the alignment view area.



Alignments are lain out in Gambit using what I call a 'brick layout' (not sure if there is an official term for this) - meaning that alignments will stack up as close as they can to maximize data available on the screen. Most modern visualizers utilize this sort of layout.

Alignments in Gambit are also grouped by 'read group'. This is a common notation in the SAM/BAM world, used to identify the source of the alignment data – to distinguish between individuals, library preps, machine runs, etc. Gambit retrieves this data, if available from the alignment file.

At the top left, you will notice a read group header:

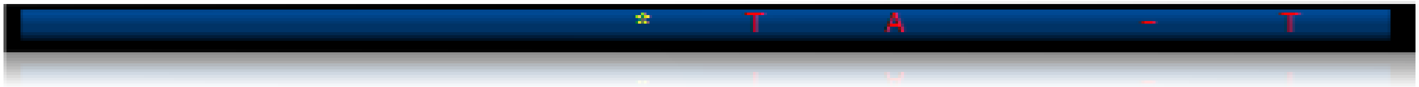


If a read group identifier is unavailable, the header will still be present, but will be labeled "Unknown".

Alignments are chiefly colored according to orientation in Gambit (other customizations are explained below). As the header tells you, the lighter blue rectangles in the view correspond to alignments aligned to the forward (+) strand, while the darker blue correspond to the reverse strand (-).

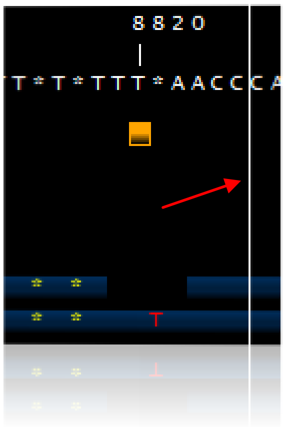
The header also provides a collapse/expand function by clicking the header icon (explained more below).

Moving on to the alignments themselves…

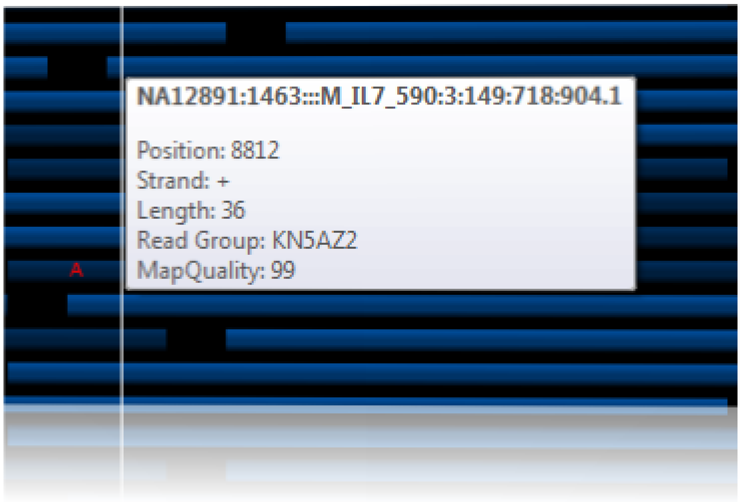


As you'll quickly notice, by default, all matching bases are not shown. This can be toggled on/off and is explained more below. All mismatches from the reference (substitutions & deletions) are shown here in red. All padding positions are shown as yellow '*'s. These colors are also customizable.

You may have noticed a white vertical line in some of the images shown previously. I've shown one such image again to the right. This is a mouse-tracking cursor bar. This bar will follow your mouse around. It serves as a visual aid to help keep track of 'where you are' in cases of deep coverage and to line up alleles across multiple reads.



Another feature of the assembly view is mouse-over tooltips. In other words, if you hover your mouse over an Alignment, Gene, or Snp item, you will see a tooltip pop up with additional data. See below:



NA12891:1463:::M_IL7_590:3:149:718:904.1

Position: 8812
Strand: +
Length: 36
Read Group: KN5AZ2
MapQuality: 99

3.5 Navigating the assembly view

Intuitive naviagation around the assembly view area is key to a useful interface. Scrolling and zooming are pretty straightforward in Gambit.

Scrolling (horizontally or vertically) can be achieved in three ways:

The first is to use the scroll bars present (if needed) at the bottom and right side of the view.
The second is to click within the view and drag the view scene the direction you want it to move.
The third (which is really just a variation on click-dragging) is to use iPhone-style 'kinetic scrolling'. This is achieved by pressing the mouse, making a quick drag in the direction you want the scene to move, then releasing. This sounds complicated to describe but is easy to use in practice. If you're unsure how to accomplish, grab a friend that uses an iPhone and ask her to demonstrate the concept.

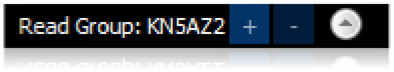
Zooming can be done in one of two ways:

The first is using the View menu on the main Gambit menu bar. There you will see the options Zoom In, Zoom Out, and Reset. Clicking either Zoom option, zooms in/out by 10%. Clicking Reset will return the view to the default zoom level.
The second method is to use (OS-specific) keyboard shortcuts. These are listed next to the options in the View menu. For example, on Windows, these are Ctrl + +(Zoom In), Crtl + - (Zoom Out), and F5 (Reset).

3.6 Customizing view options

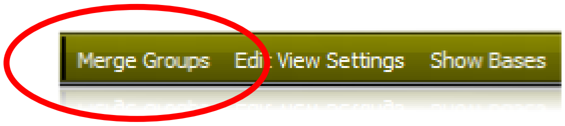
The assembly view area also provides some handy customizations to tweak the display to your tastes or needs. Options include collapse/expanding individual read groups, merging/splitting of all groups, modification of color schemes, highlighting/dimming alignments according to certain rules, and toggling the visibility of matching bases.

Collapsing individual read groups can be useful when there a many groups visible at a time. You may only be interested in comparing a subset of these. To collapse a read group, simply click the up arrow icon on the read group header, shown here:



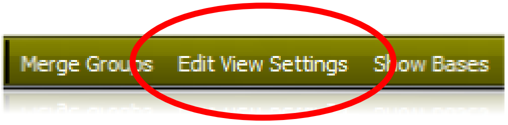
The header will still be visible, but alignments will be hidden. Click the down arrow icon to expand the group and re-display alignments.

Sometimes you don't care about the individual read groups, and want to see all the data in that region combined into one large 'brick layout'. To do this simply click the 'Merge Groups' button on the assembly toolbar:

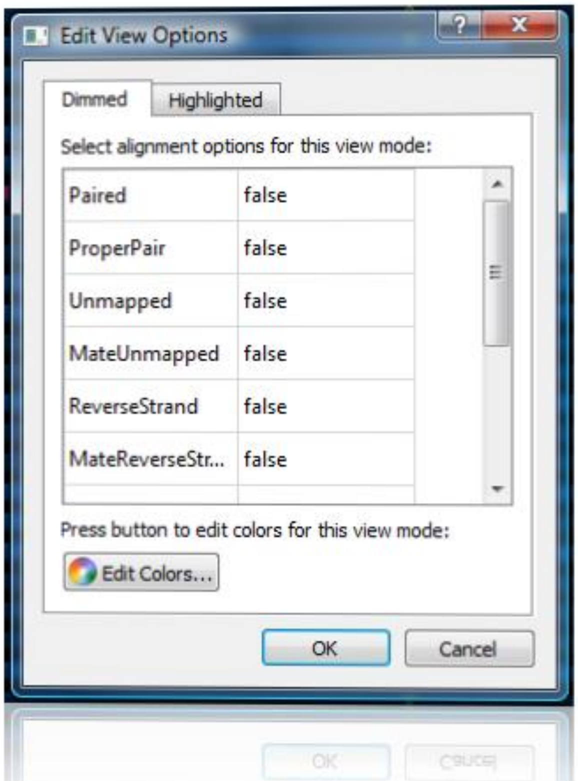


The toolbar button will change to 'Split Groups'. Click this button to re-display the alignments according to their read group.

To modify the color schemes and highlight/dim options, click the 'Edit View Settings' button on the assembly toolbar:



This will bring up a settings dialog, shown to the right:



Select alignment options for this view mode:

Paired	false
ProperPair	false
Unmapped	false
MateUnmapped	false
ReverseStrand	false
MateReverseStr...	false

Press button to edit colors for this view mode:

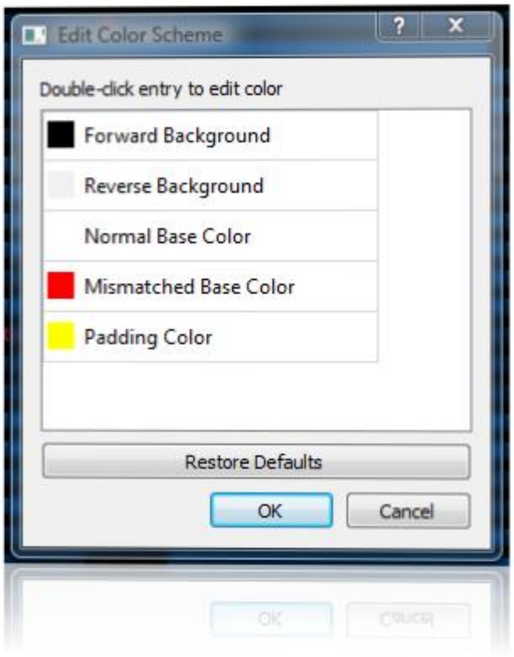
This dialog will allow you to set the rules for dimming or highlighting alignments in the view area. These rules are based on the SAM/BAM alignment flag. For more information, see the SAM/BAM documentation, available here:

http://samtools.sourceforge.net

For example, you can choose to dim all alignments whose mate is unmapped. In this case, set the MateUnmapped value to true, and click Ok.

The same values are available on the Highlighted tab, to allow you to select rules for highlighting alignments.

If you would like to modify the color schemes for Dimmed or Highlighted reads, click the “Edit Colors…” button. This will bring up another dialog, shown to the right:



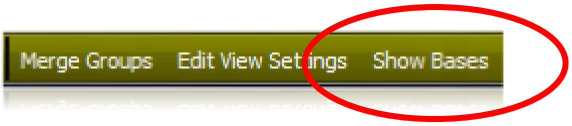
Double-click entry to edit color

- ■ Forward Background
- □ Reverse Background
- Normal Base Color
- 🟥 Mismatched Base Color
- 🟨 Padding Color

Double clicking an entry will pop up a color-chooser dialog to select a new color. Do this for any colors you would like to modify. Click Ok to apply changes.

If you find you don’t like your changes and just want to go back to the default settings, click the “Restore Defaults” button.

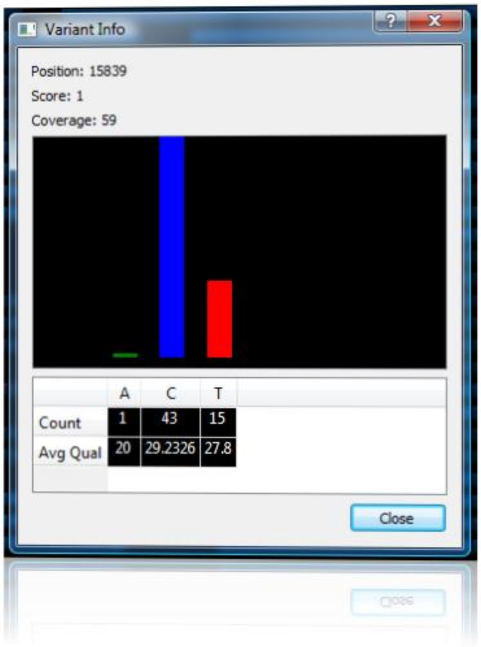
By default, Gambit does not show matching bases. This is done to reduce visual clutter on the screen. If you would like to see of those bases, click the “Show Bases” button on the assembly toolbar:



The label will change to “Hide Bases”. Click this to re-hide all matching bases.

3.7 Additional features

There is a summary dialog available for annotated Snp-type entries. If you click one of these (the orange squares), you will see a dialog something like the one shown here:



Position: 15839
Score: 1
Coverage: 59

	A	C	T
Count	1	43	15
Avg Qual	20	29.2326	27.8

This dialog provides some useful summary diagnostic values:
Snp position
Snp score (from annotation file)
Coverage at this position

In addition, the dialog shows a coverage histogram by allele. The summary table beneath gives both the coverage count and average base quality for this allele.

In this case, we have a likely C/T SNP, with a sequencing error producing the single A.

4. Upcoming Features

Gambit is nowhere near complete. Here is a sample of the things that will be coming online as soon as they are ready.

Analysis plugins

Unfortunately, Gambit doesn't currently provide much in the way of analysis. However, that will change soon. A whole plugin analysis and results graphing system will be put in place that will allow programmers to write either new tools or wrappers around their favorite analysis tools, customizing Gambit to suit their own tastes/needs. And (hopefully) they can make these add-ons public for the use of the community at large. This might turn out to be an altruistic pipe-dream, but the success of other open-source, community-driven projects is encouraging.

Web/DB integration

Annotations are great for what they are, but to provide the researcher with more information it will be useful to link up with (or at least redirect to) public databases. For example, discovering that a particular indel mutation lies in an annotated exon might be useful of itself, but a one-click link to a Pubmed search for that gene could yield the researcher even more contextual information.

Scripting interface

A rich scripting interface would also be useful, especially once analysis plugin tools are in place. This would allow a researcher to automate common tasks into a single 'macro' to run, or allow batch processing for multiple data sets. This sort of interface would require some basic programming skills (likely using JavaScript) but would not require the coding muscle needed to write a fully functional plugin.

Appendix A - Compiling from source code (details)

1. Obtain Qt libraries

If you want to compile Gambit from source, you'll first need to obtain the Qt libraries. Qt is a cross-platform API that, among other nifty features, provides the GUI interface for Gambit. For performance reasons, you really should have Qt 4.5+, but Gambit will work with any version of Qt 4.0+. To obtain the Qt libraries (or their IDE, Qt Creator, which I highly recommend for Qt development), visit their website: http://qt.nokia.com/ and follow the instructions found there. Qt is pretty well documented, but if you have any trouble, the helpful Qt gurus over at QtCentre (http://www.qtcentre.org/forum) are a great resource.

2. Compile Gambit plugins

First, you'll need to build the format reader plugins. The plugins can be found in the <GambitHome>/src/SessionManager/FileManager/FormatReaderPlugins directory. Each plugin has its own subdirectory.

Using QtCreator (1.3):

Open a plugin project: <PluginName>.pro
Select Build > Set Build Configuration > Release
Select Build > Build Project "<PluginName>"

From the command line:

Enter a plugin directory, and type the following:

```
qmake -config release
make
```

Do this for each plugin.
Each plugin's build system is set to insure that the resulting library (.so or .dll) will end up in the proper directory for Gambit to locate it on startup.

3. Compile Gambit executable

Next, build Gambit itself. The Gambit project file is in the <GambitHome> directory.

Using QtCreator (1.3):

Open Gambit.pro
Select Build > Set Build Configuration > Release
Select Build > Build Project "Gambit"

From the command line:

From the command line, type the following:

```
qmake -config release
make
```