Trackster
Visualization in Galaxy
What is trackster?

• Track/data viewer in web browser
• View your data from within Galaxy
• No file transfers to third party
• Use it locally, even without internet
Why trackster?

- Fast, responsive, interactive UI
- Renders in browser, not on server
- Explore your data with your mouse!
Wig, Bedgraph (Line Tracks)

Regular line graph display

Intensity display (shades of gray)

Filled line graph display
Bed (Feature Track)

- 3 levels of detail: automatically adjusts based on what can fit on the screen

Snippet of hg18 all mrna feature track

Sunday, May 16, 2010
High level density display

Feature display with no labels/detail

Feature display with labels, intron indicators, exon indicators
BAM (Read Track)

Detailed view of a BAM file
Paired-ends display with no labels

Paired-ends display with read labels and sequence
- Add tracks, remove tracks, reorder tracks, resize tracks, edit track settings
Well-Integrated
The Technology

- Front-end: HTML5 Canvas, jQuery

- Back-end: Datatype converters, data structures, data providers
HTML5 Canvas

- Simple graphics API that lets you draw inside a `<canvas></canvas>` element
- Supported by all major browsers, except IE
```
<script type="text/javascript" charset="utf-8">
    function draw() {
        var ctx = document.getElementById('canvas').getContext('2d');
        ctx.translate(75,75);
        for (var j = 0; j < 16; j++) {
            ctx.fillStyle = 'rgba('+20*j+','+(255-20*j)+',80)';
            ctx.rotate(Math.PI*2/(24));
            ctx.beginPath();
            ctx.arc(30,60,6,Math.PI*2,true);
            ctx.fill();
        }
        ctx.translate(-5, -25);
        ctx.rotate(2.1);
        ctx.font = "40 Myriad Pro";
        ctx.fillStyle = "black";
        ctx.fillText("Galaxy Dev 2010", 20, 30);
    }
</script>
<body onload="draw();">
    <canvas id="canvas" height="200" width="400"></canvas>
</body>
```
• The drawing module of trackster makes a simple AJAX call to fetch data, and gets JSON data in return.
Data Structures

array_tree diagram for block size = 100

Root

min: 1.3
max: 5.4
mean: 3.0
sums: 8.9

Nodes with summary information

Data points

1.3 5.4 2.2 ...
0 1 2 3

0 100 10000

100 101 102 103
Data Structures

- summary_tree

FeatureTrack / ReadTrack Decision Diagram

- Feature/Read Track data request
- Query summary_tree
- Can we display details at this resolution?
- Yes
- No
- Query interval_index
- Trackster Browser
Data Structures

- interval_index
- part of bx_python
- fast region querying
- used for feature and read tracks
def get_data( self, chrom, start, end, **kwargs ):
    start, end = int(start), int(end)
    chrom = str(chrom)
    source = open( self.original_dataset.file_name )
    index = Indexes( self.converted_dataset.file_name )
    results = []

    for start, end, offset in index.find(chrom, start, end):
        source.seek(offset)
        feature = source.readline().split()
        payload = [ offset, start, end ]
        if "no_detail" not in kwargs:
            length = len(feature)
            if length >= 4:
                payload.append(feature[3]) # name
            if length >= 6: # strand
                payload.append(feature[5])
            if length >= 8:
                payload.append(int(feature[6]))
                payload.append(int(feature[7]))
            if length >= 12:
                block_sizes = [ int(n) for n in feature[10].split(',' ') if n != '' ]
                block_starts = [ int(n) for n in feature[11].split(',' ') if n != '' ]
                blocks = zip(block_sizes, block_starts)
                payload.append( [ (start + block[1], start + block[1] + block[0]) for block in blocks ] )
        results.append(payload)

    return results

interval_index data provider
Datatype Conversion

- Automatically run converters when needed
- Converters can now depend on other converters
- \text{BAM} \rightarrow \text{summary\_tree} \text{ requires } \text{BAM} \rightarrow \text{BAI} \text{ first}

```xml
<datatype extension="bam" type="galaxy.datatypes.binary:8am" mimetype="application/octet-stream" display_in_upload="true">
    <converter file="bam_to_bai.xml" target_datatype="bai"/>
    <converter file="bam_to_summary_tree_converter.xml" target_datatype="summary_tree" depends_on="bai"/>
    <display file="ucsc/bam.xml"/>
</datatype>
```
Custom Builds

- Define name, key, len info (chrom.sizes)
- Stored in the user preferences database table
1. Custom Builds link under “User”

2. Enter your build information

3. Set your datasets to the new build
Sharing

- Publish/share visualizations with others
- Embed in Galaxy Pages

Sunday, May 16, 2010
In Development

- More file formats
  - GTF, GFF
- More display modes
- Difference mode for feature/read tracks
Try it

- Set “enable_tracks = true” in universe cfg
- Check bitbucket wiki for instructions on downloading .len files for common builds
Conclusion

- trackster enables Galaxy users to analyze their data through visualization with an elegant interface
- Make and publish your own track browser for your own builds
Demo