

ZENBU

an omics data collaboration, data
processing and visualization system

Contents

Articles

ZENBU Introduction	1
	1
ZENBU Genome Browser (gLyphs)	3
ZENBU gLyphs genome browser	3
Track visualization styles	14
Configuring Tracks	20
Data Download	25
TrackCaching System	28
Region Location	30
Data Explorer interface	33
Data Explorer Interface	33
Metadata searching	43
Metadata editing	44
User Collaborations	45
Secured user profiles	45
Data sharing collaborations	46
Data Uploading	50
Data loading	50
BAM and SAM file support	52
BED file support	53
GFF and GTF file support	55
OSCtable	57
Concepts	66
ZENBU Configurations	66
Data Sources	67
Data Stream Pool	70
Data Abstraction Model	73
References	

Article Sources and Contributors	77
Image Sources, Licenses and Contributors	78

Article Licenses

License	81
---------	----

ZENBU Introduction

Welcome to ZENBU [1]

ZENBU is a data integration, **data processing**, and visualization system based around three main web interfaces : an expression data enhanced **genome browser** interface, a secured **user system** for **data upload** and **secured data sharing**, and a **data explorer** interface to find and manipulate data across the many supported experimental data types and to find shared **user configurations**

One of the key differences is that ZENBU allows for novel data exploration through data integration and "**on-demand**" **data processing** within the system. This means that more raw or unprocessed data can be loaded into the ZENBU system, and then ZENBU can perform many of the basic data manipulations that previously required bioinformatics experts with knowledge of the unix command line and a collection of bioinformatics tools. In ZENBU, the data is not a static picture, but instead it is a living melting pot where scientists and can explore and discover. Have a look at our **case studies** to see powerful examples of the ZENBU data processing and visualization capabilities.

Another key concept in ZENBU is that of **data-pooling** from multiple **data sources** into a single merged **Track**. It is becoming much easier to do many experiments within a study. The simple process of managing different experimental combinations into different visualization tracks is becoming unmanagable. Data-pooling allows one to easily compare experimental expression within a series of related experiments that would previously require bioinformaticians to externally process each group analysis and upload each as different precalculated visualization tracks. With ZENBU, the data can be loaded independently and the system can perform the pooling and group analysis. Because the system performs the pooling/group operations, the data can be **interactively explored** via region selection within a pooled track or through filtering of experiments within the pool. These **realtime interactions** within a pooled data track are immediately reflected in both the expression profile visualization and in the "experimental expression bar graph".

ZENBU also provides a platform for scientific data social-networking through a **secured user environment** for **data upload** and controlled **data sharing** within user managed **collaborations**. Collaborations and data sharing are managed in a facebook style of "friend requests" providing users with the flexibility to create and manage their own collaborations without needing central administrators. ZENBU also provides guest access to view published, public data and without any data upload functions. **User profiles** are available to anyone and are managed through **OpenID** [2] cooperation with major sites like google [3], yahoo [4], mixi [5], genomespace.org [6] and many others.

ZENBU is divided into 3 main interface sections

gLyphs Genome Browser

Data Explorer (DEX)

User and Collaboration systems

- **gLyphs Genome Browser** interface : a highly user configurable and expression data enhanced genome browser with built in "on demand data processing". It is built on the concept of Tracks which consist of pooled data source, Data Processing, and visualization. It allows for various on-demand data manipulation and visualization of both raw mapping data or analyzed annotations uploaded from BAM, BED, GTF, and OSCTable files. It also provides interfaces for **data downloading** and interfaces for sharing through a flexible user defined **configurations** system for Tracks, Views and Scripts.
- **Data Explorer** interface to search the data which have been loaded into data sources of the system. This interface also allows for searching the user configurations of Tracks, Views and data processing scripts and to launch Views into the genome browser.
- **User system** for secure and controlled data sharing of user uploaded data with collaborations of other users.

Please have a look at the table of contents to see an overview of this documentation.

The entire wiki is also available in the form of **PDF guides**.

ZENBU is also available for **distribution** and **local installation**

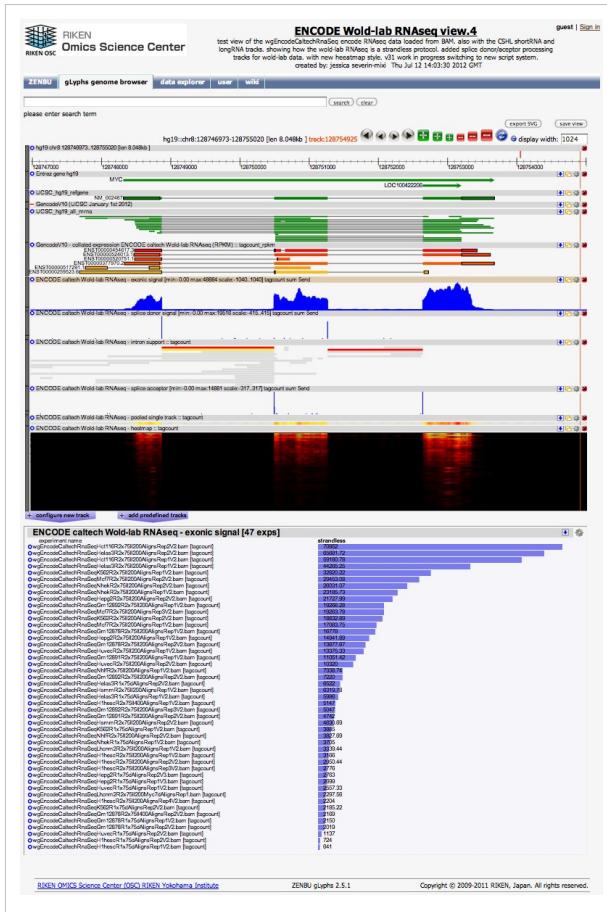
References

- [1] <http://fantom.gsc.riken.jp/zenbu/>
- [2] <http://openid.net/>
- [3] <http://developers.google.com/accounts/docs/OpenID>
- [4] <http://developer.yahoo.com/openid/>
- [5] <http://developer.mixi.co.jp/openid/>
- [6] <http://www.genomespace.org/>

ZENBU Genome Browser (gLyphs)

ZENBU gLyphs genome browser

The ZENBU genome browser [gLyphs] is designed to provide an advanced tool for not only visualizing genomics data, but also for interactively processing the data. While most genome browsers provide static visualization images, data in the ZENBU system is dynamically manipulatable by the users of the system.



Views are collections of tracks

The ZENBU genome browser is launched with a previously saved **view configuration**. A view is simply a collection of tracks with a title and description. This can be considered like an interactive figure of a paper. Any user who is logged into the system can save and share views with other users. This is done through the **save view** button at the top of the genome browser.

Users can manipulate the order of tracks in their views through "click and drag" of the side-bar or title-bar.

Each *view* receives a unique URL for sharing with your collaborators via email or for bookmarking. For example this URL links to the ZENBU genome browser view shown above

http://fantom.osco.riken.jp/zenbu/gLyphs/#config=l_D-jGt1IlehEahizVAMeB;loc=hg19::chr8:128746973..128755020

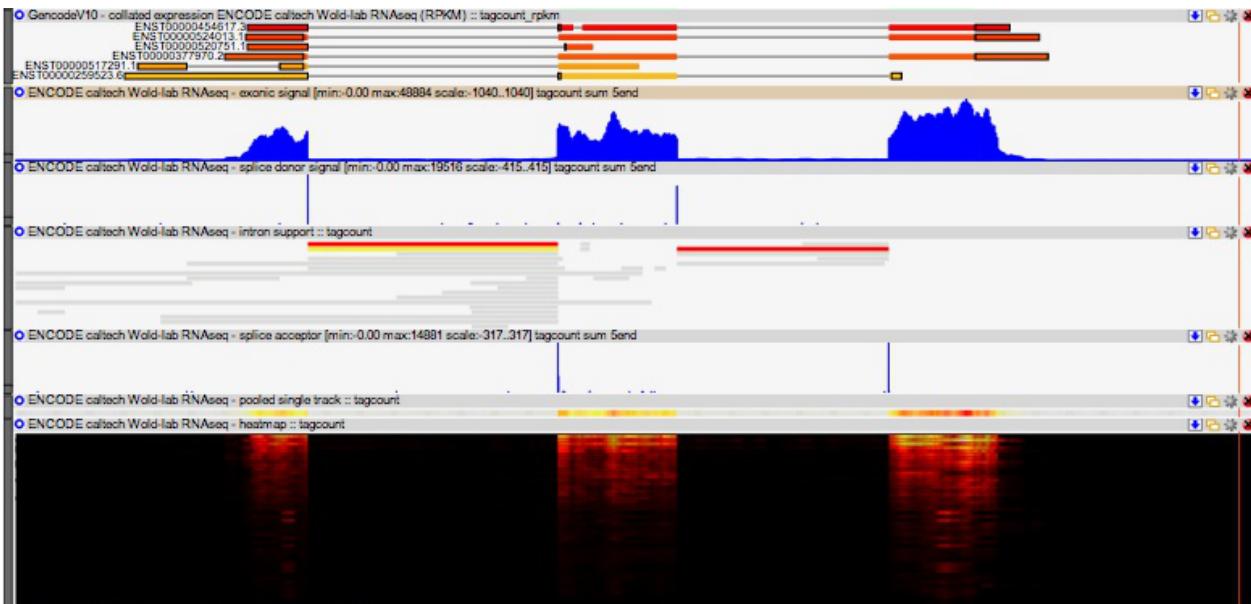
The system will also automatically save temporary views as the user makes changes. This autosave enables the use of the web browser "back arrow" to *undo* changes. Although these temporary views are in the system with unique URLs, they are not permanent and will be deleted after a period of time. To ensure your view is permanent and can be re-accessed in the future, users should login and explicitly save the view.

Multiple views on the same data

Unlike other genome browsers, ZENBU does not need to link the track visualization to specific upload file formats. This means with ZENBU, the user can upload their data once and then manipulate it in ZENBU to **create many views with the same data**. Where other genome browsers like UCSC or IGV require many different *visualization file formats* to support each different visualization track style, ZENBU can focus on a few common **data interchange** file formats (BAM, BED, GTF/GFF, OSCTable). This makes ZENBU more like a bioinformatics processing tool rather than just a visualization tool (genome browser).

Because the data is loaded once and visualization/processing is performed by ZENBU, the data content of all tracks is guaranteed to be synchronized and one can see exactly how each track was made by looking at the ZENBU data processing scripting in each track. This makes the tracks of ZENBU "data transparent"

In this example 47 RNAseq experiments were loaded once from 47 BAM files and then dynamically pooled and **processed** by ZENBU to create 7 different visualization tracks: 1) RNAseq expression collated into Gencode genes, 2) RNAseq exonic-only expression signal strength 3) alignments processed into "observed intron" evidence 4) intron evidence processed into splice acceptor sites 5) intron evidence processed into splice donor sites 6) RNAseq exonic signal as a single spectrum 7) RNAseq exonic signal as experiment-heatmap showing both genome position and experimental differential expression.



For many more examples of how ZENBU can process and visualize the same uploaded data into many different visualizations, please check out both the **data processing** sections and the **experimental case studies** sections of the documentation.

Tracks for data visualization and processing

The main aspect of the ZENBU genome browser is the concept of a track. The tracks in ZENBU are completely user definable and encapsulate the concepts of

1. dynamic data source pooling to create a track *virtual DataSource* by choosing a collection (one or more) of primary data sources
2. data processing
3. data visualization

Data pooling

The ZENBU system allows for the dynamic creation of merged data sets referred to as **data stream pools**. This provides for a great deal of flexibility when configuring tracks. It no longer is necessary to pre-merge your datasets prior to upload. Data can be loaded in an atomic, singular manner and then ZENBU can perform the mix-and-matching of data sources when users configure their tracks.

A Data Stream Pool can also be thought of as the *virtual DataSource* for a ZENBU track configuration

Dynamic track based data processing

One of unique features of the ZENBU system is the ability to apply data processing and analysis on-demand at query time and as part of the visualization process. This means that raw or unprocessed data can be loaded into the ZENBU system which translates it into the internal Data Model, and then ZENBU can perform many of the data manipulations and analysis that previously required bioinformatics experts with knowledge of the unix command line and a collection of bioinformatics tools.

The data processing system is applied on a track level at query time. This means that no intermediary result needs to be stored in a database or on disk. This allows the user to modify processing parameters and immediately see the effect of the change in the visualization. It also makes the system very fast since data is processed in memory and there is no overhead of reading and writing to slow disks.

Because data processing is applied on each track, and tracks are loaded independently, there is a level of parallelism inherent in the design of the system. The processed data result generated by ZENBU on-demand can also be downloaded into data files for further analysis by external systems like R, BioConductor, or BioPython.

Data processing is controlled through a Scripting system based on chaining Processing modules together in a manner similar to digital signal processing [1]

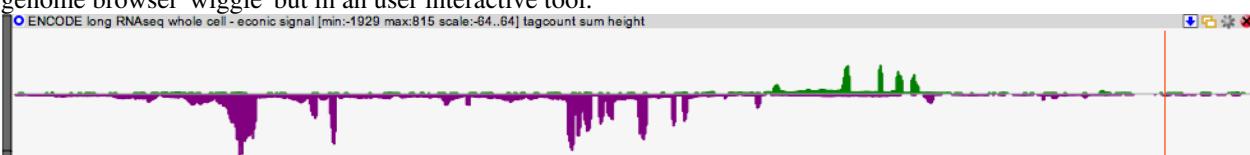
Visualization

The Tracks in the ZENBU gLyphs genome browser fall into three main categories of **visualization** styles

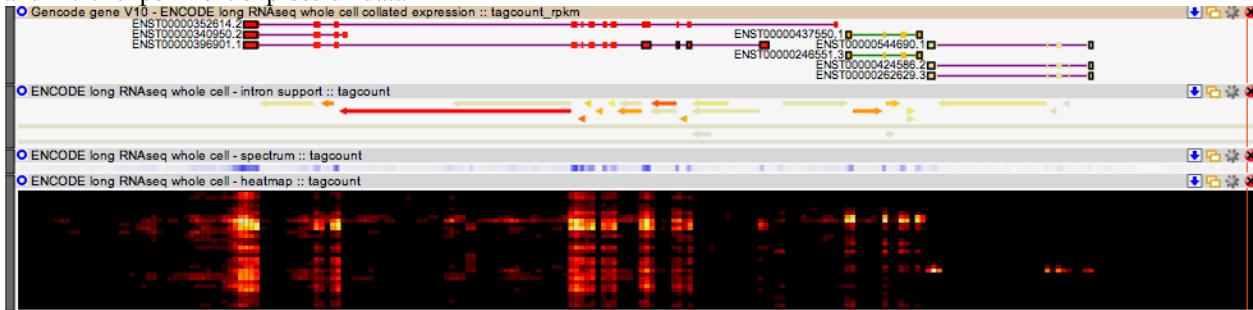
- **Annotation tracks:** where the data sources only contain genomic information and no expression



- **Expression tracks:** where expression level is displayed without feature boundaries in a style similar to the UCSC genome browser 'wiggle' but in an user interactive tool.



- **Hybrid tracks:** ZENBU enhanced visualization which allow for processed data to contain both genomic features and multi-experiment expression data



For details on all the different visualization styles and configuration options, please refer to the [Track Visualization Styles](#) section of the documentation.

Downloading processed track data

Since tracks contain dynamic pooling of data and data processing, the data output of a ZENBU track may be useful for bioinformaticians to utilize as part of external data analysis. To enable this each track has the ability to export its processed data into local files via the **download data** control

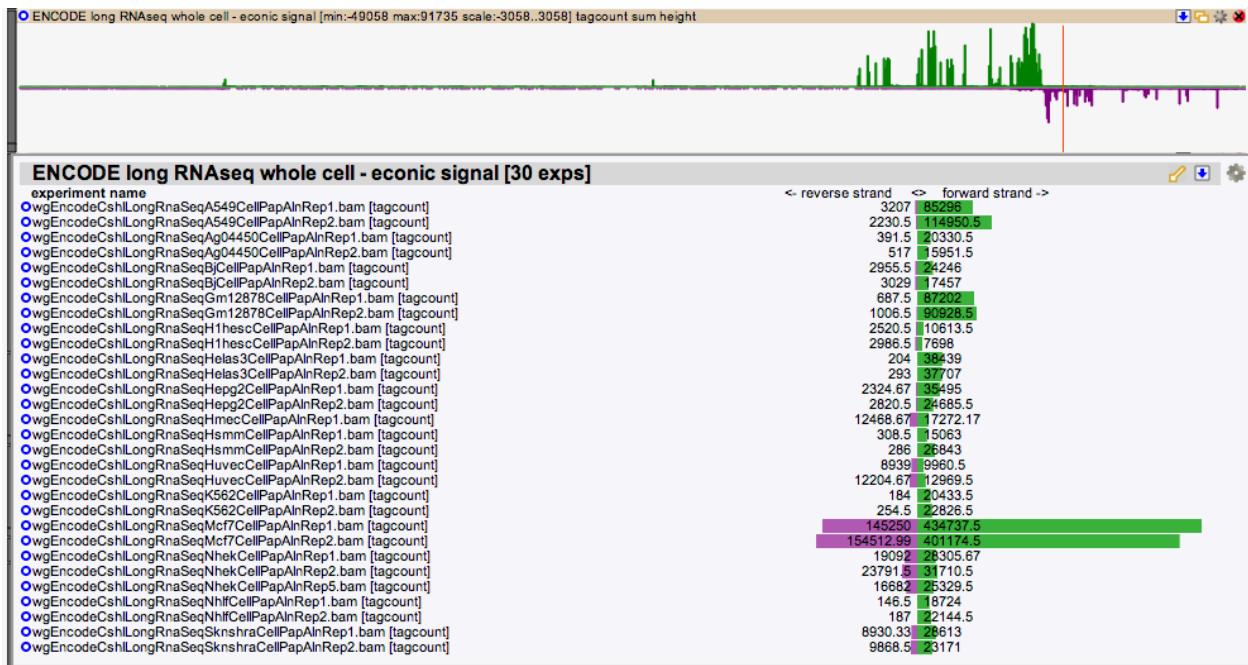
For details please refer to the [Data Download](#) section of the documentation.

Experiment expression data graph

The "Experiment expression data graph" is a track-linked display form ZENBU expression and hybrid tracks which shows the differential expression between experiments. It works in coordination with the "selected track". Clicking on the title bar or inside a track will select it. ZENBU tracks can be thought of as three-dimension data with experiments, genomic-coordinates and expression values. The track compresses the experiment dimension and the Experiment-expression-graph compresses the spatial dimension.

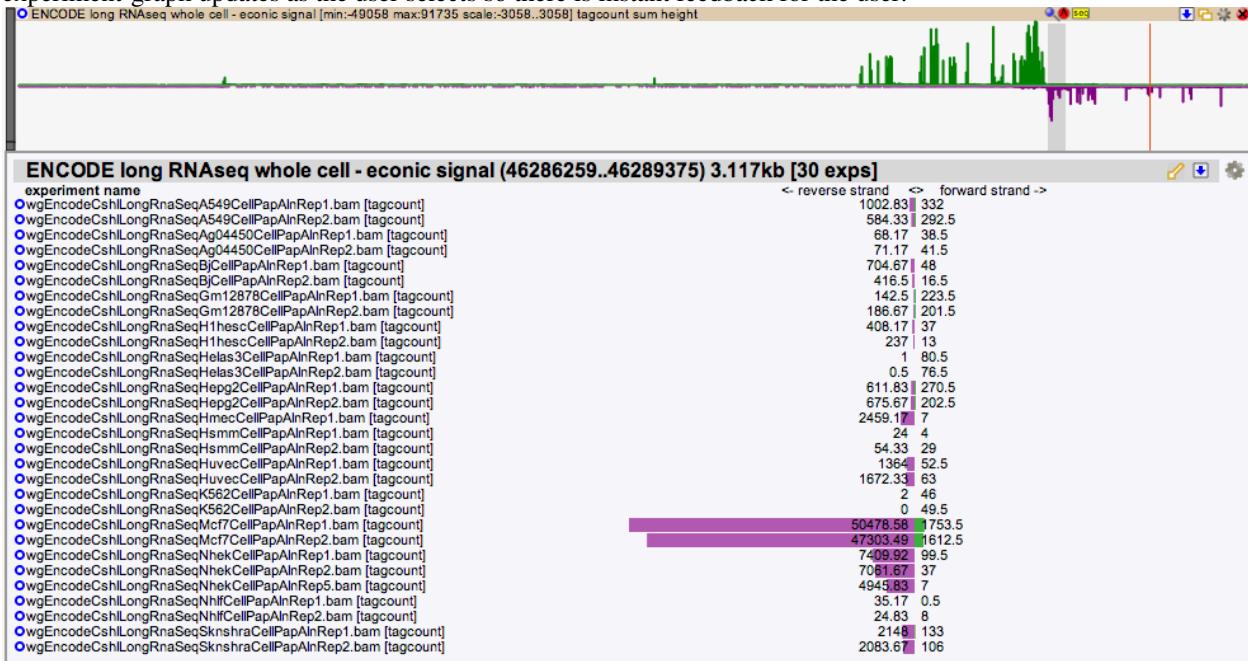
Visible region collation

By default the experiment-graph selects the entire visible region for compressing spatial information to calculate the expression value for each experiment. The calculation across spatial region is by default a summation, but can be min or max depending on how the track was configured.



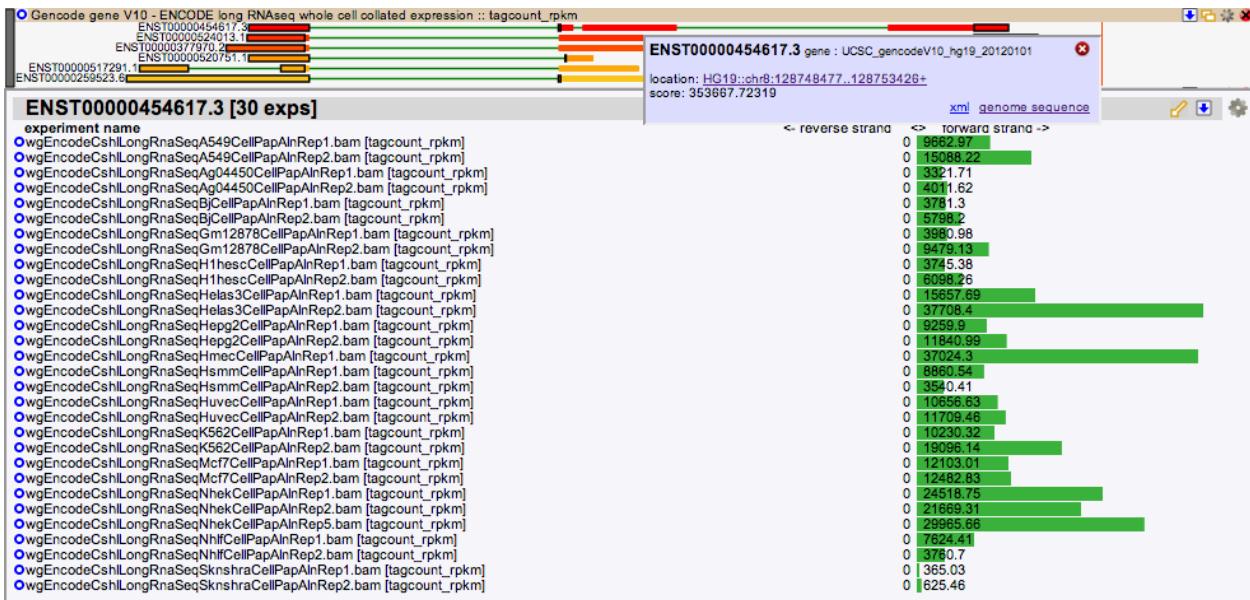
Selected region collation

It is also possible for the user to select specific regions within the track and only show the experiment expression under the selection. Selecting a region in a track is performed with simple click-and-drag like selecting text. This interaction with the view and data can be very useful for focusing on specific regions of interest. The experiment-graph updates as the user selects so there is instant feedback for the user.



Hybrid feature selection

For hybrid tracks with genomics-features which have collated expression, selection of an individual feature show the experiment-expression of only that feature.



Panel controls

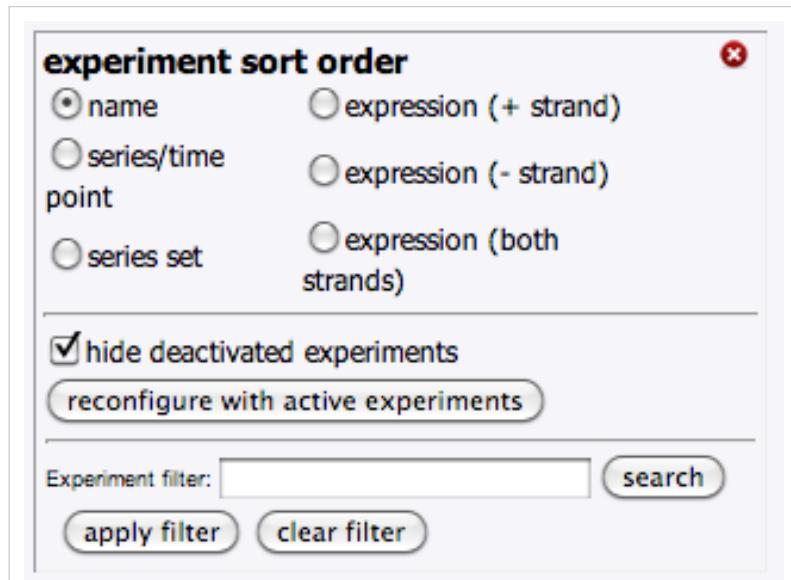
The Experiment-expression graph can be moved to a different location in the screen by click-and-dragging on its title-bar. To reset it back to its dock at the bottom of the view, click the "reattach view" widget .

The data currently displayed in panel can be exported into a tab-table with the "export data" widget . This will pop up a panel with the data which the user can copy-and-paste into excel or another application.

The Experiment-expression graph panel has several configuration options which can be accessed by clicking the "configure" widget .

The panel allows for altering the sort-order of the experiments in the view.

- name** : sort based on experiment name. Can also be activated by clicking the title "experiment name" in the view.
- expression + strand** : sort based on +strand expression from most expressed to least expressed. Also activated by clicking title "forward strand ->" in the panel
- expression - strand** : sort based on +strand expression. Also activated by clicking title "<- reverse strand" in the panel.
- expression both strands** : sort based on combined expression on both strands. Also activated by clicking title "<>" in the panel.
- series/time point** : if the experiment is loaded with metadata of tag "eedb:series_name" and "eedb:series_point" this will sort first by time_point and then by series_name within that time point. This can be very useful for time-course datasets like was collected in FANTOM4.
- series set** : if the experiment is loaded with metadata of tag "eedb:series_name" this will sort first by series_name grouping related series together. If "eedb:series_point" is present it will then do a secondary sort on time-point.



This config panel also allows for activating/deactivating individual experiments. This cause the linked expression track to recalculate its experiment summed expression value. For expression signal tracks, the shape of signal will change. This can be considered "soft filtering" of the data-source pool of the track. This accomplished by clicking the activate/deactivate widget next to the experiment name.

If the panel option "hide deactivated experiments" is selected, then the deactivated experiments are not displayed in the Experiment-graph panel.

Sometimes it is useful to create a deep data-pooled track and then interactively filter it for experiments based on the experiments metadata after loading. This can be useful when doing exploratory work on new datasets. The **Experiment filter search** allows for searching experiments and deactivating any experiment which does not match the search.

- **search** : perform the metadata search to preview which experiments will remain.
- **apply filter** : applies the metadata search logic to select active experiments. all experiments not matching the search will be deactivated.
- **clear filter** : removes the filter and activates all experiments.

Selecting regions in tracks

All tracks in the genome browser allow for selecting regions. For expression and hybrid tracks this has immediate effects on the Experiment-graph.

In this example we have zoomed out to a 1megabase region of genome and found some highly expressed "introns" so we selected the region around them

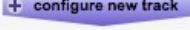


This selection also enables several options via widgets at the top of the selection.

- : **magnify** zooms region into selected region
- : **genome sequence** returns the genome sequence under the selection. This is only available for genomes which have had their sequence loaded into ZENBU. This will pop-up a panel with the sequence and the user and select and copy-paste the sequence into another a program.

Adding tracks to the view

Creating new tracks

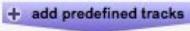
New tracks can be added to views through the **configure new track** control  which brings up this "configure new track" interface

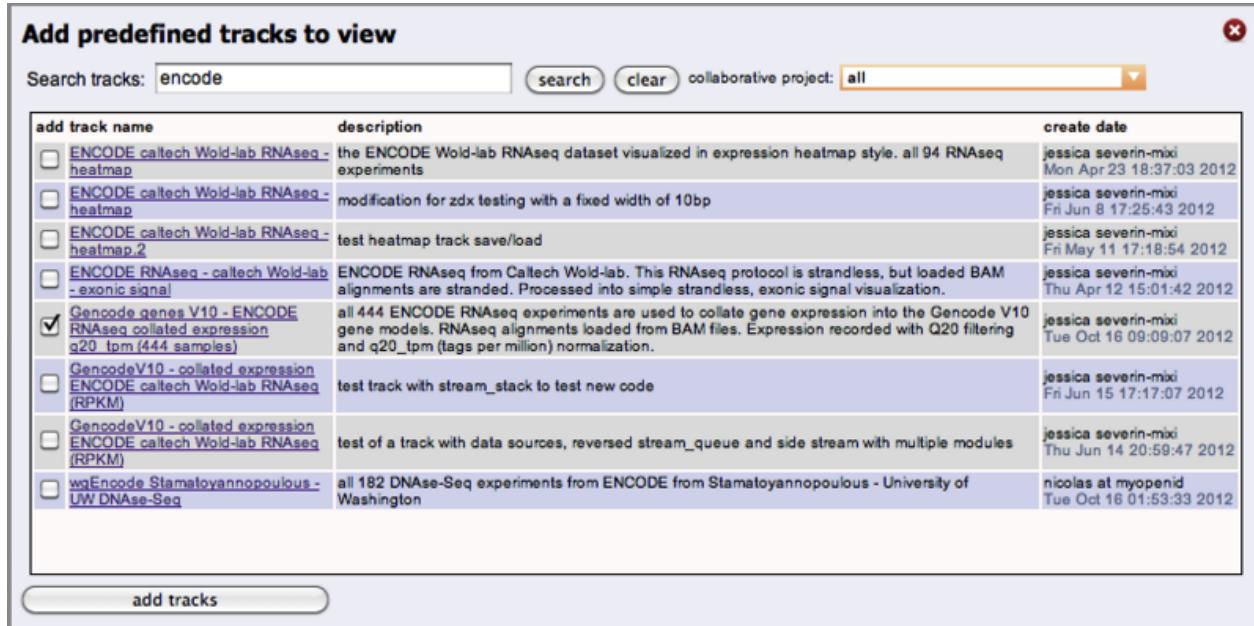
configure new track: glyphTrack111

Data sources		<input checked="" type="checkbox"/> restrict search to current species/assembly	data source type:	all data sources	collaborative project:	all		
Search data sources:		wold maseq lsplice			search	clear		
47 data sources selected		select all						
source name	description					source type		
wgEncodeCaltechRnaSeqGm12878R1x75dAlignsRep1V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Single 75 nt directed reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment		
wgEncodeCaltechRnaSeqGm12878R1x75dAlignsRep2V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Single 75 nt directed reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment		
wgEncodeCaltechRnaSeqGm12878R2x75Il200AlignsRep1V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment		
wgEncodeCaltechRnaSeqGm12878R2x75Il200AlignsRep2V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment		
	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment		
feature mode: skip_metadata expression datatype: tagcount								
Stream Processing script select processing mode: predefined script name: Strandless Experiment heatmap.10 description: This script performs the gridded genomic histogram binning for strandless expression data with subfeature structure for use with the experiment-heatmap visualization option. Primarily used with RNASEq, longRNA, shortRNA data. script uuid: e0qj9Dzgl50sIZV0V2xzB							edit script	replace
Visualization title: ENCODE caltech Wold-lab RNASEq - heatmap visualization style: experiment-heatmap color expression hide empty experiments color options fire3							 background color: black	
							cancel	accept config

For details on Creating new tracks, please see the **Configuring Tracks** section of the documentation

Adding Predefined Tracks Into the View

Previous saved and shared tracks can be added into a view of the genome browser through the **add predefined tracks** control  . Tracks are one of the types of configurations which can be saved and shared among users. This interface allows these previously saved and shared tracks to be added into ones view. Clicking the "add predefined tracks" button brings up this control panel

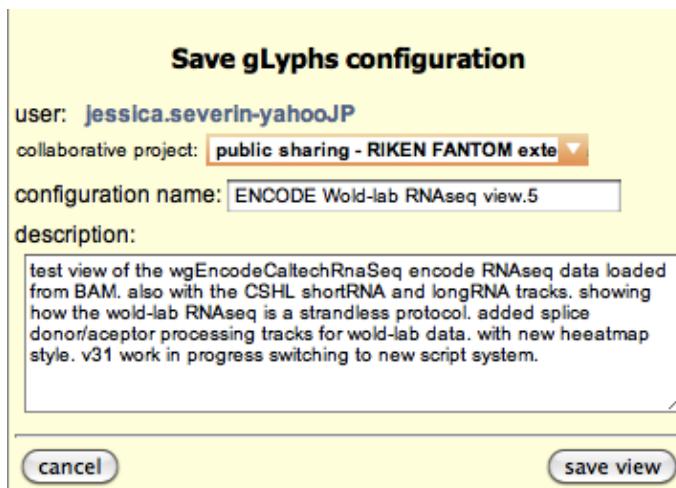


add track name	description	create date
<input type="checkbox"/> ENCODE caltech Wold-lab RNAseq - heatmap	the ENCODE Wold-lab RNAseq dataset visualized in expression heatmap style. all 94 RNAseq experiments	jessica severin-mixi Mon Apr 23 18:37:03 2012
<input type="checkbox"/> ENCODE caltech Wold-lab RNAseq - heatmap	modification for zdx testing with a fixed width of 10bp	jessica severin-mixi Fri Jun 8 17:25:43 2012
<input type="checkbox"/> ENCODE caltech Wold-lab RNAseq - heatmap_2	test heatmap track save/load	jessica severin-mixi Fri May 11 17:18:54 2012
<input type="checkbox"/> ENCODE RNAseq - caltech Wold-lab - exonic signal	ENCODE RNAseq from Caltech Wold-lab. This RNAseq protocol is strandless, but loaded BAM alignments are stranded. Processed into simple strandless, exonic signal visualization.	jessica severin-mixi Thu Apr 12 15:01:42 2012
<input checked="" type="checkbox"/> Genode genes V10 - ENCODE RNAseq collated expression q20_tpm (444 samples)	all 444 ENCODE RNAseq experiments are used to collate gene expression into the Genode V10 gene models. RNAseq alignments loaded from BAM files. Expression recorded with Q20 filtering and q20_tpm (tags per million) normalization.	jessica severin-mixi Tue Oct 16 09:09:07 2012
<input type="checkbox"/> GenodeV10 - collated expression	test track with stream_stack to test new code	jessica severin-mixi Fri Jun 15 17:17:07 2012
<input type="checkbox"/> ENCODE caltech Wold-lab RNAseq (RPKM)	test of a track with data sources, reversed stream_queue and side stream with multiple modules	jessica severin-mixi Thu Jun 14 20:59:47 2012
<input type="checkbox"/> wgEncodeStamatoyannopoulos - UW DNase-Seq	all 182 DNase-Seq experiments from ENCODE from Stamatoyannopoulos - University of Washington	nicoles at myopenid Tue Oct 16 01:53:33 2012

It provides the user with options to search the available tracks, filter based on which collaboration the track was shared with, and then select one-or-more tracks to add to your view

Saving and Sharing Views

Any user who is logged into the system can save and share view configurations with other users. This is done through the save view  button at the top of the genome browser. This will bring up the "save configuration panel"



user: **jessica.severin@yahoo.JP**

collaborative project: **public sharing - RIKEN FANTOM exte**

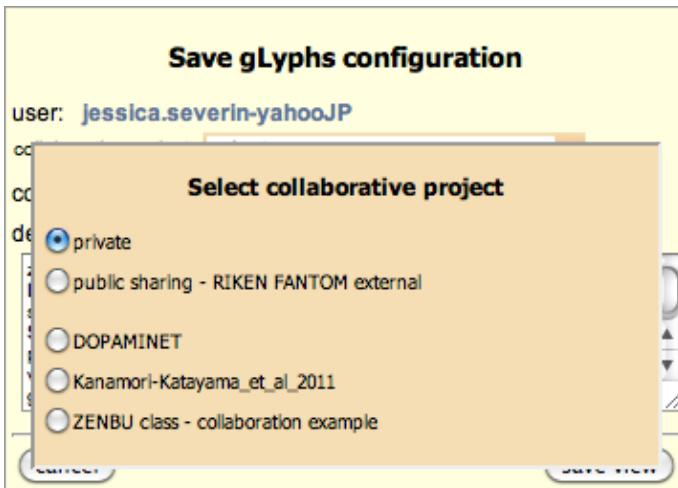
configuration name: **ENCODE Wold-lab RNAseq view.5**

description:

test view of the wgEncodeCaltechRnaSeq encode RNAseq data loaded from BAM. also with the CSHL shortRNA and longRNA tracks. showing how the wold-lab RNAseq is a strandless protocol. added splice donor/acceptor processing tracks for wold-lab data. with new heatmap style. v31 work in progress switching to new script system.

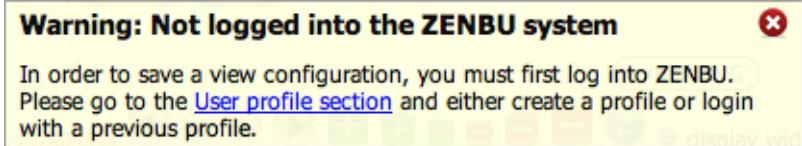
When saving a view, users can enter both a "configuration name" and a description of the view. All text entered will be searchable with the metadata search system at a later time in the data explorer.

The user must also select the user collaboration into which the view will be saved and shared with. Views are only shared into a single collaboration, but can latter be moved to a different collaboration through the editing panel which can be accessed in the data explorer views section



After saving, top of the genome browser view will display the new "configuration name" and description and a new unique URL is generated which allows direct access to this view.

Only users who have logged into the system can save and share views with others. Guests will be reminded to login before trying to save.



Exporting View as SVG Image

ZENBU can export the current view visualization as a publication-grade SVG image. Once you are happy with your view, click the "export svg" button. This will bring up an interface panel which allows for refinement of how the SVG export is generated.

- hide widgets** : will not draw the widgets of the title bar
- hide track sidebars** : will not draw the left side dark gray side bar
- hide title bar** : will not draw the title bar colored background, but will still draw the track title
- hide experiment/expression graph** : will not export the experiment/expression graph part of the display
- save to file** : enable the export into a file
- cancel** : will cancel the SVG export



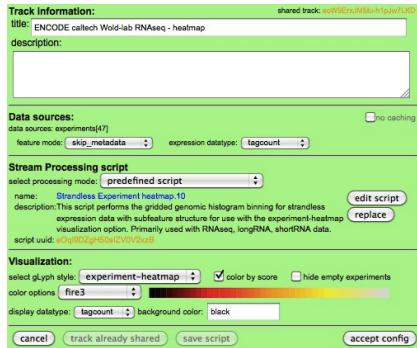
- **export svg:** will send the SVG either into another window of your web browser or to a file.

Controls and widgets

Track control widgets

The widgets and controls on each track which enables the user to modify and move the track.

- **close track** which will delete the track from the view
- **reconfigure track** brings up the **Reconfigure Track** interface which allows users to reconfigure how the track is built.

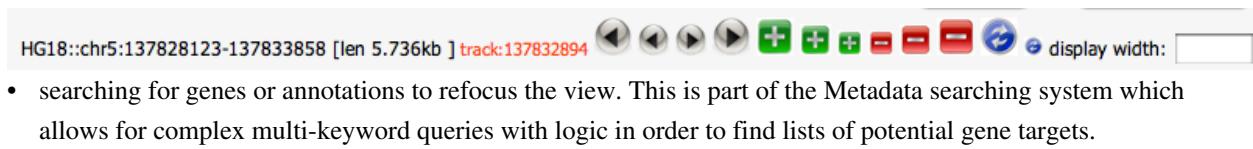


- **copy track** which will duplicate a track in the view. This is very useful when used in combination with track-reconfiguration. It is sometimes easier to take an existing track and modify it to get your desired track-configuration, and this *copy* tool enables this.
- **download data** brings up the Download Track data interface. This enables users to download their processing results into several different file formats for use in external systems or for additional analysis. This panel also provides interfacing for user control of the TrackCache building system
- **activate/deactivate track** changes the active state of a track. means the track is in an active/expanded state and clicking on the blue-circle will cause the track to deactivate and not draw its content. means the track is in a deactivated/compacted state and clicking on the red-bar will cause the track to become active.

navigation controls

There are several ways the user can change the genomic location of the Genome Browser

- navigation buttons to alter location and zoom level



- searching for genes or annotations to refocus the view. This is part of the Metadata searching system which allows for complex multi-keyword queries with logic in order to find lists of potential gene targets.

- direct entry of region location

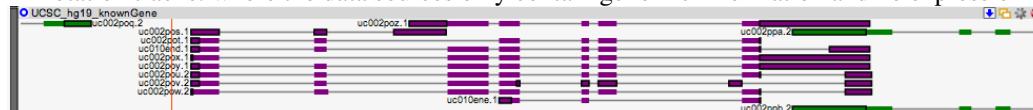
The screenshot shows the top navigation bar with tabs for 'ZENBU', 'about', 'gLyphs genome browser' (which is the active tab), 'data explorer', and 'user'. Below the navigation bar is a search bar containing the genomic location 'HG18::chr5:137828123-137833858'. To the right of the search bar are 'search' and 'clear' buttons.

For details please refer to the **Region Location** section of the documentation.

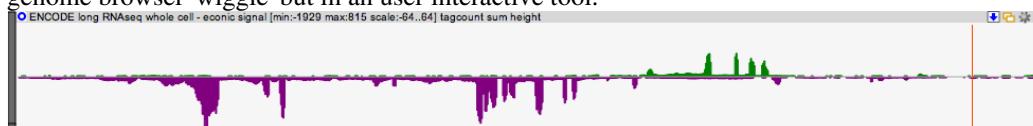
Track visualization styles

The Tracks in the ZENBU gLyphs genome browser fall into three main categories of **visualization** styles

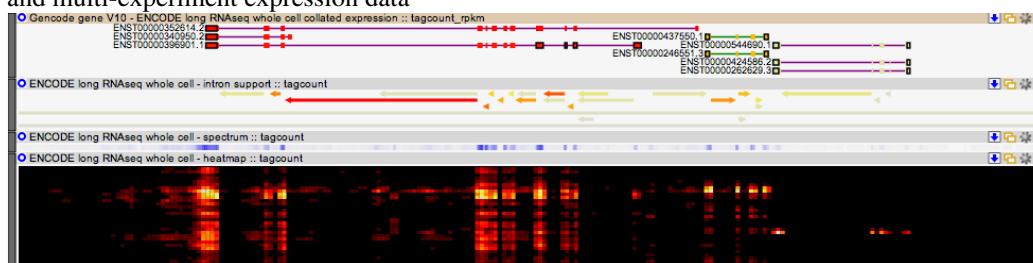
- Annotation tracks: where the data sources only contain genomic information and no expression



- Expression tracks: where expression level is displayed without feature boundaries in a style similar to the UCSC genome browser 'wiggle' but in an user interactive tool.



- Hybrid tracks: ZENBU enhanced visualization which allow for processed data to contain both genomic features and multi-experiment expression data



Annotation Tracks

Annotation tracks are for the visualization of genomic positional data within the ZENBU genome browser.



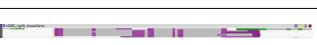
The **visualization style** of visualization can be changed in the **track configuration** interface panel's Visualization section.

Visualization:

visualization style: color expression hide empty experiments

background color:

Different visualization styles vary in the amount of information displayed and the amount of vertical screen space used. For dense data tracks, more compact visualization may be better depending on how one will use the visualization. The strand of the annotation is color coded where green is on the forward(+)strand and purple is on the reverse(-) strand.

	arrow style: a colored arrow is drawn on the 5' end of the annotation pointing toward the 3' end.
	thick arrow style: a colored rectangle with arrow head is drawn from the start to the end of the annotation data. The name of the annotation feature is also displayed. The arrow head is drawn on the 3' end of the annotation data.
	medium arrow style: a colored rectangle with arrow head is drawn from the start to the end of the annotation data. The arrow head is drawn on the 3' end of the annotation data.
	centroid style: like the <i>line</i> visualization but with a colored arrow drawn in the center of the annotation feature.
	transcript style: a complete annotation visualization of the exonic, intronic and UTR regions of a transcript. The UTR regions are displayed as an outline box, exons are displayed as colored boxes, and a gray line spans the entire length of the transcript. The name of the annotation feature is also displayed. If this style is used with annotation data without exon/intron/UTR data, the visualization displays the same as the <i>line</i> style.
	thick transcript style: similar to the <i>transcript</i> style but in a more compact vertical format and without the name of the annotation feature.
	thin transcript style: similar to the <i>thick transcript</i> style but without UTR visualization , and even more compact.
	box style: a colored rectangle is drawn from the start to the end of the annotation feature. The name of the annotation feature is also displayed.
	line style: a gray line is drawn from the start to the end of the annotation feature. The name of the annotation feature is also displayed. No color is employed to indicate strand.
	exon style: a thick colored rectangle is drawn from the start to the end of the annotation feature.
	medium-exon style: a medium sized colored rectangle is drawn from the start to the end of the annotation feature.
	thin-exon style: a thin colored rectangle is drawn from the start to the end of the annotation feature.
	UTR style: an outline box is drawn around the full length of the annotation feature.
	thin style: a compact thin colored rectangle is drawn from the start to the end of the annotation feature. a compact style to use very little vertical screen space.

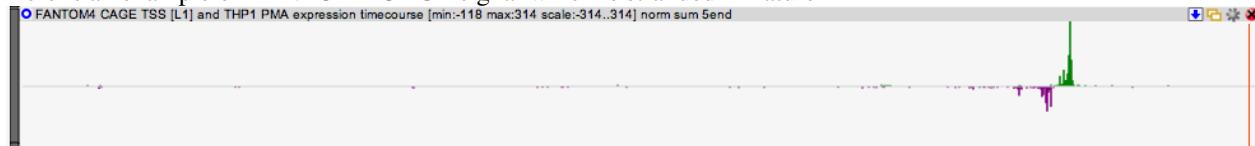
Expression Tracks

Expression tracks are for the visualization of numerical expression data from **Experiment Data Sources** in the ZENBU genome browser on a segmented genomic grid similar to the UCSC genome browser *wiggle* visualization.

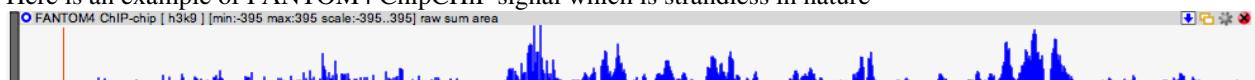
Express signal visualization

In the **express** visualization, expression is visualized as a signal-height graph along genomic coordinate space. ZENBU can visualize both strandless and stranded expression signal.

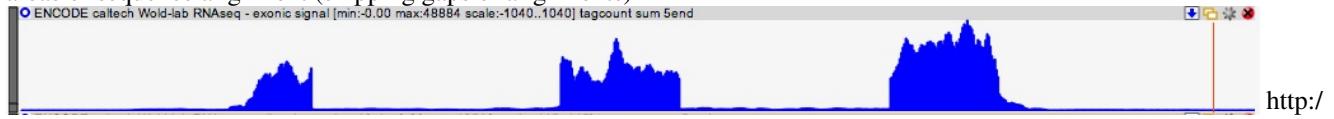
Here is an example of FANTOM4 CAGE signal which is stranded in nature



Here is an example of FANTOM4 ChIPChIP signal which is strandless in nature

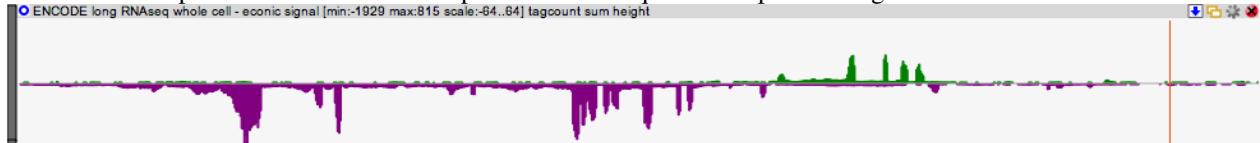


Here is an example of ENCODE strandless-protocol RNAseq signal configured to only display expression signal in areas of sequence alignment (skipping gaps of alignments)



http://fantom.gsc.riken.jp/zenbu/gLyphs/#config=l_D-jGt1IlehEahizVAMeB;loc=hg19::chr8:128746973..128755020

Here is an example of ENCODE stranded-protocol RNAseq exonic expression signal



In order to create this style of visualization the primary expression data must be processed using either the graphical interface **expression binning script GUI** processing modules or with a custom **data processing script** to create the *dynamic genomic segmented grid*.

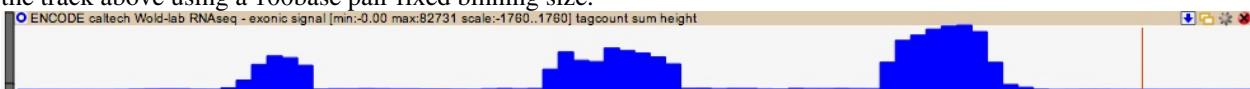
The screenshot displays the Stream Processing script GUI interface. It includes sections for 'Stream Processing script' (processing mode: expression binning script (gui), overlap mode: area under curve, expression binning: sum, process ignoring strand, overlap via subfeatures, fixed bin size), 'Visualization' (visualization style: express, color expression, hide empty experiments, display datatype: tagcount, background color, track pixel height: 50, express scale: auto, log scale, strandless), and other controls like 'process ignoring strand' and 'overlap via subfeatures'.

The expression binning processing GUI parameters are as follows :

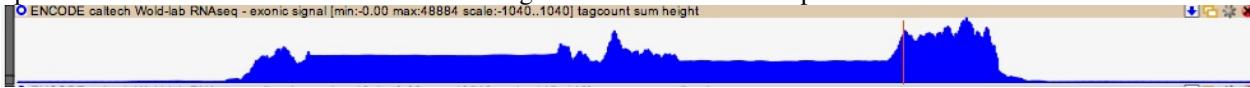
- **overlap mode** : since ZENBU can work directly with sequence alignment data (often uploaded from BAM files) it is necessary to modify the alignments to be properly visualized. The options here are::
 - **area under the curve**: the expression is spread evenly along the length of the alignment so that the area-of-the-curve represents the level of expression.
This only effects alignments which overlap more than one of the genomic segmentation bins. If all alignments are shorter than the genomic segmentation then area and height modes generate the same visualization.
 - **height**: the expression is collated so that the height of curve represents the level of expression at the genomic segment.
This only effects alignments which overlap more than one of the genomic segmentation bins. If all alignments are shorter than the genomic segmentation then area and height modes generate the same visualization.
 - **5'end**: the expression signal is concentrated at the 5'end of the sequence alignment prior to being collated into the genomic segmentation binning.
This is primarily used for CAGE-based sequencing experiments
 - **3'end**: the expression is concentrated on the 3'end of the sequence alignment prior to being collated into the genomic segmentation binning.
Currently there are few RNA sequencing technologies which can utilize this mode of processing but is included for new technology development.
- **expression binning**: the mathematical operation used when multiple expression from the same experiment collate into the same genomic segmentation bin. Each Experiment is kept distinct and this math is applied across different

expression features within the same Experiment. The options are:

- *sum* : sum the different expression values within each experiment
- *min* : calculate the minimum value of the different expression values of each experiment
- *max* : calculate the maximum value of the different expression values of each experiment
- *mean* : calculate the mean average of different expression values of each experiment
- *count* : simply report the count of different expression values within each experiment that collate into the genomic segmentation bin.
- **fixed bin size**: by default the processing script creates dynamic bin sizes based on the zoom level of the genomic view and the width of the display in order that each segmentation bin maps approximately to a single pixel width on the screen. This ensures that a fine enough visualization resolution is preserved without creating unneeded sub-pixel resolution. But if a finer or courser segmentation binning is desire it can be entered here. For example the track above using a 100base pair fixed binning size.

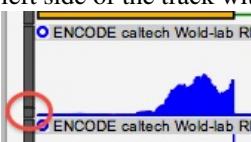


- **process ignoring strand**: if the primary expression experiments are using a strandless protocol or one wishes to process stranded expression in a strandless manner , check this and a strandless genomic segmenation binnning grid will used and strand of the primary data will be ignored. If the data is processed as strandless it is best to also select the strandless option within the visualization options.
- **overlap via subfeatures**: sometimes RNA sequencing experiments generate *gapped* sequence alignments when an RNA molecule spans an intronic splicing junction. This information is contained in BAM files and is preserved durring ZENBU uploading. To get an accurate visualization of true RNA *exonic* signal these intronic gaps should not be collated into the genomic segmentation bins. The example above of the ENCODE Wold lab RNAseq experiments contain such gapped alignments. Here is this BAM sequence alignment data processed without this option enabled and both RNA exon and intron signal is collated into the expression visualization.



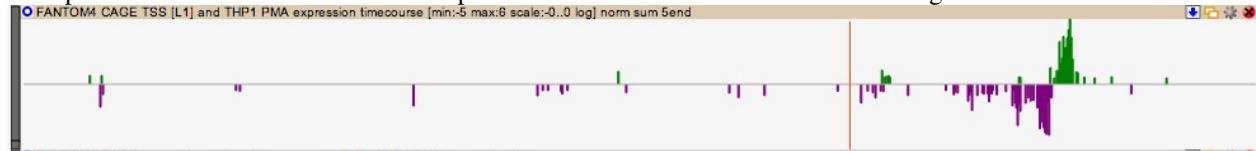
Additional visualization options available for expression Experiment tracks (visualization style of *express*)

- **hide empty experiment**: this parameter effects the track-linked **Experiment Expression panel**. If selected, only those Experiments with a non-zero expression value are displayed.
- **color expression**: currently has no effect when the track is in *express' mode*
- **display datatype**: depending on how the track was configured and processed there may be more than one datatype available for visualization. If more than one is available, please select.
- **background color**: the option of altering the background color to help visually group related tracks in very large views. color can be specified using any of the html web color syntaxes (named colors, #FFFFFF style or rgb(255,255,255) style).
- **track pixel height**: adjusts the screen height of the track. this can also be adjusted with the resize widget on the left side of the track with click-drag.



- **express scale**: adjusts the numerical scale which the expression values are displayed. by default this is *auto* meaning that the expression track is visually rescaled to fit into the height of the track. If one desires to use a fixed scaling among several tracks, this can be set here. Tracks with more expression than this scale limit are *clipped*.
- **log scale**: for visualization the expression can be dynamically compressed onto a log scale. If the expression has huge dynamic range, this can be helpful to expand the *low background* signal and compress the higher peaks. For

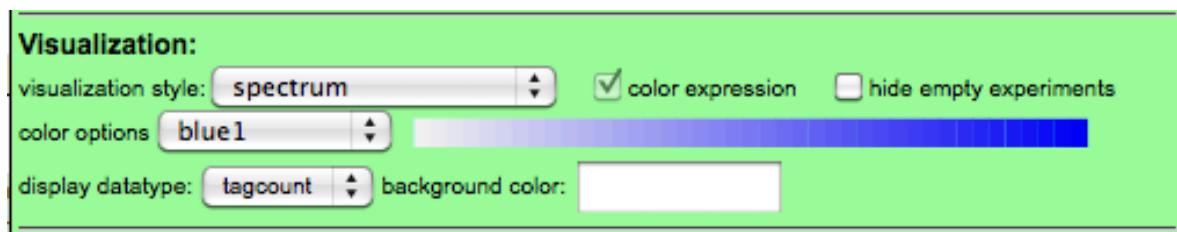
example here is the FANTOM4 CAGE expression track from above visualized on a log scale.



- **strandless:** this visualization option should be set in coordination with data processing which is also strandless.

Expression spectrum visualization

The same expression-binning data processing can be also displayed in the **spectrum** visualization style. This is a visualization style for expression data but can also be applied to non-overlapping hybrid data. It draw the expression on a single layer of a track using only the false-color-spectrum to visualize expression differences. It can be used in combination with normal "expression binning" processing or with more advanced scripts. The spectrum visualization does not display *strand* information so processing should be done in a strandless or *separated-strand* manner using custom scripting.



This example RNAseq data is processed to display only exonic signal (no gaps/introns) and displayed with the "blue1" false-color-spectrum. This style of visualization gives a very compact track which allows people to use it in situations where they might need many separate expression tracks.



Hybrid Tracks

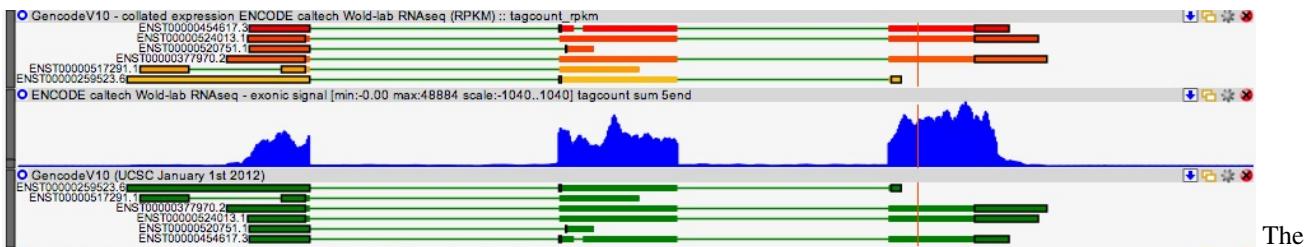
ZENBU advanced visualization tracks which combine genomic annotation and expression Experiments, often in combination with ZENBU data processing to create novel visualizations. There are several different types of visualizations which can be categorized as hybrid tracks.

Expression false coloring of genomic features

In this style of hybrid visualization, genomic annotation Features have expression collated onto them. This can either be generated inside ZENBU by a data processing script or by utilizing the BED file with score-as-expression loading options or with OSCtable files with combined annotation and expression. This visualization is enabled by selecting one of the annotation visualization styles, checking the **color expression** box and selecting a false color spectrum.



For example here is a track which uses ZENBU data processing of the ENCODE wold-lab RNAseq expression (which was loaded via BAM files) collated into Gencode Gene models to give *gene expression*. This data processing is then visualized as a hybrid track with the *transcript* visualization style and the *color expression* option with the *fire1* false-color-spectrum.



Here is a variation on the previous collated-expression situation, but here we use advanced scripting to dynamically generate new genomic-features from the primary data and then use false-color-spectrum to show their abundance.

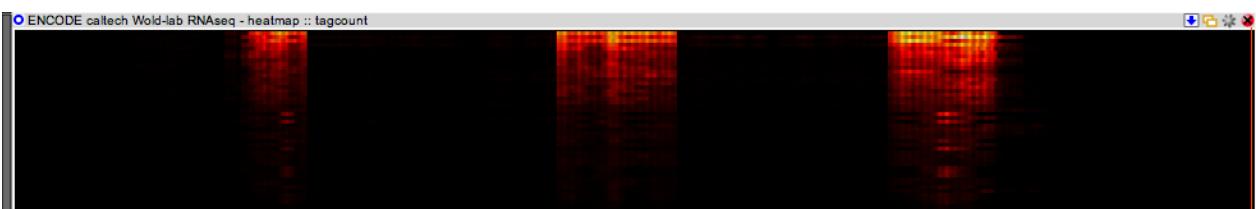


In this track RNAseq alignment gaps are extracted by ZENBU processing into new genomic-features and then "uniqued" and counted. In gapped RNAseq, long gaps mainly occur because of RNA spanning introns and these gaps represent evidence for introns. These "intron evidence" features are then filtered for length and minimum abundance before being displayed using "medium-exon" and a "fire1" spectrum.

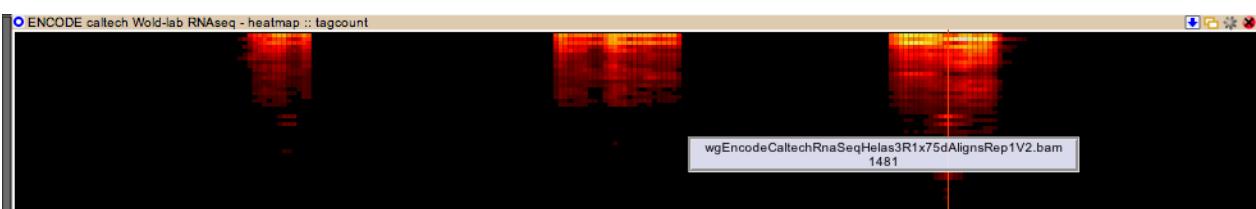
Experiment Heatmap visualization

This is a visualization style for **datasource pooled** tracks with many experiments and expression. In this style of visualization each experiment is given a unique horizontal layer in the image, vertical slices represent genomic segments, and the false-color-spectrum is applied to the expression value at the intersection of genomic-position and experiment. This style of visualization simultaneously shows spatial variation in expression and differential expression between experiments.

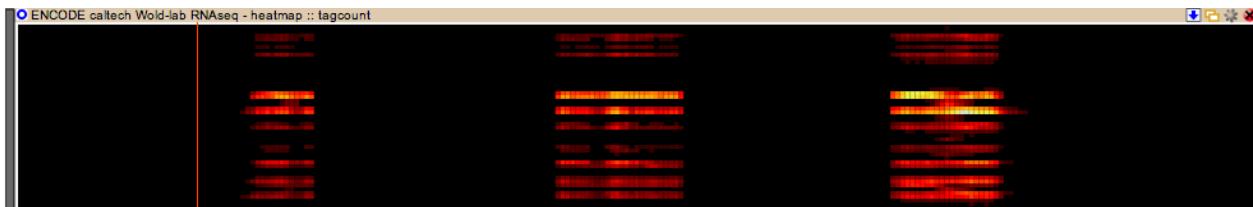
In this example the RNAseq is processed for exonic signal and binned into a genomic-segmentation grid and experiments are sorted based on most expression value.



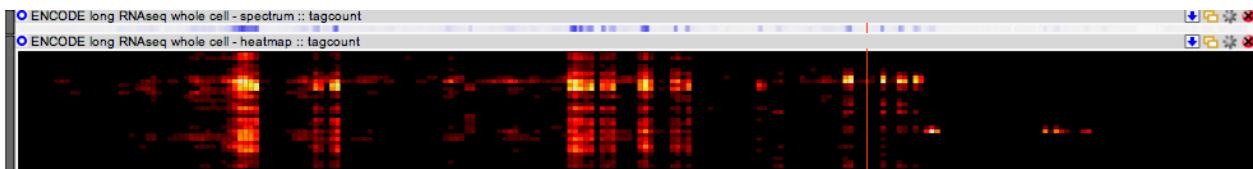
Hovering over elements in the heatmap reveals the name of the experiment, the location and the expression value collated into that genomic segment.



The order of experiments matches the order in the linked Experiment-expression graph and resorting in that panel, changes the sort-order of experiments in the heatmap. Here the sort order is changed to be by sample cell-type name.



Another example showing more dramatic differential expression and spatial difference between RNAseq exonic signal among different ENCODE samples.



Configuring Tracks

New tracks can be created in the ZENBU system either through the **genome browser** control or through the **Data explorer** interfaces.

In all cases the interfaces are similar and are based around the same concept of

1. choosing a collection (one or more) primary data sources to be the dynamically pooled **virtual DataSource** of the track
2. configuring the optional data processing for the track
3. configuring the visualization options

Here is the new track configuration interface panel from the *genome browser* and the reconfigure track interface panel. The main difference is that new-track-configuration panel allows for the configuration of the track data sources.

configure new track: glyphTrack111

Data sources		restrict search to current species/assembly	data source type:	all data sources	collaborative project:	all
Search data sources:		wold rnaseq !splice			search	clear
47 data sources selected		Select all				
source name	description					source type
<input checked="" type="checkbox"/> wqEncodeCaltechRnaSeqGm12878R1x75dAlignsRep1V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Single 75 nt directed reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment
<input checked="" type="checkbox"/> wqEncodeCaltechRnaSeqGm12878R1x75dAlignsRep2V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Single 75 nt directed reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment
<input checked="" type="checkbox"/> wqEncodeCaltechRnaSeqGm12878R2x75lI200AlignsRep1V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment
<input checked="" type="checkbox"/> wqEncodeCaltechRnaSeqGm12878R2x75lI200AlignsRep2V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.					Experiment
feature mode: skip_metadata expression datatype: tagcount						
Stream Processing script						
select processing mode: predefined script						
name: Strandless Experiment heatmap.10						
description: This script performs the gridded genomic histogram binning for strandless expression data with subfeature structure for use with the experiment-heatmap visualization option. Primarily used with RNAseq, longRNA, shortRNA data.						
script uid: e0q9DzgH50sIZV0V2xzB						
Visualization						
title: ENCODE caltech Wold-lab RNAseq - heatmap						
visualization style: experiment-heatmap color expression hide empty experiments						
color options fire3						
background color: black						
cancel accept config						

Track Information:

title: ENCODE caltech Wold-lab RNAseq - heatmap
description:

Data sources:
data sources: experiments[47]
feature mode: skip_metadata expression datatype: tagcount no caching

Stream Processing script
select processing mode: predefined script
name: Strandless Experiment heatmap.10
description: This script performs the gridded genomic histogram binning for strandless expression data with subfeature structure for use with the experiment-heatmap visualization option. Primarily used with RNAseq, longRNA, shortRNA data.
script uuid: e0qj0DZgH50sIZV0V2xzB

Visualization:
select gLyph style: experiment-heatmap color by score hide empty experiments
color options fire3
display datatype: tagcount background color: black

In the **Data Explorer** interface, data sources are selected first prior to launching the *Configure Track* interface panel.

Selecting track data sources

When configuring a track, the first step is to select which data you wish to work with. This can be one or more data sources which forms the track's *virtual* pooled Data Source. Since ZENBU is a dynamic system with many users **uploading** and sharing data, the process of finding data sources is via a *google style* interface for **searching the metadata** of the **Data Sources** in the system.

Data sources restrict search to current species/assembly data source type: all data sources collaborative project: all

Search data sources: wold rnaseq lsplice

47 data sources selected

source name	description	source type
wgEncodeCaltechRnaSeqGm12878R1x75dAlignsRep1V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Single 75 nt directed reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.	Experiment
wgEncodeCaltechRnaSeqGm12878R1x75dAlignsRep2V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Single 75 nt directed reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.	Experiment
wgEncodeCaltechRnaSeqGm12878R2x75l200AlignsRep1V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.	Experiment
wgEncodeCaltechRnaSeqGm12878R2x75l200AlignsRep2V2.bam	lymphoblastoid, International HapMap Project - CEPH/Utah - European Caucasian, Epstein-Barr Virus. Sequencing analysis of RNA expression. Insert length of 200nt. Wold - California Institute of Technology. Whole cell. TopHat v1.0.14. Paired 75 nt reads. Poly(A)+ RNA longer than 200 nt. Shows individual reads mapped to the genome and indicates where bases may mismatch.	Experiment

feature mode: skip_metadata expression datatype: tagcount

To select Data Sources for a new track, simply enter searches, and check the Data Sources you wish to use. There are several options to help with searching including

- **restrict search to current genome/assembly** : unless you are planning an advanced scripted data processing operation, in general only data mapped to the current genome is most useful.
- **data source type** : by default it searches both annotation FeatureSources and expression Experiments. This control allows the user to restrict the search only one of the two types of Data Sources.

- **collaborative project** : selects collaboration based filtering to help narrow the search. Shows only data which has been shared with selected collaboration.
- **refresh** will clear the search panel of all source which have not been **check selected**
- **clear** : will clear the search and unselect all sources so that the user can start over
- **search**: perform the *keyword logic* search

After selecting the Data Source which be pooled into the virtual merged Data Source for the track, the details of the data stream needs to be configured

- **feature mode**: this defines the level of data extracted out of the ZENBU database systems. The more data extracted the *fatter* the features are and the slower the database extraction process. Please select an appropriate level for performance tuning. If uncertain leave select at *full_feature* since this will ensure all data of the Features are available for dataprocessing and visualization. options include
 - *full_feature* : all data associated with the Feature : genomic coordinates, subfeatures, expression and metadata
 - *simple_feature* : only the primary genomic coordinates of the Feature are extracted
 - *subfeature* : primary genomic coordinates and subfeatures. no expression or metadata are extracted for use on the data stream
 - *expression* : primary genomic coordinates and expression. no subfeatures or metadata are extracted for use on the data stream
 - *skip_metadata* : primary genomic coordinates, subfeatures, and expression. no metadata is extracted for use on the data stream
 - *skip_expression* : primary genomic coordinates, subfeatures, and metadata. no expression is extracted for use on the data stream
- **expression datatype**: all expression data is tagged with a **datatype** and within an Experiment there can be multiple different datatypes. This pulldown will display the available datatypes within the pool of selected DataSources. Please select the appropriate datatype for your track.

Configure data processing

The configuration of track data processing falls into 4 categories

- **none** : no additional data processing is needed and the virtual pooled data stream is left unchanged and simply streamed into the visualization.
- **predefined script**: allows the user to select a previously saved and shared data processing script made by another user. This option enables novice users to perform complex data processing with simple *push button* ease.
- **expression binning gui**: this is a simplified user interface which allows users to manipulate their expression data into *expression signal* data. This performs genomic segmentation binning and collation of expression into the genomic bins to get expression signal at a genome level
- **custom XML scripting**: the most advanced option where users write their own data processing scripts using the ZENBU data processing XML language by chaining together modules and datastreams.

Selecting a predefined script

Simply search the existing predefined scripts which have been saved and shared by others users and choose one by clicking on its *script name*. Searching utilizes the metadata searching system to help narrow down the options. The user also has the option to **show all** and scroll through the list of predefined scripts currently in the system.

The screenshot shows a search interface for predefined scripts. At the top, there is a dropdown menu set to "predefined script". Below it is a search bar with the placeholder "Search scripts:" and three buttons: "search", "clear", and "show all". A scrollable list of scripts is displayed in a table format:

script name	description
CAGE_mapQ20 filter	take a stream of reads, filter out on score (aka mapQ for bam/read stream) above 20, contract reads to their 5' end, send into dynamic-scale coverage binning
Entrez gene coding expression_1	two step script which first filters an expression data stream for coding signal by intersecting with known coding exons. Second step collates filtered expression under Entrez gene model boundaries.
Expression Spectrum_1	This script performs the gridded genomic histogram binning for use with the 'spectrum' visualization. This visualization is strandless so expression data must be processed by ignoring the strand data. Primarily used with RNAseq, longRNA, shortRNA data.
FANTOM3 denovo clustering_1	denovo CAGE TSS clustering in FANTOM3 style approach. CTSS level1 -> simple cutoff filter -> 20bp overlap clustering.
GencodeV10 expression	Uses GencodeV10 transcripts as template for expression collation. Only collates expression for primary signal which falls on exons of GencodeV10 transcript models.
GencodeV10 expression - stranded protocols, RPKM normalized	Uses GencodeV10 transcripts as template for expression collation. Only collates expression for primary signal which falls on exons of GencodeV10 transcript models. RPKM normalized for length.
HeatmapRenameEncodeCellLine	Rename ENCODE data with the cell line the data originate from and set a grid binning of 100bp
hg19 sno-mirRNA based filtering	filter the input stream for overlap with sno and miRNA genes as listed by UCSC on Januray 1st 2012
hg19 sno-mirRNA expression level	use the input stream and gather expression levels of sno and miRNA genes as listed by UCSC on Januray 1st 2012

After a script has been selected and loaded into the track, the name and description are displayed along with the unique UUID of the saved script configuration.

The screenshot shows the details of a selected predefined script. At the top, there is a dropdown menu set to "predefined script". Below it, the script details are shown:

- name:** [GencodeV10 expression - stranded protocols, RPKM normalized](#)
- description:** Uses GencodeV10 transcripts as template for expression collation. Only collates expression for primary signal which falls on exons of GencodeV10 transcript models. RPKM normalized for length.
- script uuid:** [DS0dCSUA53opWp52HxyC](#)

On the right side, there are two buttons: "edit script" and "replace".

If the script included some **track_defaults** those options will also be toggled on the panel to their new default-state by the script-loading. If one wants to change the predefined script one can push the **replace** button which brings back the search interface. If one wants to see the details of the script XML or modify it, one can push the **edit script** which will copy the contents of the predefined script into the **custom XML scripting** editing interface.

Expression binning GUI

In order to utilize the Visualization styles of **express** or **spectrum** or **experiment-heatmap** expression data needs to be binned-and-collated into a genomic-segmentation-grid. This data processing interface provides a simplified user-interface to perform this expression-binning.

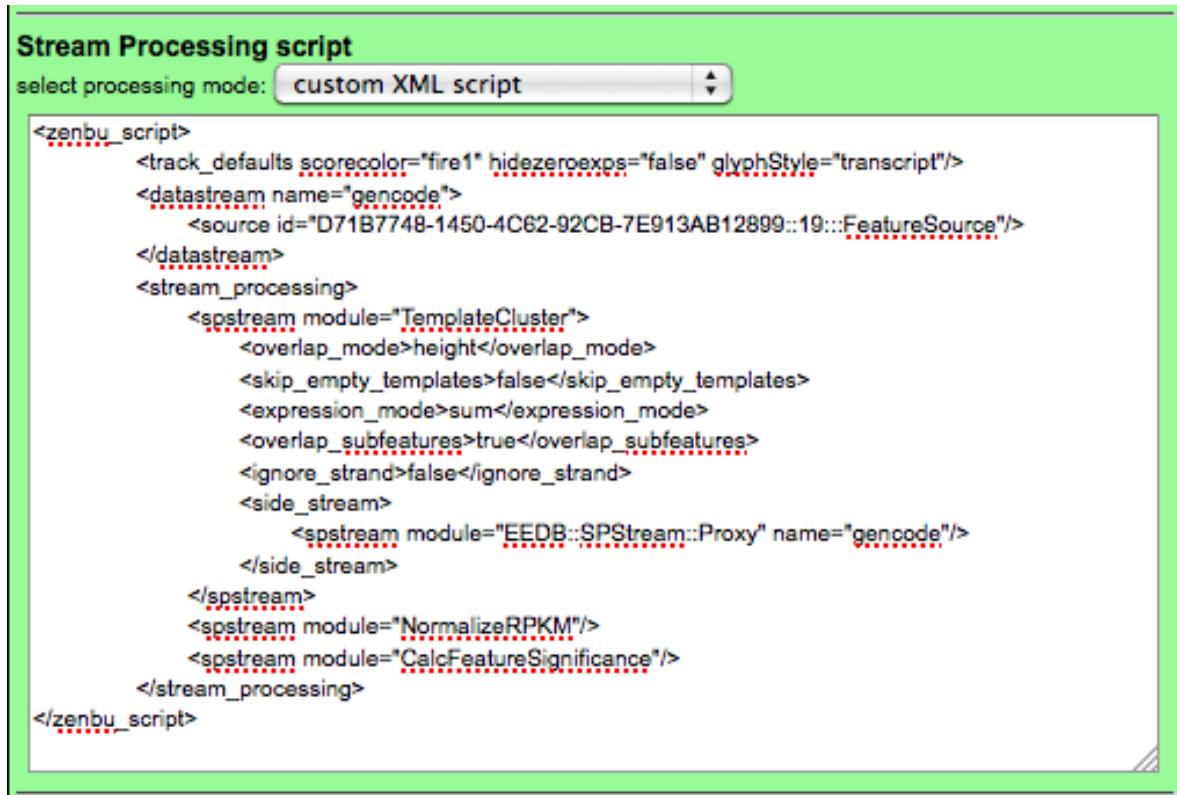
The screenshot shows the expression binning GUI interface. It includes the following settings:

- select processing mode:** expression binning script (gui)
- overlap mode:** 5' end
- expression binning:** sum
- process ignoring strand:**
- overlap via subfeatures:**
- fixed bin size:** [empty input field]

For details on the options in this panel and how they effect expression visualization, please refer to the **Expression Tracks** section of the Track Visualization section of the documentation.

Custom XML scripting

The most advanced option for configuring of data processing in a track is to directly write in the ZENBU script XML language. In this example we show the script XML from the "GencodeV10 expression - stranded protocols, RPKM normalized" predefined script.



```

<zenbu_script>
    <track_defaults scorecolor="fire1" hidezeroexps="false" glyphStyle="transcript"/>
    <datastream name="gencode">
        <source id="D71B7748-1450-4C62-92CB-7E913AB12899::19::FeatureSource"/>
    </datastream>
    <stream_processing>
        <spstream module="TemplateCluster">
            <overlap_mode>height</overlap_mode>
            <skip_empty_templates>false</skip_empty_templates>
            <expression_mode>sum</expression_mode>
            <overlap_subfeatures>true</overlap_subfeatures>
            <ignore_strand>false</ignore_strand>
            <side_stream>
                <spstream module="EEDB::SPStream::Proxy" name="gencode"/>
            </side_stream>
        </spstream>
        <spstream module="NormalizeRPKM"/>
        <spstream module="CalcFeatureSignificance"/>
    </stream_processing>
</zenbu_script>

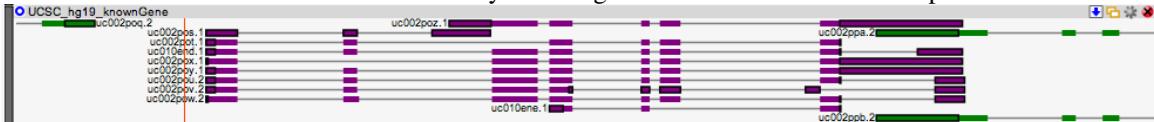
```

For details on how to write ZENBU script XML data processing and all the available processing modules, please refer to the [Data processing](#) user guide section of the documentation.

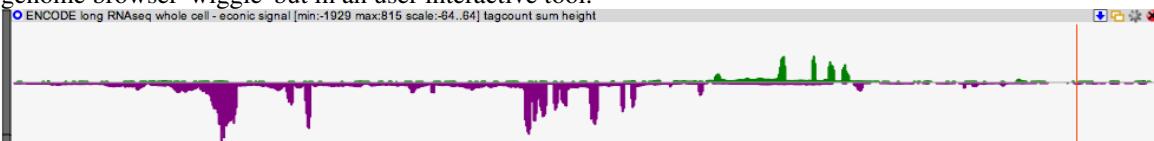
Configure track visualization

The Tracks in the ZENBU gLyphs genome browser fall into three main categories of **visualization** styles

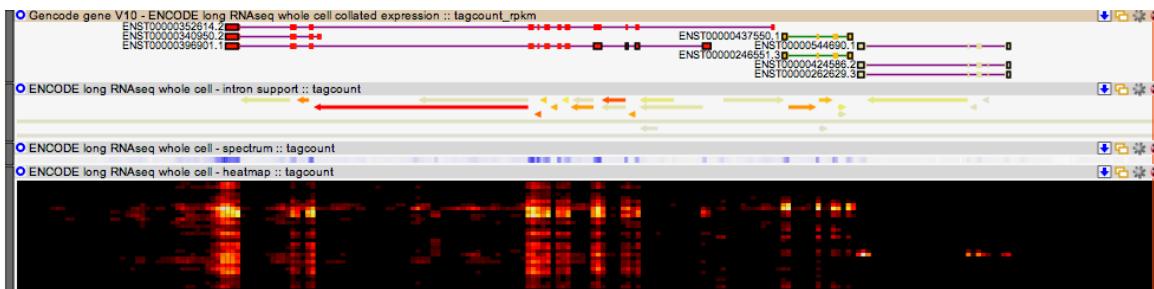
- Annotation tracks: where the data sources only contain genomic information and no expression



- Expression tracks: where expression level is displayed without feature boundaries in a style similar to the UCSC genome browser 'wiggle' but in an user interactive tool.



- Hybrid tracks: ZENBU enhanced visualization which allow for processed data to contain both genomic features and multi-experiment expression data



For details of the different visualization styles and configuration options, please refer to the **Track visualization styles** section of the documentation.

Data Download

Data from any Track in the ZENBU system can downloaded into several different output formats. Downloading is available on any track configuration whether it is a simple single Data Source, a track with a *virtual* pool of data sources or a complex track with processing.

To enable fast downloading for the users, Tracks are built in parallel on a computing cluster utilize the TrackCaching_System. This provides many benefits. If a user has selected a predefined/shared track, the TrackCache will already be prebuilt and download is immediately available. If a user configures a new track, but happens to define it with the same content (same *data pool* and script) as a previous track in the TrackCache, it will use that TrackCache and its already built segments. Another advantage of the TrackCache system, is that the track is built in parallel segments so data download becomes immeadiately available as soon as the request region segment is built, and there is no need to wait for the entire track built.

The interface for track data download is available through the gLyphs genome browser interface and the "track download widget"

Clicking on this widget will bring up the "download track data" panel.

Download Track Data panel

The Track Data download panel provides both an interface to the Data Download system and the TrackCache building system. Data download only becomes available once the genome segments for the track have been built into the TrackCache.

Once the TrackCache data is available, the user can select the file format for download

file formats include: OSCTable, BED12, BED6, BED3, GFF, zenbu xml, and DAS xml. There is the option to "save to file" or to see the result in a web browser window.

The **OSCTable** file option contains several different additional options controls on how to generate the file. The OSCTable is a general purpose tabbed-text table file with additional textual elements to help with loading and parsing. In its most basic form OSCTable is no different from an Excel table with header names. If one chooses OSCTable and do not export the *oscheader metadata* nor the *experiment metadata* then the output file is completely Excel and R friendly. OSCTable is the preferred method for data export since it ensures that all the data of the track is exported including multi-experiment-expression on genomic features. For multi-experiment expression, the different experiments and datatypes are simply appended into additional columns. The genomic-coordinate data for OSCTable follows the BED6 and BED12 formats. So a ZENBU OSCTable export can be considered a BED6/BED12 with additional columns for Feature-metadata or multi-experiment expression which makes it generally usable by many bioinformatics systems and pipelines.

Track building interfaces

Since the data download system relies on the TrackCache system, the Data download panel also serves as the interface to help prioritize track building. The TrackCache building system is automated and runs in the background. As new tracks are created by users they are inserted into the TrackCache system. Anytime a genome browser view tries to access a track for visualization, it will first try to get the data from the TrackCache. If the TrackCache segment is not built, the webservice will log an "anonymous build request" so that this region can be visualized faster next time a user comes to the region. The long ID in gold the is unique hashkey of the track.

Users can track their download requests

When users are logged into the ZENBU system, they can make personal requests to the track building system which will not only help prioritize their tracks of interest, but also is logged into their personal **Downloads** page. Users can check the progress of their Download building from both this interface and through their personal Downloads page.

For example here is a new track which is still not completely built for chr13, and the interface offers the user the option to make a ""build request" which will then be logged into their Downloads page.

Download Track data:
data sources: experiments[444]
region: visible region whole genome
 location: chr13
region build status: 5.555556% (28.8 hours)
72b42caa84536c4298445211c05eac3efc733468caaa30ba8d5b2fc1af19b6
build and email me when download is ready

this will log a request to build this region and you will be notified via email once it has completed. All your download build requests are logged and available in your user profile section. This download-build request will run in the background so you do not need to keep this panel open nor do you need to keep the browser open.

And after the request has been made the user will see this in the track downloads panel.

Download Track data:
data sources: experiments[444]
region: visible region whole genome
 location: chr13
region build status: 3.715270% (47.5 hours) build in progress
72b42caa84536c4298445211c05eac3efc733468caaa30ba8d5b2fc1af19b6
you have already requested to download this region. Please wait for an email, or check the progress % and wait until it has completed building.

If users go to their User "download tab" they will see the status of their previous download/track-build requests. Tracks which are currently building will display a percentage complete. Track-build-requests which have completed, offer a <download> button which will then bring up the "Download Track data" panel

RIKEN
Omics Science Center

ZENBU gLyphs genome browser data explorer user wiki

profile collaborative projects data uploads downloads

id	view	track	location	date	status
4524	view	Gencode genes V10 - ENCODE RNAseq collated expression q20_tpm (444 samples)	hg19 chr18	Tue Oct 16 13:28:54 2012	download ready
4523	view	Gencode genes V10 - ENCODE RNAseq collated expression q20_tpm (444 samples)	hg19 chr17	Tue Oct 16 13:28:48 2012	building 49.5%
4522	view	Gencode genes V10 - ENCODE RNaseq collated expression q20_tpm (444 samples)	hg19 chr8	Tue Oct 16 13:09:20 2012	building 57.03%
4521	view	Gencode genes V10 - ENCODE RNAseq collated expression q20_tpm (444 samples)	whole genome	Tue Oct 16 13:07:56 2012	building 9.42%
2804	view	deepCAGE hippocampus	whole genome	Fri Oct 5 13:54:53 2012	download ready
2803	view	FANTOM3 CAGE in BRAIN tissue	whole genome	Fri Oct 5 13:54:47 2012	download ready
2802	view	FANTOM3 CAGE in LIVER tissue	whole genome	Fri Oct 5 13:54:41 2012	download ready
2801	view	FANTOM3 CAGE in blood and macrophage	whole genome	Fri Oct 5 13:54:35 2012	download ready
2800	view	FANTOM3 CAGE in LUNG tissue	whole genome	Fri Oct 5 13:54:29 2012	download ready
2799	view	FANTOM3 CAGE in all tissues and samples	whole genome	Fri Oct 5 13:54:24 2012	download ready
2792	view	FANTOM3 CAGE in all tissues and samples	whole genome	Fri Oct 5 13:53:33 2012	download ready
2720	view	FANTOM4 CAGE expression collated into promoters[L3] : THP1 PMA expression timecourse	whole genome	Wed Oct 3 09:08:49 2012	download ready
2715	view	FANTOM4 ThP1 PMA timecourse via qRT-PCR	whole genome	Wed Oct 3 08:55:41 2012	download ready
2713	view	FANTOM4 siRNA perturbation expression on Illumina hg6v2 microarray	whole genome	Wed Oct 3 08:55:26 2012	download ready
2712	view	FANTOM4 CAGE TSS [L1] and THP1 PMA expression timecourse	whole genome	Wed Oct 3 08:54:56 2012	download ready
2704	view	FANTOM3 all samples as expression heatmap	whole genome	Tue Oct 2 20:21:07 2012	download ready
2694	view	FANTOM3 CAGE in all tissues and samples	whole genome	Tue Oct 2 20:12:19 2012	download ready
2238	view	deepCAGE hippocampus	whole genome	Fri Sep 28 23:03:23 2012	download ready
2237	view	FANTOM3 CAGE in BRAIN tissue	whole genome	Fri Sep 28 23:03:12 2012	download ready
2236	view	FANTOM3 CAGE in LIVER tissue	whole genome	Fri Sep 28 23:03:04 2012	download ready
26	downloads				

<< previous page | Page: [1](#) [2](#) | next page >>

RIKEN OMICS Science Center (OSC) RIKEN Yokohama Institute ZENBU 2.5.2 Copyright © 2009-2011 RIKEN, Japan. All rights reserved.

Anonymous TrackCache building requests

If users are not logged into the ZENBU system (guests), they can make *anonymous requests* to the trackbuilding system to help prioritize their regions of interest.

Download Track data:

data sources: experiments[444]

region: visible region whole genome

location: chr13

region build status: **5.555556% (19.7 hours)**
72b42caa84536c4296445211c05eac3efc733468caaa30ba8d5b2fc1af19b6

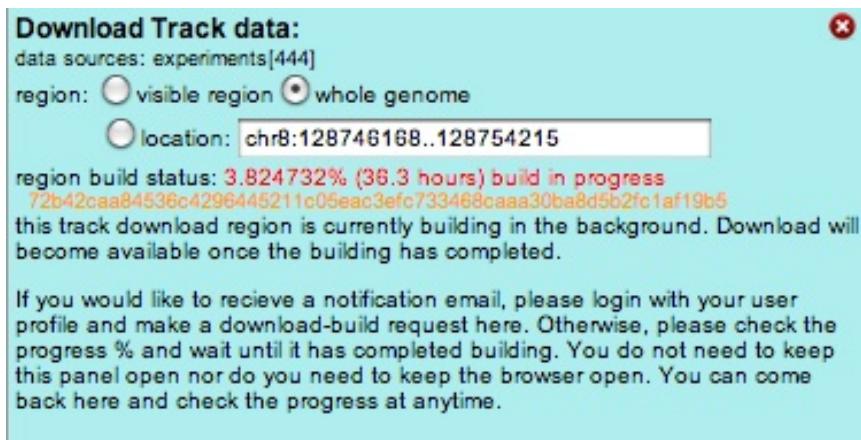
prioritize build

this will send an anonymous request to build this region. The background track building processes will try to prioritize this above other requests. The user will not receive feedback when it is completed.

If you would like to receive a notification email, please login with your user profile.

Please come back here and check the progress % and wait until it has completed building. You do not need to keep this panel open nor do you need to keep the browser open.

If the TrackCache building system already has *workers* building on the region, guests will simply see the following and need to check back periodically if the building has completed



TrackCaching System

Track Caching system

To enable fast and reliable downloading of processed data and to speed up visualization of processed track we implemented a TrackCache system based around a new binary file format called ZDX (Zenbu Data eXchangeformat).

The TrackCache is based on the concept of unique track description. Scripts are parsed such that their sole effective content (that is regardless of their formatting, the unnecessary declaration of parameters with the default value, ...) matters. This means that different people starting from scratch building tracks can generate the same "track description" (similar datasources and processing but different track title, indentation of the script, addition of comments or annotation with the script) and behind the scenes use the same trackcache

Zenbu Data eXchangeformat binary file

To enable fast and reliable downloading of processed data and to speed up visualization of processed track we implemented a TrackCache system based around a new binary file format called ZDX (Zenbu Data eXchangeformat).

The ZDX file is based on the concepts of filesystems with File-allocation-tables and inodes and file-blocks. In ZDX data is stored into znodes, but unlike filesystems where every file-block on the disk is the same size, znodes have a flexible size. The ZDX file header allows for many different subsystems of data to be stored in the same ZDX file.

The main purpose of ZDX is (like a filesystem) to allow not only fast random access, but also to allow augmentation of data to any of the 'files' contained inside it. This is in contrast to other binary file formats in genome science.

The design also allows for the file to be always sorted even when it is partially built.

Features with expression and metadata are stored in the ZDX segments as compressed ZENBU xml using the LZ4 compression algorithm. This provides very fast compress and decompress times with still excellent compression ratios. We do not want to waste too many CPU cycles on compression/decompression. This ensures very fast read/writing of data into the ZDX segments even though it is compressed.

The main sections of how we use ZDX files in our TrackCache is

- a section dedicated to the DataSources of the track
- a pre-segmented genome (TrackCache uses a 100kb non-overlapping segment)
- a sorted Feature/Expression array attached to each segment.

Because the genome is presegmented, it is possible with ZDX to independently build each segment. As a segment is built, the data is written into a znode and appended onto the end of the ZDX file. Since everything is done with inode-like znode pointers, the actual location of the znode in the file is irrelevant. When building a TrackCache segment the TrackCacheBuilder will create a linked-list of znodes where each znode is kept around 200kb. Because the TrackCacheBuilder builds one-segment at a time and the features come out of the ZENBU streaming in sorted order, the writing into the segment is in sorted order. Therefore the ZDX file is always in sorted order, there is never a need to resort the entire file. The ZDX file built with very efficient locking so that 100s of TrackBuilders can be working on creating the same ZDX file at the same time and are only limited by the disk performance of the system.

ZDX binary files inherently enable Map/Reduce parallelization

Because of the virtual file-system like approach in ZDX, we are able to randomly build different parts of the file at the same time.

Since TrackCache ZDX has a presegmented genome, it naturally enables MapReduce style building of the complete genome. And the order of segment building does not matter. The TrackCacheBuilders use the ZENBU API for data streaming and data processing so generate the same result as the webservices.

eHive based system of autonomous-agent and work-claim design

The TrackCacheBuilders follow an autonomous-agent and work-claim design originally developed in the eHive system. TrackCacheBuilders do not need to be told what to do, they check a black-board database (like ehive) for trackcache's which are unbuilt and for user requests for region building. Once they have initialized to a particular TrackCache, they can either build user-requested segments or randomly pick an unbuilt segment. Like eHive the autonomous-agent TrackCacheBuilder workers first lock-and-claim a segment (fast no-race condition) and then proceed to build the segment at whatever pace the dataprocessing allows. This enables 100s-1000s of workers to simultaneously work on the same ZDX TrackCache without colliding (each segment is built only once). This is very efficient and completely autonomous. Because the granularity of genome TrackCache building is on the 100kilobase segment size, the system has very good latency between when a user makes a request for a segment to be built and when a free TrackCacheBuilder worker can finish its current segment and pick up the job request to build another segment.

Just like eHive workers are given a limited lifespan before they "die" and are reborn. This also gives a layer of fault tolerance to the system (like eHive). If worker dies mid-build the segment is labeled as still mid-build with the workers process-ID, it is possible to identify the failure and reset the segment so that another worker can build it. The workers have failure code built-in so many fail-states are caught and the worker can reset the segment before needing to die. In addition all sorts of building stats are recorded for each segment. number of features, build time, worker processID, host machine name... Because of the design of the system it is very easy to have a cluster of computers running 100s of TrackCacheBuilder workers to enable high-degree of parallelization for TrackCacheBuilding. DataDownload is enabled once the requested segments have been built. This means that download can be enabled without requiring the track for the whole genome to have been build. This enhances the user-response experience.

This provides the complete flexibility of the ZENBU data processing and data pooling system.

Because the result of processing is stored in the ZDX TrackCache as ZENBU datamodel XML, it is read back out of the cache fully intact and this able to be reused for further ZENBU data processing and output. This enables the same track cache to download data into many different export formats (bed, gff, osctable). This allows for the web interface to provide much flexibility on data download and still use the same TrackCache/ZDX. And the same TrackCache/ZDX can be used for fast data query for the track visualization system and really enhances the user experience of using the genome browser.

Region Location

changing the location of the genomic region can be done through several means.

direct coordinate entry

coordinates can be directly entered into the **search box** at the top of the genome browser page

The screenshot shows the top navigation bar of the gLyphs genome browser. The tabs are: ZENBU, about, gLyphs genome browser (which is highlighted in blue), data explorer, user. Below the tabs is a search bar containing the text "HG18::chr5:137828123-137833858". To the right of the search bar are two buttons: "search" and "clear".

The format of direct coordinate entry is very flexible. The order of asm:: chrom: start end is arbitrary. Chromosomes are recognized by either starting with chr or ending in a single : suffix. Genome name is recognized by a double :: suffix. The minimum requirement is that a chromosome name (eg chr3:) is specified. Any numbers in the query are assumed to be the start and end, order is not important. If only a single number is found then current genome length is preserved and it is centered on this chromosome position.

Examples of valid direct chromosome locations

- HG19::chr21:34949407-34965611
- chr21:34949407-34965611
- chr21:34949407..34965611
- chr21:34,949,407-34,965,611 [UCSC genome browser style]
- chr21:34949407..34965611 HG19::
- chr21 35965611 34949407 HG19::
- chr13 34231851 35327476
- chr13 3496500
- hg19:: chr13 35000000
- 3496500 chr12

searching for genomic annotation with keywords

The annotation tracks loaded into the current view can be searched for specific features via keyword searching. This can be searching for **entrez genes**, refseq, ensembl genes, mirBASE, gencode, mRNA or any other annotation track which has metadata. Entrez genes in the ZENBU system have been loaded with extensive metadata, so one can even search for terms like *hemo*, *liver*, *amyloid*, or even complex multiple keyword queries. Then simply click the feature you want to move the view to.

The screenshot shows the top navigation bar of the gLyphs genome browser. The tabs are: ZENBU, gLyphs genome browser (highlighted in blue), data explorer, user, wiki. Below the tabs is a search bar containing the text "egr1". To the right of the search bar are two buttons: "search" and "clear".

Entrez gene hg19:: (found 6, from 651) [EGR1](#) [LOC286456](#) [NAB1](#) [NAB2](#) [PTGER4](#) [TOE1](#)

Searching by refseq identifiers will find refseq genes

The screenshot shows the top navigation bar of the gLyphs genome browser. The tabs are: ZENBU, gLyphs genome browser (highlighted in blue), data explorer, user, wiki. Below the tabs is a search bar containing the text "nm_024577". To the right of the search bar are two buttons: "search" and "clear".

UCSC_hg19_refgene:: (found 1, from 35067) [NM_024577](#)

Searching by keywords will search the metadata find related genes in the Entrez Gene track

The screenshot shows the gLyphs genome browser interface. At the top, there are tabs for ZENBU, gLyphs genome browser, data explorer, user, and wiki. Below the tabs is a search bar containing the text "leucine immunoglobulin". To the right of the search bar are "search" and "clear" buttons. A message below the search bar states: "Entrez gene hg19:: (found 21, from 651) ISLR ISLR2 LINGO2 LINGO3 LINGO4 LRFN2 LRFN3 LRFN4 LRFN5 LRIG1 LRIG2 LRIG3 LRIT1 LRIT2 LRIT3 LRRC4B LRRN1 LRRN2 LRRN3 MXRAS TFE3".

navigation with buttons

HG18::chr5:137828123-137833858 [len 5.736kb] track:137832894 display width:

Big arrows move the view left or right retaining about 20% of the visible region. The small arrows move the view retaining about 80% of the visible region.

Plus zooms the view in. Minus zooms the view out.

The large 'recycle' button will reload the region querying data from the server. The small recycle button simply redraws the display using the data currently cached on the user's computer.

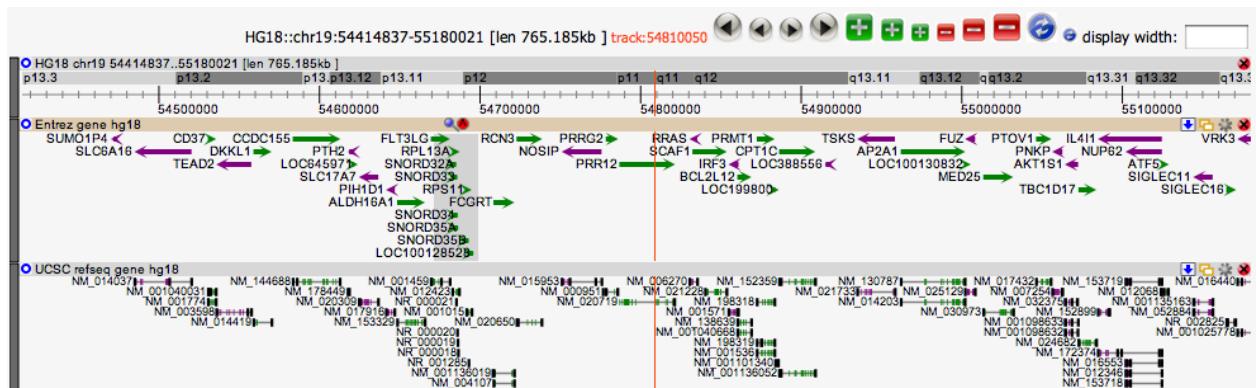
The user can also change the display width of the genome browser to better fit their larger desktop screen monitor or their smaller smartphone or tablet device.

In addition this section of the interface displays the current genomic location in black, and the current position of the **tracking line** in red.

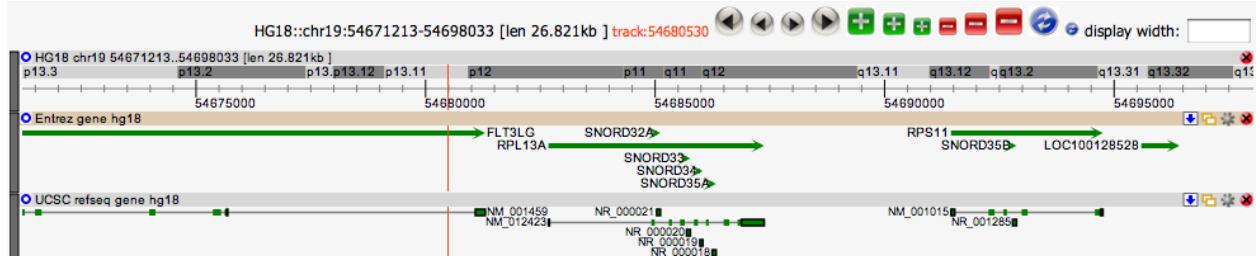
zooming into selection within a track

Any track within the genome browser allows for interactive user selection using a simple click-and-drag motion. After a selection is made one can zoom into that region by clicking the *magnify* button.

For example from this zoomed out region the user has selected a region of several genes

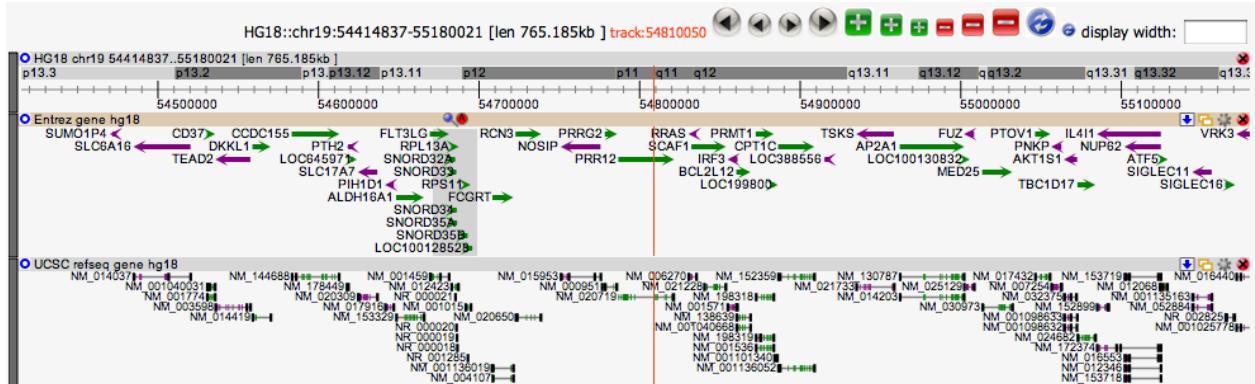


and then after clicking the *magnify* button the view is now zoomed into what was previously the selection



focusing view by selecting an annotation feature

For example from this zoomed out view



the user could *double click* the gene named AP2A1 which would select that feature and then zoom into it.



re-centering the view

If a user *double clicks* in the background of any track, the view will be re-centered on that location, but retains the same zoom level.

Data Explorer interface

Data Explorer Interface

The Data Explorer Interface (or DEX) <http://fantom.gsc.riken.jp/zenbu/dex/> is used for searching the data content of the ZENBU system.

It is divided into 5 tabbed subsections

- View configs
- Track configs
- Experiments datasources
- Annotation sources
- Processing scripts

The screenshot shows the RIKEN Omics Science Center Data Explorer interface. At the top, there is a navigation bar with tabs for ZENBU, gLyphs genome browser, data explorer (which is selected), user, and wiki. On the right side of the header, there are links for guest | Sign in, 0 tracks, 0 experiments, 0 annotation sources, visualize, download, and clear. Below the header, there is a search bar, a genome dropdown set to 'all assemblies', and a page navigation section showing 22 filtered results over 2 pages with a page size of 20. The main content area displays a table of 22 rows, each representing a configuration with columns for row, view, genome, name, description, create date, and accessed date. The descriptions provide details about the data source and processing steps for each configuration.

row	view	genome	name	description	create date	accessed
1	view	hg18	FANTOM4 ThP1 human_8	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hg6v2 microarray expression experiments including siRNA knockdowns and THP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	jessica severin-mixd Tue Sep 25 16:49:49 2012	2059
2	view	mm9	FANTOM3 : Mouse mm9_5	Mouse mm9 assembly with FANTOM3 CAGE data, and deepCAGE hippocampus datasets, showing tissue specific promoter difference in the gelsolin GSN gene.	jessica severin-mixd Fri Oct 5 14:11:14 2012	481
3	view	hg19	ENCODE Wold-lab RNAseq view_4	test view of the wgEncodeCaltechRnaSeq encode RNAseq data loaded from BAM, also with the CSHL shortRNA and longRNA tracks. showing how the wold-lab RNAseq is a strandless protocol. added splice donor/acceptor processing tracks for wold-lab data. with new heatmap style. v3.1 work in progress switching to new script system.	jessica severin-mixd Thu Jul 12 14:03:30 2012	181
4	view	hg19	Human hg19 basic annotation	basic annotation view of human hg19 assembly	jessica severin-mixd Tue Apr 24 21:52:02 2012	179
5	view	hg18	Kanamori-Katayama et al., 2011_6	Supplemental materials for Kanamori-Katayama et al., 2011.	jessica severin-mixd Mon May 28 13:52:35 2012	79
6	view	mm9	mouse mm9 basic annotation	basic configuration of common annotation tracks for mouse mm9	jessica severin-mixd Wed May 9 11:51:07 2012	58
7	view	hg19	Cshl ShortRNA with scripting examples_2	test view of the wgEncodeCshlShortRnaSeq encode shortRNA data loaded from BAM. showing de-novo paraclu clustering with different filtering approaches	jessica severin-mixd Tue May 15 19:02:17 2012	37
8	view	hg18	Human hg18 basic annotation	human hg18 assembly with public annotation tracks from UCSC and NCBI Entrez	jessica severin-mixd Wed May 9 11:41:05 2012	34
9	view	hg19	Encode RnaSeq with gencode script example_1	test view of the wgEncodeCaltechRnaSeq encode RNAseq data loaded from BAM showing script which uses Gencode V10 gene models as a proxy datasource on a side stream to collate expression using TemplateCluster followed by NormalizeRPKM	jessica severin-mixd Tue May 15 19:30:42 2012	20
10	view	hg18	FANTOM4 ThP1 human_12	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hg6v2 microarray expression experiments including siRNA knockdowns and THP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	nicoles at myopenid Wed Oct 3 19:23:14 2012	16
11	view	hg19	Encode RNAsq ExperimentRename script example	test view of the wgEncodeCaltechRnaSeq encode RNAseq data loaded from BAM showing script which renames experiments based on experiment metadata (enc:cell enc:cell_lineage enc:cell_tissue)	jessica severin-mixd Tue May 22 20:41:10 2012	11
			Encode RnaSeq			

Browsing available Views, Track, Annotations and Experiments

DEX and Metadata searching

Common to all View configs, Track configs, Experiments, Annotation sources and scripts subsection of DEX, any section of the data explorer can be searching with the metadata search system. In addition it is possible to filter results based on which specific collaboration the configuration was saved into, or data was shared into.

The screenshot shows the RIKEN Omics Science Center Data Explorer interface. At the top, there is a navigation bar with tabs for ZENBU, gLyphs genome browser, data explorer (selected), user, and wiki. Below the navigation bar, there are tabs for View configs, Track configs, Expression experiments (selected), Annotation data, and Scripts. A search bar is located at the bottom left, along with a search button, a clear button, and a collaborative project dropdown set to 'all'.

One of the primary systems in ZENBU is that of **metadata** and metadata searching.

Search is modeled on the google/yahoo approach of prefix-based multiple-keyword searching. ZENBU also provides additional logic elements to fine tune one's queries.

- **and** : by default space separating keywords in a search is interpreted as an *and* operation. This operates in the same way as *set intersection* [1]
- **or** : is used for combining queries in the same way as *set unioning* [2]
- **not <keyword or ()phrase>** : will exclude any items which match the phrase for the results. For example "not spliced" will return experiments which do not have the keyword *spliced*.
- **!** : short hand for *not*
- **()** : nesting of parenthesis logic are supported.

It is good practice to always give a good description when saving configurations (view/track/script) or uploading data. ZENBU performs automatic keyword extraction from all metadata providing a wealth of ways to search the system.

Here is a complex example searching the Encode hg19 datasets in the Data Explorer experiment interface section.

```
encode hg19 (rnaseq hepg2 !spliced) or (chip-seq)
```

```
http://          /          fantom.          gsc.          riken.          jp/          zenbu/          dex/
#section=Experiments;search=encode%20hg19%20(rnaseq%20hepg2%20!spliced)%20or%20(chip-seq)
```

Other general comments to help users with searching:

- If a search is performed with too many terms, it may fail to return any results. This is the same behavior that google or yahoo has.
- keywords are generally extracted from free-form metadata like *names* and *descriptions* but also controlled metadata like the genome assembly, controlled vocabulary, and ontology metadata.
- for OSCtable files, keywords are extracted from the ParameterValue and ExperimentMetadata sections.
- any metadata added into the system via the Metadata editing system becomes immediately available for searching

Refining searches by Collaboration

The data loaded in zenbu can be shared in the context of collaborations.

By default all views, tracks and primary datasources shared in collaborations the user is a member of are searched.

A dropdown menu located on the right side of the keyword search box allows further refine searches or, in the absence of searched keywords, display the entire content related to a specific collaboration.

View	Genome	Name	Description
1	hg18	FANTOM4_ThP1_human_8	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression experiments including siRNA knockdowns and ThP1-PMA stimulation time course experim...
2	mm9	FANTOM3 : Mouse mm9_5	Mouse mm9 assembly with FANTOM3 CAGE data, and deepCAGE hippocampus datasets, showing tissue specific pr...
3	hg19	ENCODE_Wold-lab_RNAseq_view_4	test view of the wgEncodeCaltechRnaSeq encode RNAseq data loaded from BAM, also with the CSHL shortRNA and wold-lab RNAseq is a strandless protocol, added splice donor/acceptor processing tracks for wold-lab data, with new switching to new script system.
4	hg19	Human hg19_basic_annotation	basic annotation view of human hg19 assembly
5	hg18	Kanamori-Katayama et al., 2011_6	Supplemental materials for Kanamori-Katayama et al., 2011.
6	mm9	mouse mm9_basic_annotation	basic configuration of common annotation tracks for mouse mm9
7	hg19	Cshl ShortRNA with scripting examples_2	test view of the wgEncodeCshlShortRnaSeq encode shortRNA data loaded from BAM, showing de-novo parclu...
8	hg18	Human hg18_basic_annotation	human hg18 assembly with public annotation tracks from UCSC and NCBI Entrez
9	hg18	FANTOM4_ThP1_human_12	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression experiments including siRNA knockdowns and ThP1-PMA stimulation time course experim...

DEX is also the interface with which owner of a View, Track or Script can change the View, Track or Script associated collaboration (for example moving it from "private" and sharing it with a collaboration).

Tab specific refining options

Expression experiment tab

ZENBU can render data obtained from microarray experiments, high-throughput sequencing approaches as different as ChIP-Seq, RNA-Seq, CAGE, ...

Therefore, to facilitate the exploration, we provide for *Expression experiment* datasources a drop-down menu which allows to further refine searches by experimental platforms (i.e. providing a shortcut to filtering the results for experiment associated with the metadata *eedb:platform*, please refer to the section commonly used Experiment metadata for the complete list of metadata).

The screenshot shows the ZENBU interface with the 'Expression experiments' tab selected. A dropdown menu is open under the 'experiment platforms' field, listing various platforms such as Agilent microarray, CAGE, CAGEscan, ChIP-seq, ChIP-seqP, DNA-PET, DNase-Seq, ENCODE, FAIRE-seq, Illumina microarray, Nucleosome, qRT_PCR, Repli-seq, RNA-seq, and user-defined. The menu is filtered to show only relevant platforms based on the search terms.

row	select	platform	experiment name	tissue	time point	treatment	description
1	<input type="checkbox"/>	454CAGE	C6 hippocampus 136		1,000	C6 hippocampus	
2	<input type="checkbox"/>	454CAGE	C6 hippocampus 137		1,000		
3	<input type="checkbox"/>	454CAGE	C6 hippocampus 138		1,000		
4	<input type="checkbox"/>	454CAGE	C6 hippocampus 139		1,000		
5	<input type="checkbox"/>	454CAGE	C6 hippocampus 140		1,000		
6	<input type="checkbox"/>	454CAGE	C6 hippocampus 141		1,000		
7	<input type="checkbox"/>	454CAGE	C6 hippocampus 142		1,000		
8	<input type="checkbox"/>	454CAGE	C6 hippocampus 143		1,000		
9	<input type="checkbox"/>	Agilent microarray	Agl-miRNA RIKEN1_PMA 96hr	P-1	96,000	RIKEN1 PMA 96hr	Agilent miRNA microarray RIKEN1_PMA 96hr
10	<input type="checkbox"/>	Agilent microarray	Agl-miRNA RIKEN3_PMA_0hr	P-1	0,000	RIKEN3 PMA 0hr	Agilent miRNA microarray RIKEN3_PMA 0hr
11	<input type="checkbox"/>	Agilent microarray	Agl-miRNA RIKEN3_PMA 1hr	hg18	TIB-202 THP-1	1,000	RIKEN3 PMA 1hr
12	<input type="checkbox"/>	Agilent microarray	AnLymDINA RIKEN3_DNA 2hr	hg18	TIB-202 THD-1	2,000	RIKEN3 DNA 2hr

The content of this drop-down menu is automatically tailored to only display the subset of platforms relevant to the searched keywords and selected genome assembly.

Annotation data tab

Similarly annotations types can be many, to facilitate browsing those readily loaded and available to you, the *Annotation data* tab search area contains an additional drop-down menu that enables fine-tuning searches for specific **Categories of Annotations**.

Just as for the platform filtering within the *Expression experiment* tab, the content of the drop-down menu is automatically tailored to only display the subset of **Annotation categories** relevant to the searched keywords and selected genome assembly. This annotation category is derived from the metadata *eedb:category*) associated with annotations.

The screenshot shows the ZENBU interface with the 'Annotation data' tab selected. A dropdown menu is open under the 'source categories' field, listing categories such as 3utr, 5utr, exon, mmra, and UCSC_hg18. The menu is filtered to show only relevant categories based on the search terms.

row	select	category	name	genome	count	description
1	<input type="checkbox"/>	3utr	UCSC_hg18	hg18		
2	<input type="checkbox"/>	5utr	UCSC_mmra_mmra_5utr	mm9		

Finding Views thru the View tab

A **View** is a ordered collection of **Tracks** that can be associated with metadata to facilitate its retrieval, shared in the context of a collaboration, linked to, ... In essence, **Views can be considered like interactive figures of a publication**.

Because users can save many views at many locations, there are many views in the system. And since new views can be constantly added to the system, there is a need to have a dynamic, searchable interface to these views. All the user adjustable parameters of the gLyphs genome browser, such as the hidden/shown status of tracks, the width of the displayed glyph, ...) are also preserved into the View config. Views can also override any of the default visualization parameters from Tracks it contains.

This section of the Data Explorer allows users to search for views and to page through the results of those searches. Once the correct view is found, it can then be launched into the gLyphs genome browser by clicking the **view** button.

row	view	genome	name	description	create date	accessed
1	view	hg18	FANTOM4.ThP1.human.8	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hgfv2 microarray expression experiments including siRNA knockdowns and ThP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	Tue Sep 25 16:49:49 2012	2044
2	view	hg18	FANTOM4.ThP1.human.4	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hgfv2 microarray expression experiments including siRNA knockdowns and ThP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	Mon Jun 28 14:59:05 2010	944
3	view	mm9	FANTOM3 : Mouse mm9.2	Mouse mm9 assembly with FANTOM3 CAGE data, and deepCAGE hippocampus datasets, showing tissue specific promoter difference in the gelsolin GSN gene.	Thu May 10 14:03:15 2012	420
4	view	m4	Rat.m4 basic.NChI10010_Wed Sep 07 09 2011 JST 10 59.4	entry point: annotation, NCig10007, NChI10000, NChI1010 clusters, with SNPs and repeatMasker, the gene box is closed by default because it does not auto-update anyway	Tue Sep 13 17:59:19 2011	295
5	view	hg18	dex template.1	basic configuration of common annotation tracks for multiple species (human hg18, mouse mm9, rat m4). Tracks mix across the species, but only species/assembly matching coordinate system returns data.	Tue Jun 22 10:54:44 2010	244
6	view	hg18	Kanamori-Katayama et al., 2011.2	Supplemental materials for Kanamori-Katayama et al., 2011.	jessica severin-miki Mon Apr 25 04:58:22 2011	216

When a View configuration is saved, its Title and Description are searchable in the data explorer via the metadata searching system.

Metadata associated to view can be obtained by clicking on the view name.

row	view	genome	name	description	create date	accessed
1	view	hg18	FANTOM4.ThP1.human.8	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hgfv2 microarray expression experiments including siRNA knockdowns and ThP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	Tue Sep 25 16:49:49 2012	2044
2	view	hg18	FANTOM4.ThP1.human.4	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hgfv2 microarray expression experiments including siRNA knockdowns and ThP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	Mon Jun 28 14:59:05 2010	944
3	view	mm9	FANTOM3 : Mouse mm9.2	Mouse mm9 assembly with FANTOM3 CAGE data, and deepCAGE hippocampus datasets, showing tissue specific promoter difference in the gelsolin GSN gene.	Thu May 10 14:03:15 2012	420
4	view	m4	Rat.m4 basic.NChI10010_Wed Sep 07 09 2011 JST 10 59.4	entry point: annotation, NCig10007, NChI10000, NChI1010 clusters, with SNPs and repeatMasker, the gene box is closed by default because it does not auto-update anyway	Tue Sep 13 17:59:19 2011	295
5	view	hg18	dex template.1	basic configuration of common annotation tracks for multiple species (human hg18, mouse mm9, rat m4). Tracks mix across the species, but only species/assembly matching coordinate system returns data.	Tue Jun 22 10:54:44 2010	244
6	view	hg18	Kanamori-Katayama et al., 2011.2	Supplemental materials for Kanamori-Katayama et al., 2011.	jessica severin-miki Mon Apr 25 04:58:22 2011	216

Creating Views from selected Tracks

Views are ordered collections of Tracks with a *Title* and *Description* and every other user adjustable parameters of the gLyphs genome browser. They can be created by selecting among the list of Tracks already available in ZENBU. Tracks' Title and Description are searchable in the Data Explorer (DEX) via the metadata searching system. DEx allows for the search to be narrowed by collaborations.

Note that the order in which track will be displayed mirrors the order in which tracks have been selected

The screenshot shows the RIKEN Omics Science Center Data Explorer interface. At the top, there's a navigation bar with links for 'glyphs genome browser', 'data explorer', 'user', 'wiki', 'Views' (selected), 'Tracks', 'Expression experiments', 'Annotation data', and 'Scripts'. A user session is shown at the top right: 'nicolas at myopenid | Sign out'. Below the navigation, a search bar and a dropdown menu for 'genome' (set to 'all assemblies') are visible. The main area displays a table of tracks with columns: 'row select', 'name', 'genome', 'description', 'create date', 'data sources', and 'accessed'. Rows 1, 2, 3, 5, 6, 7, 8, 9, 10, and 11 have checkboxes in the 'row select' column, with rows 1, 2, 3, and 11 being checked. Row 11 is highlighted with a red circle. The 'description' column contains details like 'Entrez gene mm9' and 'UCSC hg19_refGene'. The 'create date' column shows dates from 2010 to 2012. The 'data sources' and 'accessed' columns show counts ranging from 4 to 41. To the right of the table, a small preview window shows a genome browser interface with chromosomes and tracks.

Creating Tracks from primary data sources

Tracks are composed of :

- Data sources:** a collection of one or more data sources which are pooled together to form a dynamic merged data source.
- An optional data processing :** an optional *script* of data signal processing modules which manipulates and analyzes the data. This processing can be for either for data visualization purposes or for data exporting and offline analysis.
- Visualization parameters :** a set of default visualization parameters which can be used by visualization systems like our glyphs genome browser when the track is loaded into a View.

Tracks' Title and Description are searchable in the Data Explorer (DEX) via the metadata searching system. Track configurations can be saved and shared in a collaboration or kept private. DEx allows for the search to be narrowed by collaborations.

The ZENBU system allows for the dynamic creation of merged *virtual* Data Sources referred to as " data stream pools". This provides for a great deal of flexibility both in terms of data loading and data processing. With data pooling, there is no need to load new data every time a different "mix" is needed when configuring ZENBU tracks. One can simply use the data already loaded in the ZENBU system and create a new *virtual* DataSource mix.

Experiment and Annotation "shopping-cart selection"

Annotation data tab

The Annotation data tab lists the following useful information about annotations and data available in ZENBU

- row :** row numbering is particularly convenient when paging thru large collection of experiments
- select :** tick box to select/unselect annotation(s) from the current data collection to be pooled in a single track
- category :** the category of the annotation
- name :** the name of the annotation. The name is automatically parsed to allow for searches in DEX or Glyph
- genome :** the genome the annotations / data derive from
- count :** when available the number of features / objects
- description :** a description of the data. Keywords are automatically extracted to allow for searches in DEX or Glyph

row	select	category	name	genome	count	description
1	<input type="checkbox"/>	gene	Entrez_gene_hg19	hg19		
2	<input type="checkbox"/>	gene	UCSC_CDDs_hg19_20120101	hg19		
3	<input type="checkbox"/>	gene	UCSC_Embryo_hg19_20120101	hg19		
4	<input type="checkbox"/>	gene	UCSC_EST_hg19_20120101	hg19		
5	<input type="checkbox"/>	gene	UCSC_gencodeV10_hg19_20120101	hg19		
6	<input type="checkbox"/>	gene	UCSC_mRNA_hg19_20120101	hg19		
7	<input type="checkbox"/>	gene	UCSC_RefSeq_hg19_20120101	hg19		The RefSeq Genes track shows known human protein-coding and non-protein-coding genes taken from the NCBI RNA reference sequences collection (RefSeq). The data underlying this track was retrieved from UCSC on Jan 1st 2012.
8	<input type="checkbox"/>	gene	UCSC_sno_miRNAGene_hg19_20120101	hg19		
9	<input type="checkbox"/>	gene	UCSC_tRNAgene_hg19_20120101	hg19		

Experiment expression tab

The Experiment expression tab lists the following useful information about experiments and quantitative measures available

- **row** : row numbering is particularly convenient when paging thru large collection of experiments
- **select** : tick box to select/unselect annotation(s) from the current data collection to be pooled in a single track
- **platform** : the experimental platform used to provide quantitative measures
- **experiment name** : the name of the experiment. The name is automatically parsed to allow for searches in DEX or Glyph
- **genome** : the genome the experiment / quantitative measures derive from
- **cell line / tissue** : the cell line or tissue the experiment / quantitative measures derive from when available / relevant. Keywords are automatically extracted to allow for searches in DEX or Glyph
- **timepoint** : the timepoint the experiment / quantitative measures derive from when part of a time serie analysis. Stored as "eedb:series_name" and "eedb:series_point" metadata on the Experiment.
- **treatment** : the experimental treatment the cell, cell line or tissue underwent when available / relevant. Keywords are automatically extracted to allow for searches in DEX or Glyph
- **description** : a description of the data. Keywords are automatically extracted to allow for searches in DEX or Glyph

row	select	platform	experiment name	genome	cellline / tissue	time point	treatment	description
1	<input type="checkbox"/>	CAGE	CAGE_RIKEN1_PMA_0hr	hg18	TIB-202 THP-1	0.000	untreated cells	THP-1 control (RIKEN cell 1st) LOT-NO:2
2	<input type="checkbox"/>	CAGE	CAGE_RIKEN1_PMA_1hr	hg18	TIB-202 THP-1	1.000	cultured 1 hour after PMA	THP-1 1 hour after PMA (RIKEN cell 1st) LOT-NO:2
3	<input type="checkbox"/>	CAGE	CAGE_RIKEN1_PMA_4hr	hg18	TIB-202 THP-1	4.000	cultured 4 hours after PMA	THP-1 4 hours after PMA (RIKEN cell 1st) LOT-NO:2
4	<input type="checkbox"/>	CAGE	CAGE_RIKEN1_PMA_12hr	hg18	TIB-202 THP-1	12.000	cultured 12 hours after PMA	THP-1 12 hours after PMA (RIKEN cell 1st) LOT-NO:2

Shopping cart control of primary data pooling

In the upper right corner is "shopping cart" style control panel that display the number of currently selected source being pooled into a track in the making, tracks that have been made and allows to create a view from the collection of tracks.

Views and Tracks can be generated via sequences of

"select">"build", "select">"build", ..., "select">"build" >> "view".

0 tracks 0 experiments 0 annotation sources	0 tracks 0 experiments 2 annotation sources	1 tracks 0 experiments 0 annotation sources
visualize	build tracks	visualize
download	download	download
clear	clear	clear

To create an annotation based track,

1) Simply select to relevant annotation(s).

Until the *built track* button of the shopping cart is pressed annotation are pooled into the current track in the making.

Annotation data

Search: refgene

genome: all assemblies source categories: all categories

12 filtered 60 total feature sources select all << previous page | Page: 1 | next page >> page size: 20

row	select	category	name	genome	count	description
1	<input type="checkbox"/>	3utr	UCSC_hg18_refgene_3utr	hg18		
2	<input type="checkbox"/>	3utr	UCSC_mm9_refgene_3utr	mm9		
3	<input type="checkbox"/>	3utr	UCSC_mm9_refgene_3utr	mm9		
4	<input type="checkbox"/>	5utr	UCSC_hg18_refgene_5utr	hg18		
5	<input type="checkbox"/>	5utr	UCSC_mm9_refgene_5utr	mm9		
6	<input type="checkbox"/>	Subtr	UCSC_mm9_refgene_Subtr	mm9		
7	<input type="checkbox"/>	exon	UCSC_hg18_refgene_exon	hg18		
8	<input type="checkbox"/>	exon	UCSC_mm9_refgene_exon	mm9		
9	<input type="checkbox"/>	exon	UCSC_mm9_refgene_exon	mm9		
10	<input checked="" type="checkbox"/>	mRNA	UCSC_hg18_refgene_mRNA	hg18		
11	<input checked="" type="checkbox"/>	mRNA	UCSC_mm9_refgene_mRNA	mm9		
12	<input type="checkbox"/>	mRNA	UCSC_mm9_refgene_mRNA	mm9		

12 filtered 60 total feature sources select all << previous page | Page: 1 | next page >>

nicolas at myopenid | Sign out

0 tracks 0 experiments 2 annotation sources build tracks download clear

2) "Building Tracks" opens up a panel that allows to specify how the track will be processed and rendered.

Annotation data

Search: refgene

genome: all assemblies

12 filtered 142 total feature sources select all << previous page | Page: 1 | next page >>

configure new track from data sources

Data sources selected sources: 2 annotations edit selected sources

row	select	category	name	genome	count	description
1	<input type="checkbox"/>	centroid	refgene_exon	hg18		
2	<input type="checkbox"/>	thick-arrow	refgene_exon	hg18		
3	<input type="checkbox"/>	transcript	refgene_exon	hg18		
4	<input type="checkbox"/>	thin-transcript	refgene_exon	hg18		
5	<input type="checkbox"/>	cytoband	refgene_exon	hg18		
6	<input type="checkbox"/>	box	refgene_exon	hg18		
7	<input type="checkbox"/>	arrow	refgene_exon	hg18		
8	<input type="checkbox"/>	line	refgene_exon	hg18		
9	<input type="checkbox"/>	exon	refgene_exon	hg18		
10	<input checked="" type="checkbox"/>	utr	refgene_exon	hg18		
11	<input checked="" type="checkbox"/>	thin	refgene_exon	hg18		
12	<input type="checkbox"/>	probesetloc	refgene_exon	hg18		
13	<input type="checkbox"/>	segtree	refgene_exon	hg18		
14	<input type="checkbox"/>	mRNA	UCSC_mm9_refgene_mRNA	mm9		
15	<input type="checkbox"/>	mRNA	UCSC_mm9_refgene_mRNA	mm9		
16	<input type="checkbox"/>	mRNA	UCSC_mm9_refgene_mRNA	mm9		

12 filtered 142 total feature sources select all << previous page | Page: 1 | next page >>

nicolas at myopenid | Sign out

0 tracks 0 experiments 2 annotation sources build tracks download clear

To create an expression based track,

1) Simply select to relevant experiments(s).

Similarly to creating an annotation based track, once the selection of relevant experiments is done, users can finalize the collection of primary data to be pull into the track by simply clicking on "build track" from the shopping cart.

Annotation data

Search:

genome: all assemblies experiment platforms: all platforms

4540 filtered 4540 total experiments select all << previous page | Page: 1 2 3 4 5 6 7 8 9 ... 227 | next page >> page size: 20

row	select	platform	experiment name	genome	cellline / tissue	time point	treatment	description
1	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 136	mm9		1.000	C6 hippocampus	
2	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 137	mm9		1.000		
3	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 138	mm9		1.000		
4	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 139	mm9		1.000		
5	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 140	mm9		1.000		
6	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 141	mm9		1.000		
7	<input checked="" type="checkbox"/>	454CAGE	C6 hippocampus 142	mm9		1.000		
8	<input type="checkbox"/>	454CAGE	C6 hippocampus 143	mm9		1.000		
9	<input type="checkbox"/>	Agilent microarray	AgI-miRNA RIKEN1_PMA 96hr	hg18	TIB-202 THP-1	96.000	Agilent miRNA microarray RIKEN1_PMA 96hr	
10	<input type="checkbox"/>	Agilent microarray	AgI-miRNA RIKEN3_PMA 0hr	hg18	TIB-202 THP-1	0.000	RIKEN3 PMA 0hr	Agilent miRNA microarray RIKEN3_PMA 0hr
11	<input type="checkbox"/>	Agilent microarray	AgI-miRNA RIKEN3_PMA 1hr	hg18	TIB-202 THP-1	1.000	RIKEN3 PMA 1hr	Agilent miRNA microarray RIKEN3_PMA 1hr
12	<input type="checkbox"/>	Agilent microarray	AgI-miRNA RIKEN3_PMA 2hr	hg18	TIB-202 THP-1	2.000	RIKEN3 PMA 2hr	Agilent miRNA microarray RIKEN3_PMA 2hr
13	<input type="checkbox"/>	Agilent microarray	AgI-miRNA RIKEN3_PMA 4hr	hg18	TIB-202 THP-1	4.000	RIKEN3 PMA 4hr	Agilent miRNA microarray RIKEN3_PMA 4hr
14	<input type="checkbox"/>	Agilent microarray	AgI-miRNA DIVINe_PMA 4hr	hg18	TIB-202 THP-1	4.000	DIVINe PMA 4hr	Agilent miRNA microarray DIVINe_PMA 4hr

4540 filtered 4540 total experiments select all << previous page | Page: 1 2 3 4 5 6 7 8 9 ... 227 | next page >> page size: 20

nicolas at myopenid | Sign out

0 tracks 6 experiments 0 annotation sources build tracks download clear

2) "Building Tracks" opens up a panel that allows to specify how the track will be processed and rendered.

Annotation data

Search:

genome: all assemblies

4540 filtered 4540 total experiments select all << previous page | Page: 1 2 3 4 5 6 7 8 9 ... 227 | next page >> page size: 20

configure new track from data sources

Data sources selected sources: 6 experiments edit selected sources

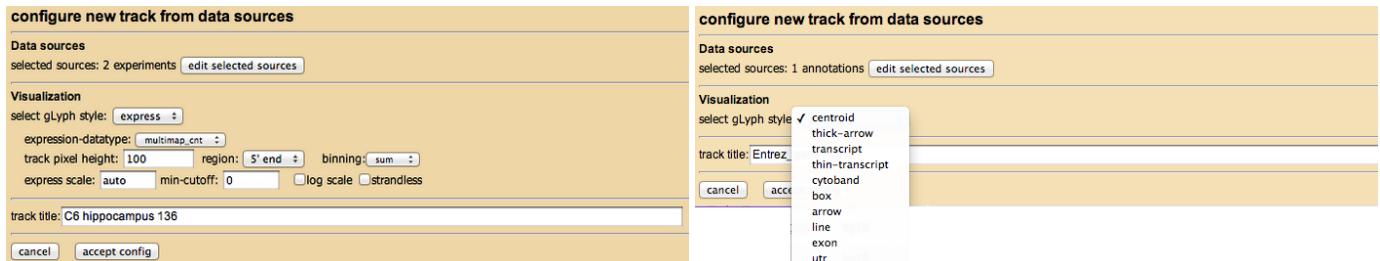
row	select	platform	option
1	<input checked="" type="checkbox"/>	454CAGE	
2	<input checked="" type="checkbox"/>	454CAGE	
3	<input checked="" type="checkbox"/>	454CAGE	
4	<input checked="" type="checkbox"/>	454CAGE	
5	<input checked="" type="checkbox"/>	454CAGE	
6	<input checked="" type="checkbox"/>	454CAGE	
7	<input type="checkbox"/>	454CAGE	
8	<input type="checkbox"/>	454CAGE	
9	<input type="checkbox"/>	Agilent microarray	
10	<input type="checkbox"/>	Agilent microarray	
11	<input type="checkbox"/>	Agilent microarray	
12	<input type="checkbox"/>	Agilent microarray	

nicolas at myopenid | Sign out

0 tracks 6 experiments 0 annotation sources build tracks download clear

Track configuration tuning via gLyph

We have kept the DEX track configuration lean, with expression-based tracks always initially drawn as histogram binned data and annotation-based tracks which can be displayed in a variety of simple visualization styles.

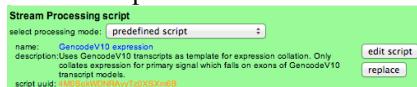


Once the view is brought to gLyph, a rich track reconfiguration interface enables the data stream to be rendered in many ways.

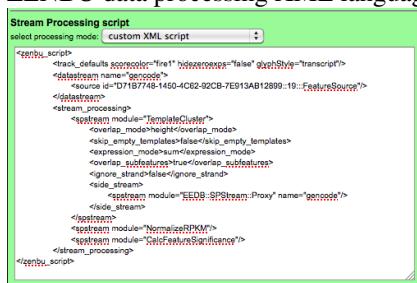
Details can be found in the relevant Track visualization and Configuration sections of the manual. In short, track reconfiguration allows for :

Setting the data processing

- **none:** no additional data processing is needed and the virtual pooled data stream is left unchanged and simply streamed into the visualization.
- **Predefined script:** allows the user to select a previously saved and shared data processing script made by another user. This option enables novice users to perform complex data processing with simple *push button* ease.

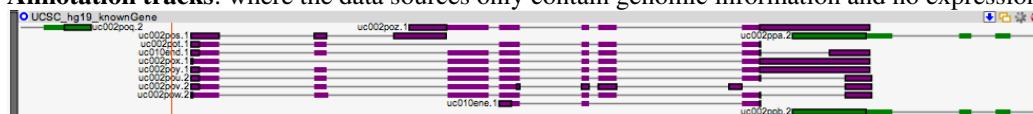


- **Expression binning GUI:** this is a simplified user interface which allows users to manipulate their expression data into *expression signal* data. This performs genomic segmentation binning and collation of expression into the genomic bins to get expression signal at a genome level
- **custom XML scripting:** the most advanced option where users write their own data processing scripts using the ZENBU data processing XML language by chaining together modules and datastreams.



Setting the visualization styles

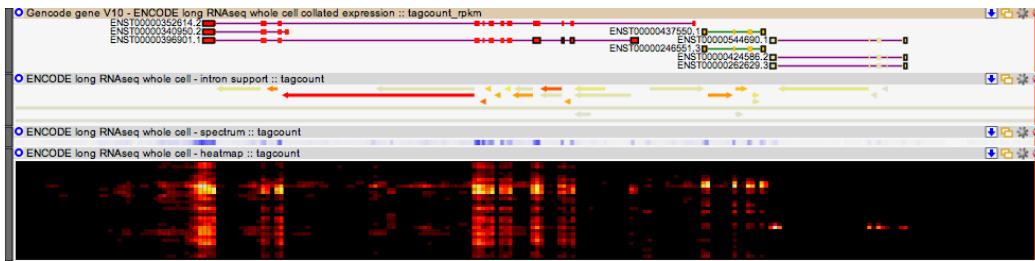
- **Annotation tracks:** where the data sources only contain genomic information and no expression



- **Expression tracks:** where expression level is displayed without feature boundaries in a style similar to the UCSC genome browser 'wiggle' but in an user interactive tool.



- **Hybrid tracks:** ZENBU enhanced visualization which allow for processed data to contain both genomic features and multi-experiment expression data



Browsing available Scripts from DEX

DEX also enable browsing thru available scripts.

Script searching can be performed using the search box and/or refined by collaboration

ZENBU	gLyph genome browser	data explorer	user	wiki
Views	Tracks	Expression experiments	Annotation data	Scripts
Search: <input type="text"/>	<input type="button" value="search"/>	<input type="button" value="clear"/>	collaborative project	curated - RIKEN FANTOM external
28 filtered 35 total scripts	<< previous page	Page: 1 2 next page >>	page size: 20	
row	name	description	create date	accessed
1	Strandless_Exonic_expression_signal	This script performs the gridded genomic histogram binning for strandless expression data with subfeature structure. Primarily used with RNAseq, longLong, shortRNA to only collate expression signal only in non-gapped parts of the alignments. Useful when combined with display styles (express with strands), singltrack with coloring, experiment-heatmap with coloring.	jessica.severn-mxi Thu Apr 12 14:39:00 2012	80
2	Entrez_gene_coding_expression_1	Two step script which first filters an expression data stream for coding signal by intersecting with known coding exons. Second step collates filtered expression under Entrez gene model boundaries.	jessica.severn-mxi Wed Sep 7 17:05:15 2011	35
3	RenameEncodeExperimentWithCellLine	Set the displayed name of experiments with its "enc:cell" and "enc:cell_lineage" associated metadata to facilitate data exploration in gLyph	nicolas at myopenid Wed May 30 20:01:27 2012	24
4	non_coding_filter_1	simple script which uses "refseq" and "all mRNA" exons to defined coding regions and then to mask them out and pass through remaining expression as non-coding signal.	jessica.severn-mxi Mon Dec 9 08:32:46 2010	23
5	ShorRNA_Paraclu_clustering	short RNA reads clustering	nicolas at myopenid Tue Apr 3 17:21:30 2012	18
6	GencodeV10_expression	Works from primary mapping results, merge reads based on 1bp overlap, clusters with Paraclu (cluster length < 200 and expression level > 30 reads) Uses GencodeV10 transcripts as template for expression collation. Only collates expression for primary signal which falls on exons of GencodeV10 transcript models.	nicolas at myopenid Wed Apr 4 20:31:06 2012	15
7	known_coding_filter_2	simple script which uses "refseq" and "all mRNA" exons to defined coding regions and then to only allow overlapping expression signal to pass through.	jessica.severn-mxi Mon Oct 1 17:48:48 2012	15
8	RNAseq_splice_acceptor_2	script which works on RNAseq gapped alignment data (usually loaded from BAM files) to calculate splice acceptor signal at 1bp resolution and visualize.	jessica.severn-mxi Thu Jul 12 13:39:34 2012	7
9	Strandless_Experiment_heatmap_pseudo-log10_rescaled	This script performs the gridded genomic histogram binning for strandless expression data with subfeature structure for use with the experiment-heatmap visualization option. Expression levels are rescaled as pseudo-log10	nicolas at myopenid Sun Oct 14 15:47:35 2012	7
10	Strandless_protocol_RNAseq_collated_onto_gencodeV10_and_RPKM_normalized_pseudo-log2	Strandless protocol RNAseq collated onto gencodeV10 and RPKM normalized pseudo-log2	nicolas at myopenid Sun Oct 14 15:35:51 2012	6

DEX script tab lists for each script:

- A concise title
- A description, detailing the purpose, thought input and output of the script
- The date it was created date and it's owner
- The number of time the script has been sed

Modification of the metadata or collaboration the script belong to can be done via DEX by clicking on the script's name or the edit button.

Script editing can be done via the richer gLyph interface, either from scratch or by modifying an existing script.

Stream Processing script

select processing mode:

```
<zenbu_script>
<track_defaults scorecolor="fire1" hidezeroexps="false" glyphStyle="transcript">
<datastream name="gencode">
<source id="D71B7748-1450-4C62-92CB-7E913AB12899::19::FeatureSource"/>
</datastream>
<stream_processing>
<spstream module="TemplateCluster">
<overlap_mode>height</overlap_mode>
<skip_empty_templates>false</skip_empty_templates>
<expression_mode>sum</expression_mode>
<overlap_subfeatures>true</overlap_subfeatures>
<ignore_strand>false</ignore_strand>
<side_stream>
<spstream module="EEDB::SPStream::Proxy" name="gencode">
</side_stream>
</spstream>
<spstream module="NormalizeRPKM"/>
<spstream module="CalcFeatureSignificance"/>
</stream_processing>
</zenbu_script>
```

Stream Processing script

select processing mode:

name: **GencodeV10_expression**
 description: Uses GencodeV10 transcripts as template for expression collation. Only collates expression for primary signal which falls on exons of GencodeV10 transcript models.
 script uuid: **4M0SokWDNRAvryTz0XSXm6B**

Changing collaborations and editing metadata from DEX

Details about how to edit metadata can be found in the metadata editing section.

In short, metadata editing from dex is available either :

By clicking on the name/title of the View, Track, Annotation, Experiment or Script

- Views' metadata look up example
(in this example, edition is not allowed this we are not its creator)

row	view	genome	name	description	create date	accessed
1	view	hg18	FANTOM4.ThP1.human.8	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hgfv2 microarray expression experiments including siRNA knockdowns and ThP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	Tue Sep 25 16:49:49 2012	2044
2	view	hg18	FANTOM4.ThP1.human.4	Collection of FANTOM4 promoter annotations, CAGE TSS expression signal, ChIP-chip signal, qRT-PCR ThP1 expression timecourse, Illumina hgfv2 microarray expression experiments including siRNA knockdowns and ThP-1 PMA stimulation time course experiments. Mapped onto human hg18 assembly.	Mon Jun 28 14:59:05 2011	944
3	view	mm9	FANTOM3 : Mouse mm9.2	Mouse mm9 assembly with FANTOM3 CAGE data, and deepCAGE hippocampus datasets, showing tissue specific promoter difference in the gelolin GSN gene.	Jessica severin-miki Thu May 10 14:03:15 2012	420
4	view	rn4	Rat rn4 basic.NCh10010_Wed Sep 07_09_2011_JST_10_59.4	Rat rn4 basic.NCh10010_Wed Sep 07_09_2011_JST_10_59.4	Tue Sep 13 17:59:19 2011	295
5	view	hg18	dex template.1	basic configuration of common annotation tracks for multiple species (human hg18, mouse mm9, rat rn4). Tracks mix across the species, but only species/assembly matching coordinate system returns data.	Jessica severin-miki Tue Jun 22 10:54:44 2010	244
6	view	hg18	Kanamori-Katayama et al., 2011.2	Supplemental materials for Kanamori-Katayama et al., 2011.	Jessica severin-miki Mon Apr 25 04:58:22 2011	216

Or

Using the dedicated **edit** button located in the column listing the data ownership & date created/uploaded

Note that only owner of the Views, Tracks, Annotation, Experiments or Script can freely edit its metadata.

- Scripts' metadata editing example
(in this example, being the creator of the script allows to edit its metadata)

row	name	description	create date	accessed
1	1bp-fixed grid histogram binning		nicolas at myopenid edit	2
2	lincma		nicolas at myopenid edit	1
3	unique features		nicolas at myopenid edit	0

DEX is also the interface with which owner of a View, Track or Script can alter the associated collaboration (for example moving it from "private" and sharing it with a collaborartion).

Edit your own copy of a View, Track or Script

It is always possible and very easy, via the gLyph interface, to make a *copy* of any Views, Tracks or Scripts.

This copy being under your name it is then simple to freely edit its metadata as you see fit.

References

- [1] [http://en.wikipedia.org/wiki/Intersection_\(set_theory\)](http://en.wikipedia.org/wiki/Intersection_(set_theory))
- [2] [http://en.wikipedia.org/wiki/Union_\(set_theory\)](http://en.wikipedia.org/wiki/Union_(set_theory))

Metadata searching

One of the primary systems in ZENBU is that of **metadata** and metadata searching.

Search is modeled on the google/yahoo approach of prefix-based multiple-keyword searching. ZENBU also provides additional logic elements to fine tune one's queries.

- **and** : by default space separating keywords in a search is interpreted as an *and* operation. This operates in the same way as *set intersection* [1]
- **or** : is used for combining queries in the same way as *set unioning* [2]
- **not** <keyword or ()phrase> : will exclude any items which match the phrase for the results. For example "not spliced" will return experiments which do not have the keyword *spliced*.
- **!** : short hand for *not*
- **()** : nesting of parenthesis logic are supported.

It is good practice to always give a good description when saving configurations (view/track/script) or uploading data. ZENBU performs automatic keyword extraction from all metadata providing a wealth of ways to search the system.

Here is a complex example searching the Encode hg19 datasets in the Data Explorer experiment interface section.

```
encode hg19 (rnaseq hepg2 !spliced) or (chip-seq)  
http:// / fantom. gsc. riken. jp/ zenbu/ dex/  
#section=Experiments;search=encode%20hg19%20(rnaseq%20hepg2%20!spliced)%20or%20(chip-seq)
```

Other general comments to help users with searching:

- If a search is performed with too many terms, it may fail to return any results. This is the same behavior that google or yahoo has.
- keywords are generally extracted from free-form metadata like *names* and *descriptions* but also controlled metadata like the genome assembly, controlled vocabulary, and ontology metadata.
- for OSCtable files, keywords are extracted from the ParameterValue and ExperimentMetadata sections.
- any metadata added into the system via the Metadata editing system becomes immediately available for searching

Metadata editing

The metadata system is available in nearly every panel of the ZENBU interface. When a user mouses-over an item, the system will give a small info popup panel. If the user then clicks that item, a full metadata panel will appear. This functionality is available in all panels of the Data explorer interface, the User system my data panel, the Experiment Expression panel and the new track configuration building panel.

If the currently logged in user is the creator/owner of the configuration (view/track/script) or the person who uploaded a data source, they are allowed to edit the metadata of that entity.

Here is an example of editing the metadata of a *view configuration* from the Data Explorer interface.

row	view	name	genome	descript
1	view	JMS encode test.4	hg19	test view of the wgEncodeCshlShortRNASeq encode shortRNA data loaded from BAM. nice region. added paraclu clustering and filtering
2	view	JMS encode test.4	eeDB_glyphs_configs	uploaded by: https://id.mixi.jp/28555316
3	view	JMS encode test.4		test view of the wgEncodeCshlShortRNASeq encode shortRNA data loaded from BAM. nice region. added paraclu clustering and filtering
4	view	JMS encode test.4		keyword: encode, jmsx, test.4, testx, addedx, bamx, clusteringx, datax, encodex, filteringx, loadedx, nicex, paraclux, regionx, shorthmax, testx, viewx, wgencodcshlshortrnaseqx eedb:assembly_name: hg19 uuid: ouAFW8Fii5AhSH0VhyRHxB configname: JMS encode test.4 date: Mon Mar 26 19:00:57 2012 Server_Name: fantom46.gsc.riken.jp Http_Host: fantom46.gsc.riken.jp Server_Port: 80x Server_Software: Apache/2.2.3 (Red Hat)x Server_Protocol: HTTP/1.1x CGI_Revision: CGI/1.1x browser: Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.10.229 Version/11.61x REMOTE_ADDR: 134.160.84.66
5	view	JMS encode test.4		add new metadata
6	view	JMS encode test.4		save metadata changes
8	filtered 8 to			xml

- items with a red X can be deleted
- items in blue when clicked on can be changed
- new metadata items can be added
- to cancel your edits, simply close the panel with the button in the upper right of the panel

row	select	category	name	genome count
1	<input type="checkbox"/>	ADNP_IRX_SIX_ZHX.p1	TFBS swissregulon sites altogether_ADNP_IRX_SIX_ZHX.p1	mm9

1 filtered 4219 total feature sources [select](#)

RIKEN OMICS Science Center (OSC) RIKEN Yoichi

TFBS swissregulon sites altogether_ADNP_IRX_SIX_ZHX.p1

created by: <https://nicoles-berlin.myopenid.com/>
sourceID: BC2346CB-A210-49F9-8766-0E727B30E390::FeatureSource

Running all SwissRegulon motifs together with MotEvo against mm9 genome

eedb:assembly_name: mm9
keyword: adnp, againstx, allx, altogetherx, altogether_adnp_irx_six_zhx.p1x, genome, inx, mm9x, motevox, motifx, oscx, ptx, runningx, sitesx, sixx, swissregulon, tfbsx, togetherx, zhx
eedb:name: TFBS swissregulon sites altogether_ADNP_IRX_SIX_ZHX.p1
filetype: oscx
genome_assembly: mm9

[add new metadata](#)

add metadata:

Tag:
Value:

[cancel](#) [add](#)

User Collaborations

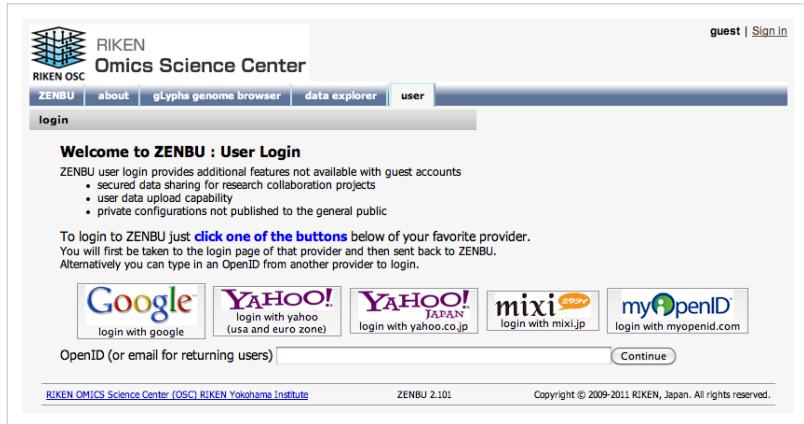
Secured user profiles

ZENBU provide users with data security so that they can safely **upload their data** and **share their data** and **genome browser views** with the specific **collaborations of users** that they trust. Users of the system can also save and share **tracks** and **processing scripts**. It also provides access to their **Download requests** which they made from the Track download interface of the Genome browser

User profiles and secured login is provided through cooperation with highly secured social networking sites like yahoo, google or mixi via the OpenID protocol [2]. This means that users of ZENBU do not need to remember another password since ZENBU stores no passwords.

User profile creation

Profile creation is a simple process of selecting which social networking provider one wants to use for your OpenID profile identity. Users will be directed to that external site for login confirmation if they are not already logged in. ZENBU is never aware of the user's password on that external site. After external login the user will be asked to confirm that it is OK to allow ZENBU to use that OpenID as an external profile. The user is then redirected back to ZENBU and their profile is automatically created. Please complete the process by entering a *nickname* and your *email* address.



logging back into ZENBU

To re-login to the ZENBU system, simply either click the button for the same social networking provider, or enter your email address into the box at the bottom of the login panel. Advanced users may also directly enter their OpenID URL into this box for both profile creation and login.

User system interfaces

After logging into the ZENBU system, users have access to several interface panels to enable them to manage their collaborations and data.

Data sharing collaborations

ZENBU provides for fine level control of who can have access to uploaded data through the concept of a **Collaboration**.

While other genome browsers like UCSC and Ensembl provide for user upload of private data, and they may provide a limited amount of data security, they do not provide any means of sharing that data with a limited group of other scientists in a secured manner. Since Science is a collaboration process involving many people from PIs, to research scientists, to technicians, to bioinformaticians, it makes sense to extend that collaboration process into the genome browser. But of course prior to publication the data and any discoveries in it, must be kept secured.

With the concept of the Collaboration, users of ZENBU can select who they want to have access to which data. In addition a user can create or be a member of more than one collaboration which reflects the how scientists often are working on more than one project at the same time.

Creating Collaborations

All users of the system are allowed to create new collaborative projects. Collaborative projects can be useful not only for sharing data and views with other users in the system, but also for managing your own data and projects like folders on a desktop.

The process of creating a new collaboration is simple. In the User Profile system, make sure the **collaborative projects** sub-menu is selected. At the top of the page one will see a button for creating a new collaboration.

The screenshot shows the RIKEN Omics Science Center ZENBU interface. At the top, there is a navigation bar with the RIKEN logo, 'RIKEN Omics Science Center', and a sign-out link. Below the navigation bar, there is a secondary menu with links for 'profile', 'collaborative projects' (which is highlighted in red), 'data upload', and 'my data'. A button labeled 'create new collaborative project' is visible. The main content area displays a table with columns for 'row', 'name', 'description', 'owner', '# members', 'shared dbs', and 'status'. There is one row of data in the table.

row	name	description	owner	# members	shared dbs	status

Once pushed, the user is presented with a form to describe the collaboration. In addition the user can select if the new collaboration will be *publicly announced* to all users of the system. Publicly announcing a collaboration simply means that any user of the system can request to join the collaboration. The owner of the collaboration still has the control to decide who can and who can not join the collaboration. Publicly announcing a collaboration does not mean the data will be shared with the public.

RIKEN Omics Science Center

ZENBU gLyphs genome browser data explorer user wiki

profile collaborative projects data upload my data

Welcome to ZENBU : collaboration management

create new collaborative project:

collaboration is publicly announced to all users. Each user request must still be approved by collaboration owner. If not checked, the collaboration name is hidden and collaboration owner must send invitations to users.

Collaboration name:

please enter description of collaboration group

collaboration with XXX and YYY to collect brains samples via micro-dissection for RNA profiling with next generation sequencers

create new collaborative project

[RIKEN OMICS Science Center \(OSC\) RIKEN Yokohama Institute](#) ZENBU 2.3.3 Copyright © 2009-2011 RIKEN, Japan. All rights reserved.

After creation the new collaboration will appear in the user's list of available collaborations. They can now share data into this collaboration, save view/track/script configurations into this collaboration, and invite other users to join.

RIKEN Omics Science Center

ZENBU gLyphs genome browser data explorer user wiki

profile collaborative projects data upload my data

create new collaborative project

row	name	description	owner	# members	shared dbs	status
1	public sharing - RIKEN FANTOM external	collaboration for sharing public data and public views/tracks/scripts on the external FANTOM ZENBU server.	public	-	3	member
2	YYY brain micro-dissection RNA profiling	collaboration with XXX and YYY to collect brain samples via micro-dissection for RNA profiling with next generation sequencers.	jessica.severin-myopenid https://jessica-severin.myopenid.com/	1	0	owner - private invite users
3	DOPAMINET	Molecular Networks of Dopaminergic Neurons in Chordates This research involves an in-depth study of dopaminergic neurons which are implicated in Parkinson's Disease (PD). See http://www.dopaminet.eu/ for more information..	https://launchpad.net/~plessy	-	0	request to join ignore
4	Kanamori-Katayama_et_al_2011	Supplemental Materials for Kanamori-Katayama et al. 2011.	https://nicolas-berlin.myopenid.com/	-	0	request to join ignore
5	ZENBU class - collaboration example	This is an example collaboration for use in the ZENBU class to teach the process of user login, collaboration and data sharing. All data in the collaboration will be previously published datasets..	https://me.yahoo.co.jp/a/dRWWhfcULI6IKHtDEhPk2Wt22vU5p1DzTA-#fe4fe	-	0	request to join ignore

5 filtered 5 total system collaborations [**<< previous page**](#) | Page: **1** | [**next page >>**](#)

[RIKEN OMICS Science Center \(OSC\) RIKEN Yokohama Institute](#) ZENBU 2.3.3 Copyright © 2009-2011 RIKEN, Japan. All rights reserved.

Joining Collaborations / Inviting Users

Depending on if the collaboration was **publicly announced** or not the process of adding users to a collaboration is different.

For publicly announced collaborations, all users of the system will see the collaboration in their list of *collaborative projects* but they are not members which means they can not see the data in the collaboration nor can they share things with the collaboration. To join a **publicly announced** collaboration the user clicks the **request to join** button. The owner of the collaboration will then see this new user in the list of *pending requests* for the collaboration and then decide if they wish to accept or reject that users. The user also has the option of *ignoring* the collaboration so that it no longer appears in their collaborative projects list.

For **private** collaborations, the owner of the collaboration must explicitly invite users to join the collaboration. This is done by entering the user OpenID (not their email). If a correct OpenID has been entered, the invited user will appear in a list of outstanding invitations. The user who was invited, when they visit their collaborative projects page, will see the invitation and then decide if they will accept or decline it.

Overall the process is similar to other social networking sites like the facebook 'friend request' system.

Sharing data

After data has been uploaded it can be shared with specific collaboration projects. This provides a secured means for only members of that collaboration to see the data. By focusing data sharing at a Collaboration level, it makes it easier for many people to work together on a particular set of pre-publication data.

Sharing data is provided through the **my data** interface section. Prior to sharing data, the user must upload it into the system. After uploading the new data sources will appear in the **my data** table and become available for *sharing*.

The screenshot shows the RIKEN Omics Science Center website interface. The top navigation bar includes links for ZENBU, gLyphs genome browser, data explorer, user, wiki, profile, collaborative projects, data upload, and my data. The 'my data' tab is currently selected. Below the navigation is a table listing uploaded data sources:

row	dataset name	datasources	genome	description	upload date	share	delete
1	hg19 UCSC gencode V11	1 annotation sources	hg19	hg19 UCSC gencode V11	Wed May 16 15:37:21 2012	share	delete
1 uploaded data sources <a href"=""><< previous page Page: 1 next page >>							

At the bottom of the page, there are links for RIKEN OMICS Science Center (OSC) RIKEN Yokohama Institute, ZENBU 2.3.3, and a copyright notice: Copyright © 2009-2011 RIKEN, Japan. All rights reserved.

Select the data source you want to share and click the **share** button. This will then popup a panel where the user can select which collaboration(s) they wish to share the data with.

The screenshot shows the same RIKEN Omics Science Center website interface as above, but with a modal dialog box overlaid on the 'my data' table. The dialog title is "Share uploaded database : hg19 UCSC gencode V11". It contains a table with two rows of sharing information:

row name	description	# members	action
1 public sharing - RIKEN FANTOM external	collaboration for sharing public data and public views/tracks/scripts on the external FANTOM ZENBU server.	-	share
2 YYY brain micro-dissection RNA profiling	collaboration with XXX and YYY to collect brain samples via micro-dissection for RNA profiling with next generation sequencers.	1	share

The dialog has a close button (X) in the top right corner. At the bottom right of the dialog, there is a link: RIKEN, Japan. All rights reserved.

After a data source has been shared with a collaboration, the panel will show the sharing

Share uploaded database : hg19 UCSC gencode V11			
row name	description	# members	action
1	public sharing - RIKEN FANTOM external	-	sharing
2	YYY brain micro-dissection RNA profiling	1	share

Public sharing

In addition to all user created collaborations, the ZENBU system provides a special collaboration called **public sharing** which all users are automatically members. This collaboration is used for sharing configurations or data with all users of the system, thereby making the data **public** to the world. The *public sharing* collaboration is useful for

- creating new views/tracks/scripts based on already published data
- loading publicly available data from other systems and databases
- sharing your newly published data with the world on the ZENBU system.

Data Uploading

Data loading

ZENBU supports several files types for uploading primary data into the system. Since ZENBU provides built in **data processing** capabilities, it is possible to upload data in a more raw or primary format. When data is loaded into the system it is first translated into the internal ZENBU **Data Model** which allows the ZENBU system to manipulate that data as genomic annotation, expression data, and descriptive metadata.

File formats

Since ZENBU can process data internally to create its visualization, it does not need to support many *visualization file formats*, but instead can focus on just a few **data interchange** file formats which are commonly used for bioinformatics analysis. The leading tools in bioinformatics these days are bedtools (BED files) and samtools (BAM/SAM files) which makes BED and BAM the most important *data interchange files*. It is a benefit of ZENBU that we only have a few file formats. It means that the bioinformatics pipelines that would feed into ZENBU only need to worry about a handful of already common file formats.

The file types currently supported by ZENBU upload are:

- **BAM & SAM** sequence alignment files. These are the primary data files produced by sequence alignment and are the starting point for next generation sequencing (DNA/RNA) based bioinformatics. ZENBU can work directly with these files to create many different tracks. Since all information is available via BAM/SAM this is the recommended format for loading your RNA/DNA sequencing data into ZENBU.
- **BED** annotation files. This is a general purpose genome annotation format which has become very commonly used by bioinformaticians for genome coordinate data interchange. ZENBU can also interpret the **BED score** as an expression value.
- **GFF GFF2 GTF** annotation files. This is another common genome annotation file format primarily used by Ensembl and GBrowse.
- **OSCTable**. This is highly flexible tabbed-text table format which is compatible with Excel, R and any program which can parse tabbed tables. OSCTable includes *controlled vocabulary* for column names and *metadata* to allow ZENBU to automatically parse these files into the internal data model. Even most custom bioinformatics analysis table output can be *wrapped* with an OSCTable header which can allow it to be loaded into ZENBU.
- **Other tab delimited formats** such as *BED+n fields* (ENCODE BroadPeaks : BED6+3, ENCODE NarrowPeaks : BED6+4) can easily be uploaded as **OSCTable** with a custom header to take advantage of the those additional *n* fields

The ZENBU track data download system can export data in these file formats

- **BED** annotation files.
- **GFF GFF2 GTF** annotation files
- **OSCTable**.
- **ZENBU XML**. This is the native ZENBU XML data interchange format which contains the full data model content.
- **DAS XML**. The XML interchange format used by the DAS system <http://www.biodas.org/>

Secured data uploading

ZENBU provides for data loading through the secured user profile system.

This guarantees that the data is only available to the specific users who should have access to it.

After a user has securely logged into the ZENBU system they can upload data for either private use or for sharing with specific collaborations.

Uploading of data with associated experiment/expression

UCSC genome browser or the IGV genome browser tie the data upload format to its visualization. For example in UCSC, BED^[1] files are always displayed as annotation and wig^[2] files are always displayed as "wiggle" tracks. With UCSC or IGV, all processing must be performed externally to the system prior to creating their *visualization files*.

In contrast, ZENBU offers greater flexibility : typical annotations containing files (ESTs, gene models, ...) in BED format can be used to produce wiggle tracks or heatmaps, bam files can be displayed as annotations (so as to see individual reads), etc...

Experiment expression data can be loaded via different means.

- as BED files
 - BED file based data uploading offers the option to use the *score* column and assign its value to a specific expression data type by clicking the **[BED.score column has expression values]** option and selecting the datatype associated to those expression values.
 - If the expression is simply a count of '1' for every feature (for example, used when loading mapped reads), then one can use BED or GFF style files and simply check the **[single-best-mapping expression]** option.
- as OSCtable files
 - ZENBU OSCtable parser is able to parse both tab-separated and space-separated files.
 - OSCtables provide a rich set of control vocabulary to specify multiple experiments within a single file, experiment metadata and multiple datatypes in multiple columns in the file.
 - OSCtable based data uploading allows all possible mapping of data into the internal data model. It is possible for the ZENBU OSCtable parser to have an extended vocabulary of metadata directives and column name spaces. Details can be found in the OSCtable specifications page.

References

- [1] <http://genome.ucsc.edu/FAQ/FAQformat.html#format1>
- [2] <http://genome.ucsc.edu/FAQ/FAQformat.html#format6>

BAM and SAM file support

SAM/BAM files are commonly produced by sequence alignment programs like BWA and TopHat. Most next generation sequencing systems provide BAM alignments as the final output of their systems. ZENBU provides both native BAM/SAM file support, and extended column namespace of the OSCtable format to allow 'wrapping' of SAM files with an OSCtable header.

ZENBU interpretation of BAM alignments

Since BAM is an alignment file format, it is required for ZENBU to interpret these alignments into the ZENBU data model. The ZENBU data model is composed of data sources (FeatureSource, Experiment), genomic location information (Features), expression count data (Expression), and descriptive metadata. The BAM alignments are mapped into the ZENBU system as follows:

- The genomic bounds of the alignment on the genome are mapped to the Feature genome locations.
 - Currently the BAM CIGAR is not interpreted
- Each BAM alignment is given an expression of "1 tagcount". If sequences are aligned to a single location this is an accurate interpretation. If the alignment program aligns to multi-locations, these sequence tags can be over-represented.
- The BAM map quality (MAPQ) is mapped into both the Feature score and into an expression value of type "mapquality"
- A single FeatureSource and Experiment are created for each BAM file
- The BAM header is parsed for descriptive metadata which is transferred into the FeatureSource and Experiment.

The following SAM/BAM header tags are parsed into ZENBU metadata

- ReadGroup ID => sam:id
- ReadGroup CN (SequencingCenter) => sam:center
- ReadGroup SM (sample) => sam:sample
- ReadGroup LB (library) => sam:library
- ReadGroup DS (description) => sam:description
- ReadGroup FO (flow order) => sam:flow_order
- ReadGroup PL (platform/technology) => sam:platform
- ReadGroup PI (predicted median insert size) => sam:predicted_insert_size
- ReadGroup PU (platform unit) => sam:platform_unit
- ReadGroup KS (key sequence) => sam:key_sequence
- ReadGroup DT (date of production) => sam:production_date
- ReadGroup PG (processing program) => sam:program

Future extensions

In the future we hope to extend the ZENBU interpretation of the BAM alignments. These future extension will not require reloading of data since the BAM files are retained in their original form and interpretation is performed on-demand at query time.

SAM as OSCtable header

Since SAM files are tab-separated text files, it is possible to represent them with an OSCtable column header line using the ZENBU extended column namespace and load them directly into the ZENBU system.

```
eedb:name      eedb:sam_flag      eedb:chrom      eedb:start.1base
              eedb:score      eedb:sam_cigar      sam:mrnm      sam:mpos
sam:isize      eedb:seqread      sam:qual      eedb:sam_opt
```

SAM specification

The official SAM/BAM specification is available here <http://samtools.sourceforge.net/SAM1.pdf>

BED file support

BED files are a common interchange format for genomic annotations. BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used.

The 12 BED columns are labeled as follows:

1. *chrom* - The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. *chromStart* - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
3. *chromEnd* - The ending position of the feature in the chromosome or scaffold. The chromEnd base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and span the bases numbered 0-99.
4. *name* - Defines the name of the BED line.
5. *score* - A score between 0 and 1000.
6. *strand* - Defines the strand - either '+' or '-'.
7. *thickStart* - The starting position at which the feature is drawn thickly (for example, the start codon in gene displays).
8. *thickEnd* - The ending position at which the feature is drawn thickly (for example, the stop codon in gene displays).
9. *itemRgb* - An RGB value of the form R,G,B (e.g. 255,0,0). If the track line itemRgb attribute is set to "On", this RGB value will determine the display color of the data contained in this BED line.
10. *blockCount* - The number of blocks (exons) in the BED line.
11. *blockSizes* - A comma-separated list of the block sizes. The number of items in this list should correspond to blockCount.
12. *blockStarts* - A comma-separated list of block starts. All of the blockStart positions should be calculated relative to chromStart. The number of items in this list should correspond to blockCount.

ZENBU interpretation of BED files

The BED file format easily maps into the ZENBU data model.

- *chrom, chromStart, chromEnd, strand* are directly interpreted as genomic coordinates. It should be noted that BED files are in a zero-exclusive coordinate space, while ZENBU uses a 1-based-inclusive coordinate space. ZENBU automatically handles the conversion between coordinate spaces.
- *name* is stored in the ZENBU **Feature** name
- *score* is stored in the **Feature** significance. On data uploading there is an option to copy the score into an **Expression** value of a specified **DataType**.
- the three columns *blockCount, blockSizes, blockStarts* work together and are interpreted into **SubFeatures** on the primary Feature. Each of these SubFeatures are created with a **FeatureSource** category of **block**. If these columns generate SubFeatures then ZENBU can also interpret the *thickStart* and *thickEnd* columns as follows
 - if *thickStart* is not equal to *chromStart* then the region from *chromStart* to *thickStart* is interpreted into a SubFeature of category **5utr**
 - if *thickEnd* is not equal to *chromEnd* then the region from *thickEnd* to *chromEnd* is interpreted into a SubFeature of category **3utr**
- *itemRGB* allow manual coloring of features in tracks (loaded from bed12 or bed9 files). If the itemRgb column is empty it is not inserted into the metadata of the features. To visualize the itemRGB metadata-stored color, make sure the metadata is present and make sure "full_feature" is enabled for source outmode.

for example this BED line

chr5	137801180	137805004	NM_001964	0.00	+
	137801451	137803770	0	2	576,2558 0,1265

is interpreted into the ZENBU data model (here displayed in a ZENBU XML export/interchange format)

```

<feature name="NM_001964" start="137801181" end="137805004"
strand="+">
  <chrom chr="chr5" asm="hg19" ucsc_sm="hg19" ncbi_asm="GRCh37"
taxon_id="9606" length="180915260"/>
    <featureresource category="refgene" name="UCSC_hg19_refgene"
feature_count="35067"/>
    <subfeatures count="4">
      <feature category="5utr" start="137801181" end="137801451"
strand="+"/>
      <feature category="block" start="137801181" end="137801757"
strand="+"/>
      <feature category="block" start="137802446" end="137805004"
strand="+"/>
      <feature category="3utr" start="137803770" end="137805004"
strand="+"/>
    </subfeatures>
  </feature>

```

BED as OSCtable header

BED files can easily be represented with an OSCtable column header line using the ZENBU extended column namespace.

```
eedb:chrom      eedb:start_0base      eedb:end      eedb:name
eedb:score      eedb:strand      eedb:bed_thickstart
eedb:bed_thickend      bed:itemRgb      eedb:bed_block_count
eedb:bed_block_sizes      eedb:bed_block_starts
```

BED specification

The official BED specification is available here <http://genome.ucsc.edu/FAQ/FAQformat.html#format1>

GFF and GTF file support

The GFF (General Feature Format) format consists of one line per feature, each containing 9 columns of data, plus optional track definition lines. The following documentation is based on the Version 2 specifications.

The GTF (General Transfer Format) is identical to GFF version 2.

ZENBU currently provides support for parsing GFF, GFF2 and GTF files. GFF3 support will be provided in the future.

Fields

Fields must be tab-separated. Also, all but the final field in each feature line must contain a value; "empty" columns should be denoted with a '.'.

1. seqname - name of the chromosome or scaffold; chromosome names can be given with or without the 'chr' prefix.
 2. source - name of the program that generated this feature, or the data source (database or project name)
 3. feature - feature type name, e.g. Gene, Variation, Similarity
 4. start - Start position of the feature, with sequence numbering starting at 1.
 5. end - End position of the feature, with sequence numbering starting at 1.
 6. score - A floating point value.
 7. strand - defined as + (forward) or - (reverse).
 8. frame - One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on..
 9. attribute - A semicolon-separated list of tag-value pairs, providing additional information about each feature.
- Format is tag=value or tag <space> value.

Sample GFF output from Ensembl export:

X	Ensembl	Repeat	2419108	2419128	42	.	.	hid=trf; hstart=1; hend=21
X	Ensembl	Repeat	2419108	2419410	2502	-	.	hid=AluSx; hstart=1; hend=303
X	Ensembl	Repeat	2419108	2419128	0	.	.	hid=dust; hstart=2419108; hend=2419128
X	Ensembl	Pred.trans.	2416676	2418760	450.19	-	2	genscan=GENSCAN0000019335
X	Ensembl	Variation	2413425	2413425	.	+	.	
X	Ensembl	Variation	2413805	2413805	.	+	.	

ZENBU interpretation of GFF / GTF files

The GFF/GTF file format easily maps into the ZENBU data model. Both ZENBU and GFF use a 1base-exclusive coordinate system so there is adjust between coordinate spaces.

- *seqname* : is mapped to Feature chromosome
- *source* : is not interpreted but simply stored as metadata on the Feature with the tag *gff:source*
- *feature* is interpreted as a **FeatureSources** category multiplexer. This allows a complex GFF file with many different *feature* / category types to be organized into separate ZENBU FeatureSources after loading.
- *start* : is Feature chrom_start 1base-exclusive coordinate system
- *end* : is Feature chrom_end
- *score* is stored in the Feature significance. On data uploading there is an option to copy the score into an **Expression** value of a specified **DataType**.
- *strand* : is Feature strand
- *frame* : is not interpreted but simply stored as metadata on the Feature
- *attributes* : is parsed into **Metadata** attached to the Feature with the exception of a few special tags mostly derived from the GFF3 [1] format extention
 - ID= is used for feature/subfeature linking, not stored into metadata.
 - Parent= to used for feature/subfeature linking. not stored into metadata.
 - Name= is stored as the name of the Feature, not stored into metadata.

GFF as OSCtable header

GFF files can easily be represented with an OSCtable column header line using the ZENBU extended column namespace.

```
eedb:chrom      gff:source      eedb:fsrc_category
eedb:start.1base    eedb:end      eedb:score      eedb:strand
gff:frame      gff:attributes
```

The **gff:attributes** column has a complete ZENBU parser attached to it. The parser can interpret this column in either the older GFF/GTF tag<space>value format or the GFF2/GFF3 style tag=value format. In the future, this *gff:attributes* parser will be expanded to parse the special GFF3 specification tags for 'feature names' and the GFF3 style of storing feature/subfeature relationships. Currently all data in the *gff:attributes* is parsed into metadata of the Feature.

gff:source and *gff:frame* are currently not interpreted but simply stored as Metadata.

GFF GTF specifications

For more information about this file format, see the documentation on these external websites.

<http://asia.ensembl.org/info/website/upload/gff.html>

<http://www.sanger.ac.uk/resources/software/gff/spec.html>

<http://www.sequenceontology.org/gff3.shtml>

<http://genome.ucsc.edu/FAQ/FAQformat#format4>

References

[1] <http://www.sequenceontology.org/gff3.shtml>

OSCTable

OSCTable1.1 specification

Basic structure

- A simple tabdelimited text file.
- Column order is flexible
- Lines starting with '#' are comments.
- Lines starting with '##' are attributes or metadata of the table. (See **Metadata** section below)
- The first line after the comments/metadata (see below) is a header line, which indicate column names of the table.
- All the comment and attribute lines should appear above the header line
- The first column should describe a 'key' (unique in many cases, but not necessarily) of the data, and the column name should be 'id'
- If a cell needs to include multiple values, comma',' is recommended to be used as a separator.
- All the columns should be described in Metadata (See 'Metadata' section below)

Metadata ('##' lines)

The basic structure is '##qualifier = value'.

Required metadata: FileFormat, Date, ProtocolREF, ColumnVariable, ContactName, ContactEmail

'genome_assembly' for parameter value is required, when using

```
##NameSpace=genomic_coordinate
```

Required (mandatory) metadata

- **FileFormat** -- describes file format of this file.
##FileFormat = OSCTable1.1
- **Date** -- describes the date when the data file is generated
##Date = 20090602
- **ProtocolREF** -- describes the protocol used to generate the data file.
##ProtocolREF = CAGEmappingv1.0
- **ColumnVariable** -- describes ALL the columns used in the data file


```
##ColumnVariable[start] = this is a start position of the genomic coordinate
      ##ColumnVariable[end] = this is a stop position of the genomic coordinate
      ##ColumnVariable[norm.THP10h] = this is TPM normalized value with 10h
      ##ColumnVariable[entrez_gene_id] = Entrez gene ID, which is assigned to the cluster
```
- **ContactName** -- describes the contact name about the data file.
##ContactName = Hideya Kawaji
- **ContactEmail** -- describes the contact address about the data file
##ContactEmail = kawaji@gsc.riken.jp

Optional metadata

- **InputFile** -- describes the file(s) used to generate the data file


```
##InputFile = lane1.fa
##InputFile = lane2.fa
```
- **ParameterValue** -- describes the parameter(s) used to generate the data file in the protocol. The parameter(s) should be consistent with the protocol description


```
##ParameterValue[alignment_program] = BWA
##ParameterValue[aligment_program_version] = 1.3.5
##ParameterValue[UCSC_gene_tracks] = RefSeq
##ParameterValue[UCSC_gene_tracks] = ENSEMBL transcript
```
- **NameSpace** -- describes the name space for the column names. See below (NameSpace)


```
##NameSpace=genomic_coordinate
##NameSpace=expression
```

Column Name Spaces

- A set of column names (and parameters) to be used for a specific purpose or context.
- The same column names with the same name space are recognized as the same (equivalent) meaning.
- Supported name space: genomic_coordinate, expression

genomic_coordinate

- column names are: chrom, start.0base, start.1base, end, strand
- parameter value: genome_assembly
 - chrom: chromosome name used in the genome assembly. For example, chr1, chr2, chr3, ... chrM for the UCSC hg18 genome assembly.
 - start.0base: start position (bp) on the chromosome in 0start coordinate system (BED, PSL, BLAT, exonerate, and nexAlign style)
 - start.1base: start position (bp) on the chromosome in 1start coordinate system (conventional coordinate system; adopted in GFF as well)
 - end: end position (bp) on the chromosome
 - strand: strand on the chromosome; optional
 - Note:
 - All of the above columns are not necessarily required. For example, start.0base would not be required if you have start.1base, and strand would not be required if the annotation do not have strand distinction
 - **'genome_assembly' for parameter value is required.**

expression

- **generic expression tags** -- column names follow the form : exp.YYY.ZZZ, raw.ZZZ, norm.ZZZ, or mapcount
 - **exp.YYY.ZZZ** is the general form for describing an expression column. *YYY* labels the **Data Type** of the expression and should not include any dot (.) characters. **ZZZ** indicates the name of the **Experiment** source of the expression.
 - **raw.ZZZ** is shorthand for **exp.raw.ZZZ** where *raw* is a datatype for un-processed values of expression such as raw_counts and raw signal intensities.
 - **norm.ZZZ** is shorthand for **exp.norm.ZZZ** where *norm* is a datatype for normalized value.
 - **mapcount** specifies the number of locations where this element has been mapped onto the genome.

ZENBU interpretation of OSCtable files

OSCtable is one of the main interchange formats for ZENBU. It allows all possible mapping of data into the ZENBU data model. Since the OSCtable specification is highly flexible, it was possible for the ZENBU OSCtable parser to have an extended vocabulary of metadata directives and column name spaces. In addition, the ZENBU OSCtable parser is able to parse both tab-separated and space-separated files.

Metadata

While the official OSCtable specification include mandatory metadata elements, the ZENBU OSCtable parser relaxes this requirement. All ## metadata lines are parsed as optional metadata. Even ##ColumnVariable[] directives are consider as optional metadata. The only requirement for a valid OSCtable for ZENBU is the column header line.

The primary ## Metadata syntax used by ZENBU are the

```
##ParameterValue[key] = value
##ColumnVariable[col_name] = description
##key = value
```

Experiment metadata

The original OSCtable specification works very well with single experiment data files, but does not provide enough fine control of assigning metadata in a multi-experiment data file. To alleviate this, ZENBU added an additional metadata directive to the OSCtable specification

- **##ExperimentMetadata[experiment-name][key] = value**

experiment-name is the same as the ZZZ in the expression column header descriptions. By referencing to the experiment-name it is possible to have the same experiment used in multiple columns with different datatypes.

Example (please note that the text is wrapped below for display)

```
##ParameterValue[filetype] = osc
##ParameterValue[genome] = mm9
##ColumnVariable[eedb:chrom] = chromosome name
##ColumnVariable[eedb:start.Obase] = chromosome start in Obase
coordinate system
##ColumnVariable[eedb:end] = chromosome end
##ColumnVariable[eedb:strand] = chromosome strand
##ColumnVariable[eedb:score] = score or significance of the feature
##ColumnVariable[exp.tagcount.Mouse_Embryoid_Body_RNAseq_exonic] =
tagcount Mouse_Embryoid_Body_RNAseq_exonic
##ExperimentMetadata[Mouse_Embryoid_Body_RNAseq_exonic] [eedb:display_name]
= Mouse_Embryoid_Body_RNAseq_exonic
##ExperimentMetadata[Mouse_Embryoid_Body_RNAseq_exonic] [eedb:platform]
= SQRL_RNAseq
##ExperimentMetadata[Mouse_Embryoid_Body_RNAseq_exonic] [description] =
This is the exon junction signal for RNAseq of Mouse Embryoid body
cells after 4 days of differentiation to the 'primitive streak stage'
(see PMID:17286599 , and should contain expression of brachyury, mixl1,
tbx6, and flk1) carried out at the IMB on Applied Biosystems SOLiD
system (PMID: 18516046). Mouse strain: SV129. Mapping: published
```

```

version.

##ExperimentMetadata[Mouse_Embryoid_Body_RNAseq_exonic][strain] = SV129
##ExperimentMetadata[Mouse_Embryoid_Body_RNAseq_exonic][tissue] = Mouse
Embryoid body
##ColumnVariable[exp.tagcount.Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal]
= tagcount Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal
##ExperimentMetadata[Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal][eedb:display_name]
= Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal
##ExperimentMetadata[Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal][eedb:platform]
= SQRL_RNAseq
##ExperimentMetadata[Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal][description]
= Description: This is the exonic signal for RNAseq of Mouse Embryonic
Stem Cells carried out at the IMB on Applied Biosystems SOLiD system
(PMID: 18516046). Mouse strain: SV129. Mapping: published version
##ExperimentMetadata[Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal][strain]
= SV129
##ExperimentMetadata[Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal][tissue]
= Mouse Embryonic Stem Cells
eedb:chrom      eedb:start.0base      eedb:end      eedb:name
eedb:score      eedb:strand

exp.tagcount.Mouse_Embryoid_Body_RNAseq_exonic
exp.tagcount.Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal
exp.tpm.Mouse_Embryoid_Body_RNAseq_exonic
exp.tpm.Mouse_Embryonic_Stem_Cell_RNAseq_exonic_signal

chr7      52253690      52253691      block_chr7:52253691..52253691+
        186.00      +      84.00      102.00      8.40      10.20
chr7      52253691      52253692      block_chr7:52253692..52253692+
        184.00      +      83.00      101.00      8.30      10.10
chr7      52253692      52253693      block_chr7:52253693..52253693+
        185.00      +      83.00      102.00      8.30      10.20
chr7      52253693      52253694      block_chr7:52253694..52253694+
        180.00      +      81.00      99.00      8.10      9.90
chr7      52253694      52253695      block_chr7:52253695..52253695+
        174.00      +      77.00      97.00      7.70      9.70

```

Column name spaces

This section is in reference to the names of columns in the **header line**. ZENBU does not parse the `##Namespace` metadata directive but instead uses a controlled vocabulary of column names for interpreting the data and mapping it into the ZENBU **DataModel**.

official OSCtable column names

The official OSCtable specification has very few predefined column names, but all are understood by ZENBU and easily mapped onto the **Feature**

- *chrom* -- chromosome name
- *start.0base* -- chromosome start in a 0base coordinate system

- *start.1base* -- chromosome start in a 1base coordinate system
- *end* -- chromosome end location
- *strand* -- chromosome strand
- *ID* -- interpreted as the name of the **Feature**
- *exp.YYY.ZZZ* -- interpreted as expression value of datatype YYY for experiment named ZZZ
- *raw.ZZZ* -- interpreted as expression value of datatype *raw* for experiment named ZZZ
- *norm.ZZZ* -- interpreted as expression value of datatype *norm* for experiment named ZZZ

While ZENBU uses a 1-based-inclusive coordinate space internally, it can automatically handle the conversion between coordinate spaces at load time and when exporting data out of the system.

ZENBU additional column namespaces

- *eedb:name* -- the name of the **Feature**
- *eedb:score* -- is stored in the **Feature** significance.
- *eedb:bed_block_count* -- are taken from the BED file specification.
- *eedb:bed_block_sizes* -- are taken from the BED file specification.
- *eedb:bed_block_starts* -- are taken from the BED file specification.
 - These three columns work together and are interpreted into **SubFeatures** on the primary Feature. Each of these SubFeatures are created with a **FeatureSource** category of **block**.
- *eedb:bed_thickstart*
- *eedb:bed_thickend* -- are taken from the BED file specification.
 - if *eedb:bed_thickstart* is not equal to *start* then the region from *start* to *bed:thickStart* is interpreted into a SubFeature of category **5utr**
 - if *eedb:bed_thickend* is not equal to *end* then the region from *bed:thickEnd* to *end* is interpreted into a SubFeature of category **3utr**
- *bed:itemRgb* -- bed file style rgb color stored as Feature Metadata
- *eedb:genome* -- if specified in a column, this allows data from multiple-genomes within the same OSCtable file
- *eedb:fsrc_category* -- causes creation of multiple FeatureSources from a single OSCtable file using the value of this column as different *categories*
- *eedb:sam_flag* -- ZENBU parses strand out of the SAM file *flag* column.
- *eedb:sam_cigar* -- can be parsed into *chrom_end* and subfeatures
- *eedb:sam_opt*
- *gff:attributes* -- GFF2/GFF3 style tag=value; data with extended support for feature/subfeature linking
- *eedb:ctg_cigar* -- a ZENBU specific extension of the cigar concept allowing multiple overlapping subfeature layers with different FeatureSource categories.
 - example *eedb:ctg_cigar* -- 3utr:4738N415M,5utr:375M,block:404M1628N769M1377N977M

Aliased column names

Several column names have aliases to other column name spaces

- *name* -- same as *eedb:name*
- *score* -- same as *eedb:score*
- *eedb::chrom* -- same as *chrom*
- *eedb:start.0base* -- same as *start.0base*
- *eedb:start.1base* -- same as *start.1base*
- *eedb:end* -- same as *end*'
- *eedb:strand* -- same as *strand*
- *eedb:mapcount* -- same as *mapcount*

- *eedb:significance* -- same as *eedb:score*

ignoring columns

The OSCtable allows for easy *wrapping* of any tab-delimited file into an OSCtable by simply pre-pending a header with the appropriate column names. But sometimes these original files contain columns which one might not really need. To simplify the process of loading, ZENBU added a special column-name

- *ignore.xxxx* -- ignore this column, where xxxx would be the original column name

If a column is labeled as such, on loading this column will be stripped from the data file and thrown away. This can simplify the process for bioinformatician and avoid un-needed data-scripting to *munge* data^[1] prior to loading.

feature metadata columns

Any column with an unknown name is mapped into the Metadata of the Feature. The column name becomes the *key* of the metadata.

Wrapping external file formats with OSCtable headers

With the extended vocabulary of the ZENBU OSCtable parser, it is possible to *wrap* external file formats very easily with an OSCtable header and load them into ZENBU. In fact the ZENBU upload support for BED, GTF and GFF are done through wrapping predefined OSCtable column headers onto these files.

BED oscheader

Here is the column header line to wrap a BED file

```
eedb:chrom      eedb:start_0base      eedb:end      eedb:name
eedb:score      eedb:strand      eedb:bed_thickstart
eedb:bed_thickend      bed:itemRgb      eedb:bed_block_count
eedb:bed_block_sizes      eedb:bed_block_starts
```

GFF oscheader

GFF files can easily be represented with an OSCtable column header line using the ZENBU extended column namespace.

```
eedb:chrom      gff:source      eedb:fsrc_category
eedb:start_1base      eedb:end      eedb:score      eedb:strand
gff:frame      gff:attributes
```

The **gff:attributes** column has a complete ZENBU parser attached to it. The parser can interpret this column in either the older GFF/GTF tag<space>value format or the GFF2/GFF3 style tag=value format. The *gff:attributes* column can be used to store feature/subfeature relationships(GFF3 specification), the name of the feature(GFF2 & GFF3), and all variable metadata of the Feature (original GFF specification)

gff:source and *gff:frame* are currently not interpreted but simply stored as Metadata.

SAM oscheader

Here is the column header line to wrap a SAM file

```
eedb:name      eedb:sam_flag      eedb:chrom      eedb:start.1base
              eedb:score      eedb:sam_cigar      sam:mrnm      sam:mpos
sam:isize      eedb:seqread      sam:qual      eedb:sam_opt
```

ENCODE NarrowPeak oscheader

ENCODE narrowPeak (or Point-Source) format is used to provide called peaks of signal enrichment based on pooled, normalized (interpreted) data. It is a BED6+4 format.

- chrom - Name of the chromosome (or contig, scaffold, etc.).
- chromStart - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
- chromEnd - The ending position of the feature in the chromosome or scaffold. The chromEnd base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and span the bases numbered 0-99.
- name - Name given to a region (preferably unique). Use '.' if no name is assigned.
- score - Indicates how dark the peak will be displayed in the browser (0-1000). If all scores were '0' when the data were submitted to the DCC, the DCC assigned scores 1-1000 based on signal value. Ideally the average signalValue per base spread is between 100-1000.
- strand - +/- to denote strand or orientation (whenever applicable). Use '.' if no orientation is assigned.
- signalValue - Measurement of overall (usually, average) enrichment for the region.
- pValue - Measurement of statistical significance (-log10). Use -1 if no pValue is assigned.
- qValue - Measurement of statistical significance using false discovery rate (-log10). Use -1 if no qValue is assigned.
- peak - Point-source called for this peak; 0-based offset from chromStart. Use -1 if no point-source called.

Here is an example of narrowPeak format:

```
track type=narrowPeak visibility=3 db=hg19 name="nPk" description="ENCODE narrowPeak Example"
browser position chr1:9356000-9365000
chr1    9356548 9356648 .      0      .      182      5.0945  -1  50
chr1    9358722 9358822 .      0      .      91       4.6052  -1  40
chr1    9361082 9361182 .      0      .      182      9.2103  -1  75
```

To wrap ENCODE broadPeak (or Regions) file formats very easily with an OSCtable header and load them into ZENBU, the **OSC header** should be

```
##ParameterValue[filetype] = osc
##ParameterValue[display_name] =
<THE-DISAPPLY-NAME-TO-BE-EDITED-HERE>
##ExperimentMetadata[x][eedb:display_name] =
<THE-DISPLAY-NAME-TO-BE-EDITED-HERE>
##ColumnVariable[eedb:chrom] = chromosome name
```

```

##ColumnVariable[eedb:start.0base] = chromosome start in 0base
coordinate system
##ColumnVariable[eedb:end] = chromosome end
##ColumnVariable[eedb:strand] = chromosome strand
##ColumnVariable[eedb:score] = score or significance of the feature
##ColumnVariable[exp.signal.x] = measurement of overall (usually,
average) enrichment for the region
##ColumnVariable[exp.qvalue.x] = measurement of statistical
significance using false discovery rate (-log10). Use -1 if no qValue
is assigned
##ColumnVariable[exp.pvalue.x] = measurement of statistical
significance (-log10). Use -1 if no pValue is assigned
##ColumnVariable[point_source] = point-source called for this peak;
0-based offset from chromStart. Use -1 if no point-source called
eedb:chrom      eedb:start.0base      eedb:end      eedb:name
eedb:score      eedb:strand      exp.signal.x      exp.pvalue.x
exp.qvalue.x      point_source

```

Upon data loading, leave the "display name" textbox empty so has to not overwrite the one available from the OSCtable (and keep the featureSource name and ExperimentSource names in sync)

ENCODE BroadPeak oscheader

ENCODE broadPeak (or Regions) format is used to provide called regions of signal enrichment based on pooled, normalized (interpreted) data. It is a BED 6+3 format.

- chrom - Name of the chromosome (or contig, scaffold, etc.).
- chromStart - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
- chromEnd - The ending position of the feature in the chromosome or scaffold. The chromEnd base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and span the bases numbered 0-99. If all scores were '0' when the data were submitted to the DCC, the DCC assigned scores 1-1000 based on signal value. Ideally the average signalValue per base spread is between 100-1000.
- name - Name given to a region (preferably unique). Use '' if no name is assigned.
- score - Indicates how dark the peak will be displayed in the browser (0-1000).
- strand - +/- to denote strand or orientation (whenever applicable). Use '.' if no orientation is assigned.
- signalValue - Measurement of overall (usually, average) enrichment for the region.
- pValue - Measurement of statistical significance (-log10). Use -1 if no pValue is assigned.
- qValue - Measurement of statistical significance using false discovery rate (-log10). Use -1 if no qValue is assigned.

Here is an example of broadPeak format:

```

track type=broadPeak visibility=3 db=hg19 name="bPk" description="ENCODE broadPeak Example"
browser position chr1:798200-800700
chr1    798256 798454 .      116      .      4.89716 3.70716 -1
chr1    799435 799507 .      103      .      2.46426 1.54117 -1
chr1    800141 800596 .      107      .      3.22803 2.12614 -1

```

To wrap ENCODE broadPeak (or Regions) file formats very easily with an OSCTable header and load them into ZENBU, the **OSCTable header** should be

```
##ParameterValue[filetype] = osc
##ParameterValue[display_name] =
<THE-DISAPPLY-NAME-TO-BE-EDITED-HERE>
##ExperimentMetadata[x][eedb:display_name] =
<THE-DISPLAY-NAME-TO-BE-EDITED-HERE>
##ColumnVariable[eedb:chrom] = chromosome name
##ColumnVariable[eedb:start.Obase] = chromosome start in Obase
coordinate system
##ColumnVariable[eedb:end] = chromosome end
##ColumnVariable[eedb:strand] = chromosome strand
##ColumnVariable[eedb:score] = score or significance of the feature
##ColumnVariable[exp.signal.x] = measurement of overall (usually,
average) enrichment for the region
##ColumnVariable[exp.qvalue.x] = measurement of statistical
significance using false discovery rate (-log10). Use -1 if no qValue
is assigned
##ColumnVariable[exp.pvalue.x] = measurement of statistical
significance (-log10). Use -1 if no pValue is assigned
eedb:chrom      eedb:start.Obase      eedb:end      eedb:name
eedb:score      eedb:strand        exp.signal.x    exp.pvalue.x
exp.qvalue.x
```

Upon data loading, leave the "display name" textbox empty so has to not overwrite the one available from the OSCTable (and keep the featureSource name and ExperimentSource names in sync).

References

- [1] http://en.wikipedia.org/wiki/Data_munging

Concepts

ZENBU Configurations

The ZENBU system is designed around the concept of user configurable visualization and processing. This is accomplished through a **configuration** saving and sharing system. The configurations in ZENBU are broken down into three main concepts: Tracks, Views, and Scripts.

Track configuration

Tracks are one of the central concepts in the ZENBU system.

A Track consists of three parts

- **Data sources** : a collection of one or more data sources which are pooled together to form a dynamic merged data source.
- **Data processing** : an optional *script* of data signal processing modules which manipulates and analyzes the data. This processing can be for either data visualization purposes or strictly for use in data exporting and offline analysis.
- **Visualization parameters** : a set of default visualization parameters which can be used by visualization systems like our glyphs genome browser when the track is loaded into a View.

Each Track in the system is internally identified based on its configuration content and a hashkey system. Anytime a user changes the definition of a Track, its configuration is different and receives a different hashkey. If two users happen to configure the same track, based on content description, they will be the same and get the same hashkey. This concept to Tracks-defined by their descriptions (not based on names or saving) is the foundation for the TrackCache system and enables the same TrackCache to be utilized by different users in different views.

In addition there is a **Track Sharing** system in place. This allows any user to name and describe their tracks to make it share with other users of the system. This is done in the track reconfiguration panel with the button labeled <share track>. This enables every user created Track to be referenced and reused. Although every different track configuration generates a new hashkey and track configuration within the TrackCache system, only those Tracks which are explicitly saved, can be searched for in the data explorer and shared as a predefined track.

The Track and its unique content-based hashkey are the foundation on which the Data Download, and TrackCache systems are built.

View configuration

A View is an ordered collection of Tracks with additional parameters.

These additional parameters include a *Title* and *Description* of the view, and every other user adjustable parameter of the gLyphs genome browser. In addition, a View can override any of the default visualization parameters from predefined Tracks which were loaded into the view.

When a View configuration is saved, its Title and Description are searchable in the data explorer via the metadata searching system.

Script configuration

A **Script configuration** is composed of a data processing script with additional parameters.

These parameters include a *Title* and *Description* of the script to allow for searching and sharing, and a set of recommended visualization parameters.

When a predefined script is loaded into a new or reconfigured Track, it will reset the visualization parameters to its recommendations. The user then has the option to override these visualization defaults and to even modify the script modules and parameters.

Saving and Sharing configurations

Every configuration is able to be saved by users of the systems. Saved configurations are appended with both a *Title* and a *Description* which are then searchable via the metadata searching systems in both the data explorer and gLyphs genome browser interfaces.

Data Sources

The ZENBU system was designed on first principle to be a collaborative OMICS data integration system where primary data is dynamically uploaded by users of the system. Because of the data processing capabilities designed into ZENBU, this uploaded data is used as *sources* for input into data processing scripts where the result of that processing can then be downloaded or visualized in the ZENBU genome browser.

The primary data which has been **uploaded** into the system are collectively referred to as **Data Sources**. When data is loaded into the system, each data file is translated into the internal ZENBU **Data Model** and grouped into **one or more annotation data sources** and/or **Expression experiment data sources** depending in the uploaded data file format and upload parameter options.

Since ZENBU is based on loading data into an abstract **data model** concept, it is important to be able to find our data after upload since there is not always a direct one-to-one mapping of *data upload file* to **DataSource**. To accomplish this ZENBU utilized a metadata search system. When data is uploaded, user are asked to provide a *name* and *description* of the file and its data content. By providing good descriptions, it not only allows for easily finding your data at a latter time, but also makes it easier for your collaborators to understand the content and nature of your data.

Uploadable data file formats

Before data can become available as a *Data Source* in ZENBU, it must first be uploaded into the system through one of the supported file types. The file types currently supported by ZENBU upload are:

- **BAM & SAM** sequence alignment files.
- **BED** UCSC style genome annotation files
- **GFF GFF2 GTF** Ensembl/gbrowse style genome annotation files
- **OSCTable** open format tab separated tables for genome annotation and multiple experiment expression (RIKEN/ZENBU format).

Annotation Data Sources

An **annotation data source** [also called a **FeatureSource**] is a collection of genomic features. This corresponds to a *data set* like "gene sets" or "promoter sets" or "micro array probe set" or "repeat elements". In the UCSC genome browser this is what they refer to as a **UCSC track** (i.e. data track) which was loaded from a BED file. When uploading data into ZENBU, the data in each file is mapped into a one or more **FeatureSources**.

BED is the simplest file format where each BED file maps to a single FeatureSource. There is an option when uploading a BED file to extract expression from it and optionally create an Experiment for the the file. The options include mapping the *score* onto expression or to simply count each location with an expression value of **1**.

BAM/SAM files since they always contain sequence alignments are always mapped into a single FeatureSource and single Experiment.

GFF/GTF files can be mapped into one or more FeatureSources depending on the content of the 3rd column of the file. This 3rd column encapsulates the GFF *feature* concept which is the same as the ZENBU FeatureSource concept. Every different GFF/GTF *feature* type is mapped into a different ZENBU FeatureSource.

OSCTable files are also mapped into one or more FeatureSources depending on how they are configured. The OSCTable format allows for complete flexibility to control how the data is mapped into FeatureSources upon loading into ZENBU.

Expression Experiment Data source

An expression experiment data source [also simply called a **Experiment**] is a collection of expression data. By the definition of the ZENBU data model, each expression data element is attached to a **genomic feature** which is also part of a FeatureSource. Since a genomic Feature can have many Expression data points attached, the Experiment is critical to describing the Expression.

Expression refers to a single measurement data element with an associated expression **Data Type** (eg: "tagcount", "tpm", "mapquality" "score" "pvalue" "rle" to name a few). Depending on the data file used, there can be none to many Experiments associated to each uploaded data file.

BED is primary used only for genomic annotations, but ZENBU allows for optional expression experiments to be defined. If enabled at upload time, an Experiment can be created and associated with the data file. There are options to interpret the BED *score* as expression or to simply count each *bed line location* with an expression value of **1**.

GFF/GTF in the same way as BED files, ZENBU allows the score column to be mapped onto Expression and to create an optional Experiment associated with each GFF/GTF file.

BAM/SAM files since they always contain sequence alignments are always mapped into a single FeatureSource and single Experiment.

OSCTable format allows for complete flexibility to control how the data is mapped into Experiments and Expression. There can be many Experiments defined within a single OSCTable file.

Types of data which can be loaded

ZENBU was designed with a data model abstraction which allows all types of genomic annotation and expression data to be uploaded into the ZENBU system. Here are examples of the different types of experimental and analysis results data which can be loaded by users into ZENBU using the available upload file types.

Genome mapped RNA/DNA sample sequences

This class of data includes RNAseq, shortRNA, CAGE, ChIP-seq. The nature of this data is a sample of DNA or RNA which is processed by a molecular biology protocol and then sequenced. It is now very common to use next generation sequencing instruments like Illumina HiSeq2000, Illumina G3, SOLiD, 454 or Heliscope for this sequencing. This class of instrument produces millions to 100s of millions of short sequences often referred to as sequence-tags. Because of the short nature of these tags, often the best way to analyze them is to first map them onto a reference genome assembly with a program like BWA or TopHat.

The ZENBU system can directly load these genome aligned sequences with out need for additional processing. A common format for these alignments is BAM.

Genome annotations

This class of data is often the result of a bioinformatics analysis pipeline or through manual curation efforts. Common data file formats for genome annotations include BED and GFF/GTF. Since the nature of this data is descriptive, it is sometimes very useful to include descriptive metadata along with genomic location information. The OSCTable format is a highly flexible format which allows for attaching very complex metadata, expression, and numerical values onto genomic annotation features.

Microarray expression

The ZENBU system provides a means to load micro-array data. Once microarray data is loaded, it can be processed and visualized as either "expression tracks" or as "annotated-expression hybrid tracks".

Currently the loading of micro-array data is a little complicated, but we hope in the future to make this process easier for users. ZENBU currently has several micro-array probe-sets from Illumina and Affymetrix mapped onto the genome. To load micro-array expression, one needs to download one of the probe-sets into OSCTable format and then to extend the columns of the file to add the raw/normalized microarray expression for each probe. Then upload the new modified OSCTable file back into ZENBU.

Annotated Expression analysis results

The ZENBU system is able to work with very complex analysis results which can often consist of genomic locations, descriptive metadata, expression signal from multiple samples, and numerical analysis results. The OSCTable format allows any complex table of data to be mapped with column names and loaded into ZENBU. Since each analysis process/result is often unique, the flexibility of OSCTable allows the data to be loaded in its original form, rather than having to convert it into a *standard* file format and thereby have to throw away some information.

After loading complex analysis data from OSCTable files, ZENBU can process and manipulate any and all aspects of the dataset. It can be treated as simply as genomic annotation, or in complex ways with data processing and hybrid annotated-expression visualization tracks

Novel genomes

The ZENBU system was designed in the era where many novel genomes are being generated, so we made sure the system could expand into the range of 1000s of genomes. In ZENBU, genomes are treated as a "namespace" which means that the process of creating a new genome is as easy as naming it. For example if a data file names genome "human-bobsmith" chromosome "chr3" it will create it if it does not exist. The reference genome sequence and size of the chromosome are treated as additional data loaded into that "genome namespace". Once uploaded the ZENBU system will become aware of the new genome name.

Although currently not available through the web interfaces, novel genome sequences can be loaded by system administrators through ZENBU command line tools. In a near future upgrade this functionality will be available via the web interface for user upload .

Virtual DataSources - Track Data pooling

Although the DataSources are defined at data load time, the ZENBU system provides a dynamic flexible *data mixing* technology called **Data Pooling**. This allows for *virtual* DataSources to be created when configuring ZENBU tracks by mix-and-match from the uploaded physical DataSources. This provides for a great deal of flexibility both in terms of data loading and data processing without requiring external data processing and additional data loading.

Data Stream Pool

The ZENBU system allows for the dynamic creation of merged *virtual* Data Sources referred to as "data stream pools". This provides for a great deal of flexibility both in terms of data loading and data processing. With data pooling, there is no need to load new data every time a different "mix" is needed when configuring ZENBU tracks. One can simply use the data already loaded in the ZENBU system and create a new *virtual* DataSource mix.

Data pooling can be on a mix of either annotation FeatureSources or a mix of expression Experiment data sources. In both cases the *mixed* pool is a union or merging of the data.

The main advantage of data pooling is for data processing and analysis. It becomes possible to pool many experiments from many samples or across replicates for differential expression visualization and analysis. And with the data download capabilities these processed data pools can be exported into statistical systems like R and BioConductor for more advanced analysis. It also is easy to create merged annotation datasets without requiring a special upload. For example it is possible to create a merged data pool of all gene models (gencode, refseq, ensembl, ucsc known gene) in a single track.

But data pooling also can help with the data organization and the data loading process. Since ZENBU can merge data on demand, it becomes possible to organize data prior to upload at a more atomic level. For example we can keep each sequencing replicate of a sample in separate BAM files and allow ZENBU to create the *virtual mix* of all data from the same sample. This provides a high level of flexibility for being able to load the data files as they exist, rather than requiring pre-processing of the data prior to loading. This also gives the user great flexibility in creating new *groupings* of data after the data has been loaded even if the grouping was not in the original experimental design.

Examples

To better illustrate the concept of data pooling, we present several examples.

Pooling annotation FeatureSources - different mixes of repeat sets

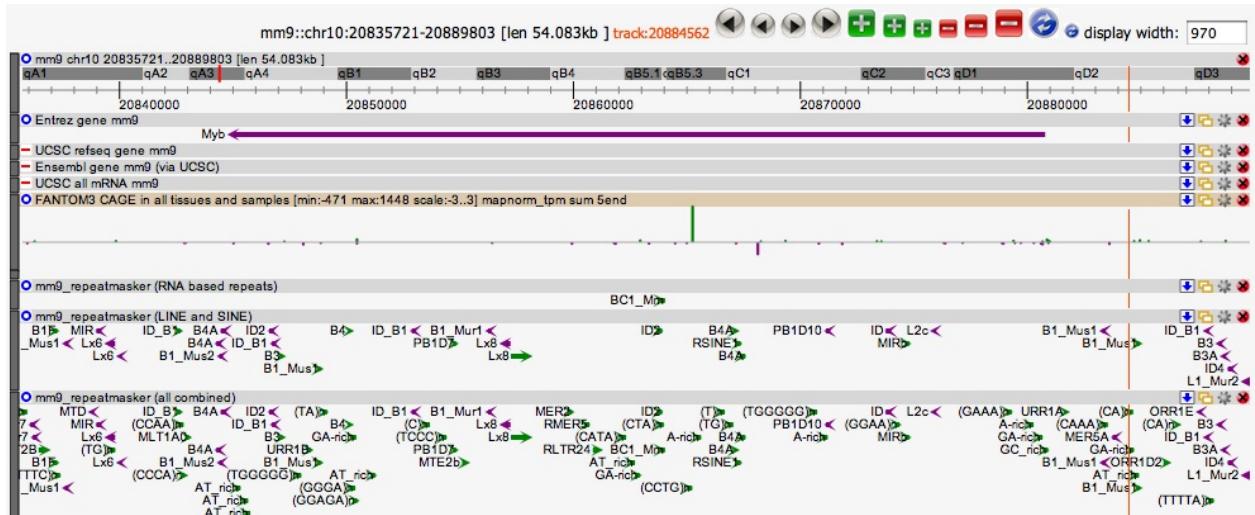
There are many different classes of repeats in the genome. Sometimes we need to work with specific repeat class, sometimes we want to work with a broader class of repeats and sometimes we don't care about the class and are only concerned if *any* repeat is present. The data pool works very well here.

For example we have loaded the mouse mm9 repeatmasker data from UCSC where each class of repeat is mapped into a different annotation FeatureSource

```
http://fantom.gsc.riken.jp/zenbu/dex/#section=Annotation;asm=mm9;search=repeat
```

In this example we have created three different tracks with different data pooling of the repeat annotation FeatureSources.

1. repeat track is mix of all RNA-based repeats (RNA, rRNA, scRNA, snRNA, srpRNA, tRNA)
2. repeat track is mix of only LINE or SINE repeats
3. repeat track is mix of all 16 different repeat classes



link to genome browser view referenced above http://fantom.gsc.riken.jp/zenbu/gLyphs/?#config=EW5pkPEsLN_uq9B-warpcD;loc=mm9::chr10:20835721..20889803

Pooling expression Experiments - FANTOM3 differential promoter expression

Differential expression is one of the key aspects when studying RNA. RNA is inherently expressed at different levels in different tissues and samples. One available RNA expression technique is called CAGE which not only records the expression level of RNAs but also identifies the RNA's 5' end location on the genome which is interpreted as the "transcription start site" for the RNA. In the FANTOM3 project, there were 465 different mouse samples which were analyzed with CAGE and sequenced. In this example we create three different tracks with different mixes of these 465 sample Experiments.

In the following view we have created three different expression tracks with different data pooling of the FANTOM3 CAGE expression Experiments.

1. expression track with virtual mix of all 465 FANTOM3 CAGE expression samples. In the view on the GSN gelsolin gene we can see two distinct CAGE expression peaks which correspond to different transcription starting sites and thus expression of different splicing isoforms of the GSN gene.

2. expression track with only 26 blood related FANTOM3 CAGE samples. In this track we can see that the blood related samples exclusively express the left most CAGE transcription start site. By comparing to the known annotation tracks, we can see there is a long Ensembl transcript/gene (ENSMUST00000113016gene) which aligns perfectly with this CAGE peak. It is thus easy to infer that this splicing form is the one expressed in the blood.
3. expression track with only 28 lung related FANTOM3 CAGE samples. In this track we can see that the lung related samples exclusively express the right most CAGE transcription start site and thus the main splicing isoform of the GSN gene.



link to genome browser view referenced above <http://fantom.gsc.riken.jp/zenbu/gLyphs/?#config=OIs9ocIoGpXlzbRQEd418B;loc=mm9::chr2:35103454..35167609>

Data Abstraction Model

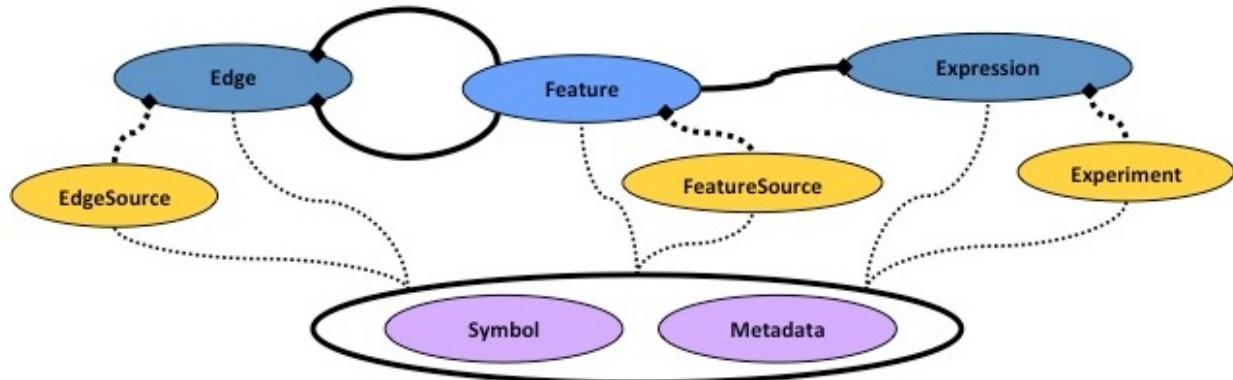
Although the internal data abstraction model is not obviously exposed to the users of the system, understanding the internal model can help to understand how data is stored and processed by the system. For advanced users of the script processing system, understanding the data model is important to write your own custom processing scripts.

ZENBU internal data model

The data model is an evolution of the model first described in the FANTOM4 EdgeExpress system^[1] (Genome Biol. 2009;10(4):R39. Epub 2009 Apr 19^[2]).

The ZENBU data model is composed of

- data sources (FeatureSource, Experiment, EdgeSource)
- genomic location information (Features)
- numerical expression value data (Expressions)
- connections between Features (Edges)
- and descriptive metadata.



Features and associated SubFeatures

Features

The Feature is the central element in the data model.

It represents a generic object in the system. A Feature must belong to a FeatureSource. The primary attributes of a Feature include a name, a significance, and genomic coordinates.

Genomic coordinates are defined as:

- chromosome assigned to a specific species assembly
- chrom_start
- chrom_end
- strand

For Features to be visualized in the ZENBU genome browser, genomic coordinates are mandatory.

ZENBU Feature genomic coordinates are 1base-exclusive which means that chromosomes starts at 1 and features of length 1bp have the same chrom_start and chrom_end.

In addition a Feature can have Expression and Metadata attached to it.

SubFeatures

In addition a Feature can have other Features attached under it which are called **SubFeatures**.

The most common use for SubFeatures is to define exon/intron/UTR spliced gene-model details of the primary Feature, but any type of *category* can be defined for the **FeatureSources** of attached SubFeatures. For example one could define protein domains SubFeature regions of the primary Feature with different categories in addition to the exon structure.

SubFeatures are allowed to overlap each other and do not need to be exclusive.

Currently (as of version 2.5) SubFeatures cannot have another layer of SubFeatures under them.

Here is example of very complete Feature with subfeatures and expression (here displayed in a ZENBU XML export/interchange format)

```
<feature name="NM_001964" start="137801181" end="137805004"
strand="+">
  <chrom chr="chr5" asm="hg19" ucsc_sm="hg19" ncbi_asm="GRCh37"
taxon_id="9606" length="180915260"/>
  <featureresource category="refgene" name="UCSC_hg19_refgene"
feature_count="35067"/>
  <subfeatures count="4">
    <feature category="5utr" start="137801181" end="137801451"
strand="+">
      <feature category="block" start="137801181" end="137801757"
strand="+">
        <feature category="block" start="137802446" end="137805004"
strand="+">
          <feature category="3utr" start="137803770" end="137805004"
strand="+">
        </subfeatures>
    </feature>
```

Expression

Represents a single expression data element. An Expression must be attached to a Feature. In addition to the actual expression value, each Expression element has a mandatory DataType. The Expression DataType is used to describe and categorize the values so that expression from many FeatureSources and many Experiments can be pooled together for comparison. Example DataTypes include "tagcount", "tpm", "mapquality" "score" "pvalue" "rle" to name a few.

By definition each Expression data element has one Feature, one Experiment, one FeatureSource, one **DataType** and one value (number).

Edge

A connection between two Features in the system. Currently Edges are rarely used in the ZENBU system, but they have been retained from the EdgeExpress system for backward compatibility and possible future expansions.

Data Sources

These represent a collection of data of a certain class in the system and are made visible to the users in the data explorer interface.

Every DataSource has metadata describing the source which also allows for users to search and find data sets so that the data can be manipulated, and visualized.

FeatureSource

A collection of Features. Each Feature is part of only one FeatureSource.

Often used to represent a collection of annotation like "Human hg19 Entrez genes". But in addition, every file uploaded into the system is assigned a primary FeatureSource to represent that file as a collection of data.

FeatureSources can be dynamically generated by processing modules of the system to represent dynamically created Features.

Experiment

A collection of Expression data, and by connection a collection of Features.

Since a Feature can have many Expression objects attached, the Experiment is critical to describing the Expression.

EdgeSource

A collection of Edges.

This is rarely used by the current ZENBU system, but has been retained for backward compatibility to EdgeExpressDB and for future expansion capabilities.

Metadata system

Metadata is descriptive text which can be attached to any object in the ZENBU datamodel. Metadata is divided into two concepts. Metadata and Symbols.

Metadata

Metadata elements are not searchable but represent a blob of text or data.

The ZENBU system provides automatic **keyword symbol** extraction from Metadata text so that effectively to the user, the Metadata appears searchable.

In general Metadata is used for descriptive text, but it can also be XML or uuencoded data.

Symbols

Symbols are small atomic text units which can be searched.

These are often keywords or controlled vocabulary terms. Symbols can be ad-hoc or from controlled Ontologies.

Search system

The ZENBU system provides a complete metadata search system modeled on google/yahoo searching capabilities, with the addition of rigorous logic control - ***and, or, not*** and parenthesis ()

References

- [1] <http://fantom.gsc.riken.jp/4/edgeexpress>
 - [2] <http://dx.doi.org/10.1186/gb-2009-10-4-r39>
-

Article Sources and Contributors

Source: <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3376> **Contributors:** Jessica Severin, MediaWiki default, Nicolas.bertin

ZENBU gLyphs genome browser **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3423> **Contributors:** Jessica Severin, Nicolas.bertin

Track visualization styles **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3310> **Contributors:** Jessica Severin, Nicolas.bertin

Configuring Tracks **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3103> **Contributors:** Jessica Severin

Data Download **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2848> **Contributors:** Jessica Severin, Nicolas.bertin

TrackCaching System **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2638> **Contributors:** Nicolas.bertin

Region Location **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2205> **Contributors:** Jessica Severin

Data Explorer Interface **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3308> **Contributors:** Jessica Severin, Nicolas.bertin

Metadata searching **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=969> **Contributors:** Jessica Severin

Metadata editing **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2118> **Contributors:** Jessica Severin, Nicolas.bertin

Secured user profiles **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2636> **Contributors:** Jessica Severin

Data sharing collaborations **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=873> **Contributors:** Jessica Severin

Data loading **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3434> **Contributors:** Jessica Severin, Nicolas.bertin

BAM and SAM file support **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=1187> **Contributors:** Jessica Severin

BED file support **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3371> **Contributors:** Jessica Severin, Nicolas.bertin

GFF and GTF file support **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=1189> **Contributors:** Jessica Severin

OSCtable **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3445> **Contributors:** Jessica Severin, Nicolas.bertin

ZENBU Configurations **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3273> **Contributors:** Jessica Severin, Nicolas.bertin

Data Sources **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2039> **Contributors:** Jessica Severin, Nicolas.bertin

Data Stream Pool **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=2740> **Contributors:** Jessica Severin

Data Abstraction Model **Source:** <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?oldid=3318> **Contributors:** Jessica Severin, Nicolas.bertin

Image Sources, Licenses and Contributors

File:ZENBU_genomebrowser_wold.crop.1.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:ZENBU_genomebrowser_wold.crop.1.png *License:* unknown
Contributors: Nicolas.bertin

File:DEX.CuratedViews11Oct12.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.CuratedViews11Oct12.png> *License:* unknown *Contributors:* Nicolas.bertin

File:User-collaborations.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-collaborations.png> *License:* unknown *Contributors:* Jessica Severin

File:ZENBU_genomebrowser_wold.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:ZENBU_genomebrowser_wold.jpg *License:* unknown *Contributors:* Jessica Severin

File:Save_view_button.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Save_view_button.jpg *License:* unknown *Contributors:* Jessica Severin

File:Wold_RNAseq_many_views_same_data.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Wold_RNAseq_many_views_same_data.png *License:* unknown
Contributors: Jessica Severin

File:Annotation-track-transcript.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-transcript.png> *License:* unknown *Contributors:* Jessica Severin

File:LongRNASeq_CSH_exonic_expression.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:LongRNASeq_CSH_exonic_expression.png *License:* unknown
Contributors: Jessica Severin

File:Hybrid_tracks_longRNAseq.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Hybrid_tracks_longRNAseq.png *License:* unknown *Contributors:* Jessica Severin

File:Track_controls-download.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-download.jpg *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_wholeregion.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_wholeregion.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_selection.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_selection.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_feature_selection.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_feature_selection.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_widget-reattach.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_widget-reattach.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_widget-exportdata.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_widget-exportdata.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_widget-config.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_widget-config.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_configpanel.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_configpanel.png *License:* unknown *Contributors:* Jessica Severin

File:Experiment-graph_activate.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Experiment-graph_activate.png *License:* unknown *Contributors:* Jessica Severin

File:Track_selection.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_selection.png *License:* unknown *Contributors:* Jessica Severin

File:Track_selection-widgets-magnify.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_selection-widgets-magnify.png *License:* unknown *Contributors:* Jessica Severin

File:Track_selection-widgets-sequence.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_selection-widgets-sequence.png *License:* unknown *Contributors:* Jessica Severin

File>Create-new-track.jpg *Source:* [http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File/Create-new-track.jpg](http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File>Create-new-track.jpg) *License:* unknown *Contributors:* Jessica Severin

File>Create_new_track_panel.jpg *Source:* [http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File/Create_new_track_panel.jpg](http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File>Create_new_track_panel.jpg) *License:* unknown *Contributors:* Jessica Severin

File:Add-shared-track.jpg *Source:* [http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File/Add-shared-track.jpg](http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File>Add-shared-track.jpg) *License:* unknown *Contributors:* Jessica Severin

File>Add_predef_tracks_panel.png *Source:* [http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File/Add_predef_tracks_panel.png](http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File>Add_predef_tracks_panel.png) *License:* unknown *Contributors:* Jessica Severin, Nicolas.bertin

File:Saveview_panel.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Saveview_panel.png *License:* unknown *Contributors:* Jessica Severin, Nicolas.bertin

File:Saveview_select_collaboration.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Saveview_select_collaboration.png *License:* unknown *Contributors:* Jessica Severin

File:Saveview_new_url_title_desc.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Saveview_new_url_title_desc.png *License:* unknown *Contributors:* Jessica Severin, Nicolas.bertin

File:Saveview_user_login.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Saveview_user_login.png *License:* unknown *Contributors:* Jessica Severin

File:Export_svg_panel.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Export_svg_panel.png *License:* unknown *Contributors:* Jessica Severin

File:Export_svg_button.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Export_svg_button.png *License:* unknown *Contributors:* Jessica Severin

File:Track_controls-deactive.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-deactive.png *License:* unknown *Contributors:* Jessica Severin

File:Track_control_widgets.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_control_widgets.jpg *License:* unknown *Contributors:* Jessica Severin

File:Track_controls-closestrack.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-closestrack.jpg *License:* unknown *Contributors:* Jessica Severin

File:Track_controls-reconfigure_track.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-reconfigure_track.jpg *License:* unknown *Contributors:* Jessica Severin

File:Reconfigure-wold-heatmap.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Reconfigure-wold-heatmap.png> *License:* unknown *Contributors:* Nicolas.bertin

File:Track_controls-copytrack.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-copytrack.jpg *License:* unknown *Contributors:* Jessica Severin

File:Track_controls-contract.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-contract.jpg *License:* unknown *Contributors:* Jessica Severin

File:Track_controls-expand.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_controls-expand.jpg *License:* unknown *Contributors:* Jessica Severin

File:Glyphs_navigation_buttons.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_navigation_buttons.png *License:* unknown *Contributors:* Jessica Severin

File:Glyphs_feature-search-egr1.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_feature-search-egr1.png *License:* unknown *Contributors:* Jessica Severin

File:Glyphs_gene-search-metadata.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_gene-search-metadata.png *License:* unknown *Contributors:* Jessica Severin

File:Glyphs_direct_coordinate_entry.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_direct_coordinate_entry.png *License:* unknown *Contributors:* Jessica Severin

File:Track_visualization_interface.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_visualization_interface.jpg *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-arrow.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-arrow.png> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-thickarrow.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-thickarrow.png> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-mediumarrow.jpg *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-mediumarrow.jpg> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-centroid.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-centroid.png> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-thicktranscript.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-thicktranscript.png> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-box.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-box.png> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-line.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-line.png> *License:* unknown *Contributors:* Jessica Severin

File:Annotation-track-exon.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-exon.png License: unknown Contributors: Jessica Severin

File:Annotation-track-medium-exon.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-medium-exon.png License: unknown Contributors: Jessica Severin

File:Annotation-track-thin-exon.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-thin-exon.png License: unknown Contributors: Jessica Severin

File:Annotation-track-utr.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-utr.png License: unknown Contributors: Jessica Severin

File:Annotation-track-thin.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Annotation-track-thin.png License: unknown Contributors: Jessica Severin

File:Glyphs_expression_track.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_expression_track.png License: unknown Contributors: Jessica Severin

File:Glyphs_expression_track-chipchip.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_expression_track-chipchip.png License: unknown Contributors: Jessica Severin

File:Woldlab_RNAseq_exonic_expression_signal.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab_RNAseq_exonic_expression_signal.jpg License: unknown Contributors: Jessica Severin

File:Expression_track_configuration.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Expression_track_configuration.jpg License: unknown Contributors: Jessica Severin

File:Woldlab_RNAseq_exonic_expression_signal_100bp.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab_RNAseq_exonic_expression_signal_100bp.jpg License: unknown Contributors: Jessica Severin

File:Woldlab_RNAseq_exonic_expression_signal_withgaps.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab_RNAseq_exonic_expression_signal_withgaps.jpg License: unknown Contributors: Jessica Severin

File:Express_track_height_resize.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Express_track_height_resize.jpg License: unknown Contributors: Jessica Severin

File:Express_track_fantom4_cage_locale.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Express_track_fantom4_cage_locale.jpg License: unknown Contributors: Jessica Severin

File:Configure_spectrum_visualization.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_spectrum_visualization.png License: unknown Contributors: Jessica Severin

File:Woldlab-rnaseq-exonic-spectrum.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab-rnaseq-exonic-spectrum.png License: unknown Contributors: Jessica Severin

File:Hybrid_track_colored_features_config.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Hybrid_track_colored_features_config.jpg License: unknown Contributors: Jessica Severin

File:Hybrid_track_gencode_gene_RNAseq_expression.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Hybrid_track_gencode_gene_RNAseq_expression.jpg License: unknown Contributors: Jessica Severin

File:Woldlab-intronsupport.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab-intronsupport.png License: unknown Contributors: Jessica Severin

File:Woldla_exonic_rnaseq_heatmap.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldla_exonic_rnaseq_heatmap.png License: unknown Contributors: Jessica Severin

File:Woldlab-RNAseq-heatmap-mousehover.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab-RNAseq-heatmap-mousehover.png License: unknown Contributors: Jessica Severin

File:Woldlab-RNAseq-heatmap-sample-sort.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Woldlab-RNAseq-heatmap-sample-sort.png License: unknown Contributors: Jessica Severin

File:Hybrid_track_longRNAseq_spectrum.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Hybrid_track_longRNAseq_spectrum.png License: unknown Contributors: Jessica Severin

File:Reconfigure-wold-heatmap.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Reconfigure-wold-heatmap.jpg License: unknown Contributors: Jessica Severin

File>New_track_data_search.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File>New_track_data_search.jpg License: unknown Contributors: Jessica Severin

File:Configure_predefined_script.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_predefined_script.png License: unknown Contributors: Jessica Severin

File:Configure_predefined_script_selected.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_predefined_script_selected.png License: unknown Contributors: Jessica Severin

File:Configure_expression_binning_gui.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_expression_binning_gui.png License: unknown Contributors: Jessica Severin

File:Configure_custom_XML_script.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_custom_XML_script.png License: unknown Contributors: Jessica Severin

File:Track_download_panel.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_download_panel.jpg License: unknown Contributors: Jessica Severin

File:Track_download_file-options.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_download_file-options.jpg License: unknown Contributors: Jessica Severin

File:Track_download_unbuilt.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_download_unbuilt.jpg License: unknown Contributors: Jessica Severin

File:Track_download_already_requested.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_download_already_requested.jpg License: unknown Contributors: Jessica Severin

File>User_downloads_panel.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File>User_downloads_panel.jpg License: unknown Contributors: Jessica Severin

File:Track_download_guest_request.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_download_guest_request.jpg License: unknown Contributors: Jessica Severin

File:Track_download_build_in_progress.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Track_download_build_in_progress.jpg License: unknown Contributors: Jessica Severin

File:Glyphs_feature-search-refseq.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_feature-search-refseq.png License: unknown Contributors: Jessica Severin

File:Glyphs_gene-search-metadata.jpg Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_gene-search-metadata.jpg License: unknown Contributors: Jessica Severin

File:Glyphs_magnify_widget.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_magnify_widget.png License: unknown Contributors: Jessica Severin

File:Glyphs_selection_for_zoom1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_selection_for_zoom1.png License: unknown Contributors: Jessica Severin

File:Glyphs_selection_for_zoom2.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_selection_for_zoom2.png License: unknown Contributors: Jessica Severin

File:Glyphs_feature_selection.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Glyphs_feature_selection.png License: unknown Contributors: Jessica Severin

image:DEX.CuratedViews11Oct12.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.CuratedViews11Oct12.png License: unknown Contributors: Nicolas.bertin

File:DEX_titlebar.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX_titlebar.png License: unknown Contributors: Jessica Severin

File:DEX.collab.selection.1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.collab.selection.1.png License: unknown Contributors: Nicolas.bertin

File:DEX.collab.selection.2.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.collab.selection.2.png License: unknown Contributors: Nicolas.bertin

File:DEX-experiment.search.1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-experiment.search.1.png License: unknown Contributors: Nicolas.bertin

file:DEX-annoation.search1b.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-annoation.search1b.png License: unknown Contributors: Nicolas.bertin

File:DEX-view.2.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-view.2.png License: unknown Contributors: Nicolas.bertin

File:DEX-view2.2.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-view2.2.png License: unknown Contributors: Nicolas.bertin

File:DEX-view.1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-view.1.png License: unknown Contributors: Nicolas.bertin

File:DEX-view.1.2.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-view.1.2.png License: unknown Contributors: Nicolas.bertin

Image:DEX.track.1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.track.1.png License: unknown Contributors: Nicolas.bertin

Image:DEX.track.2.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.track.2.png License: unknown Contributors: Nicolas.bertin

file:DEX-anntab.1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-anntab.1.png License: unknown Contributors: Nicolas.bertin

file:DEX-exptab.1.png Source: http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-exptab.1.png License: unknown Contributors: Nicolas.bertin

file:DEX-shoppingcart.annsel.0.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-shoppingcart.annsel.0.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-shoppingcart.annsel.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-shoppingcart.annsel.1.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-shoppingcart.expsel.2.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-shoppingcart.expsel.2.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-annotation.trackbuilding.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-annotation.trackbuilding.1.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-annotation.trackbuilding.3.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-annotation.trackbuilding.3.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-experiment.trackbuilding.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-experiment.trackbuilding.1.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-shoppingcart.expsel.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-shoppingcart.expsel.1.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-experiment.trackbuilding.2.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-experiment.trackbuilding.2.png> *License:* unknown *Contributors:* Nicolas.bertin

Image:DEX.expconf.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.expconf.1.png> *License:* unknown *Contributors:* Nicolas.bertin

Image:DEX.annconf.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.annconf.1.png> *License:* unknown *Contributors:* Nicolas.bertin

File:Configure_predefined_script_selected.1b.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_predefined_script_selected.1b.png *License:* unknown *Contributors:* Nicolas.bertin

File:Configure_expression_binning_gui.1b.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_expression_binning_gui.1b.png *License:* unknown *Contributors:* Nicolas.bertin

Image:DEX.script.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX.script.1.png> *License:* unknown *Contributors:* Nicolas.bertin

Image:Configure_custom_XML_script.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_custom_XML_script.png *License:* unknown *Contributors:* Jessica Severin

Image:Configure_predefined_script_selected.1c.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Configure_predefined_script_selected.1c.png *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-script.1.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-script.1.png> *License:* unknown *Contributors:* Nicolas.bertin

file:DEX-script.2.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX-script.2.png> *License:* unknown *Contributors:* Nicolas.bertin

File:DEX_edit_view_info.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:DEX_edit_view_info.png *License:* unknown *Contributors:* Jessica Severin

File:Metadata-edition.2.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Metadata-edition.2.png> *License:* unknown *Contributors:* Nicolas.bertin

File:User_login.png *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User_login.png *License:* unknown *Contributors:* Jessica Severin

File:User-collaboration-create-button.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-collaboration-create-button.png> *License:* unknown *Contributors:* Jessica Severin

File:User-collaboration-create.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-collaboration-create.png> *License:* unknown *Contributors:* Jessica Severin

File:User-share-uploaded-data.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-share-uploaded-data.png> *License:* unknown *Contributors:* Jessica Severin

File:User-share-uploaded-data-panel2.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-share-uploaded-data-panel2.png> *License:* unknown *Contributors:* Jessica Severin

File:User-share-uploaded-data-after-sharing.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-share-uploaded-data-after-sharing.png> *License:* unknown *Contributors:* Jessica Severin

File:User-upload.png *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:User-upload.png> *License:* unknown *Contributors:* Jessica Severin

File:Data-pooling-repeat-example.jpg *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Data-pooling-repeat-example.jpg> *License:* unknown *Contributors:* Jessica Severin

File:Data-pooling-experiment-example.jpg *Source:* <http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:Data-pooling-experiment-example.jpg> *License:* unknown *Contributors:* Jessica Severin

File:EEDB_DataModel.jpg *Source:* http://fantom.gsc.riken.jp/zenbu/wiki/index.php?title=File:EEDB_DataModel.jpg *License:* unknown *Contributors:* Jessica Severin

License

Creative Commons Attribution Share Alike
<http://creativecommons.org/licenses/by-sa/3.0/>