

# Trackster

Visualization in Galaxy

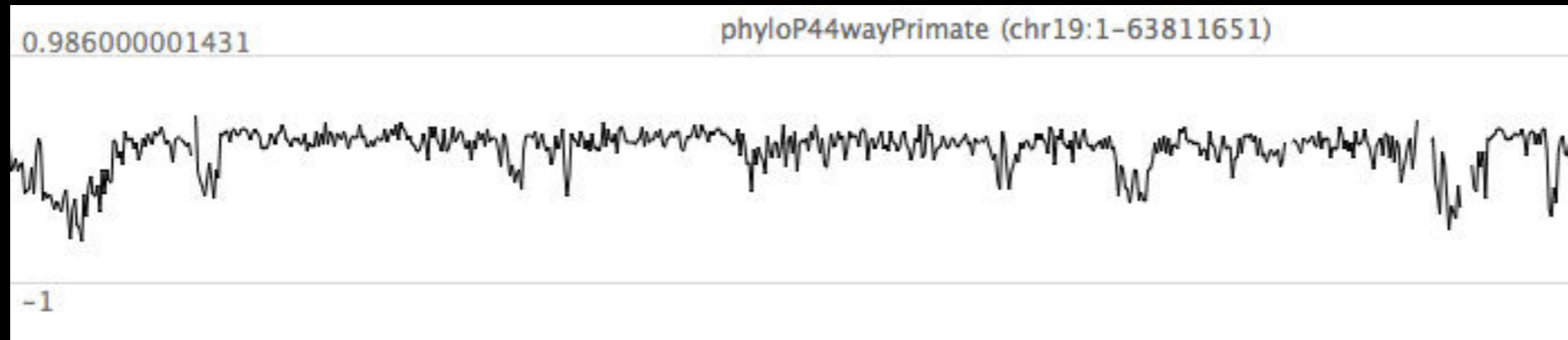
# What is trackster?

- 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

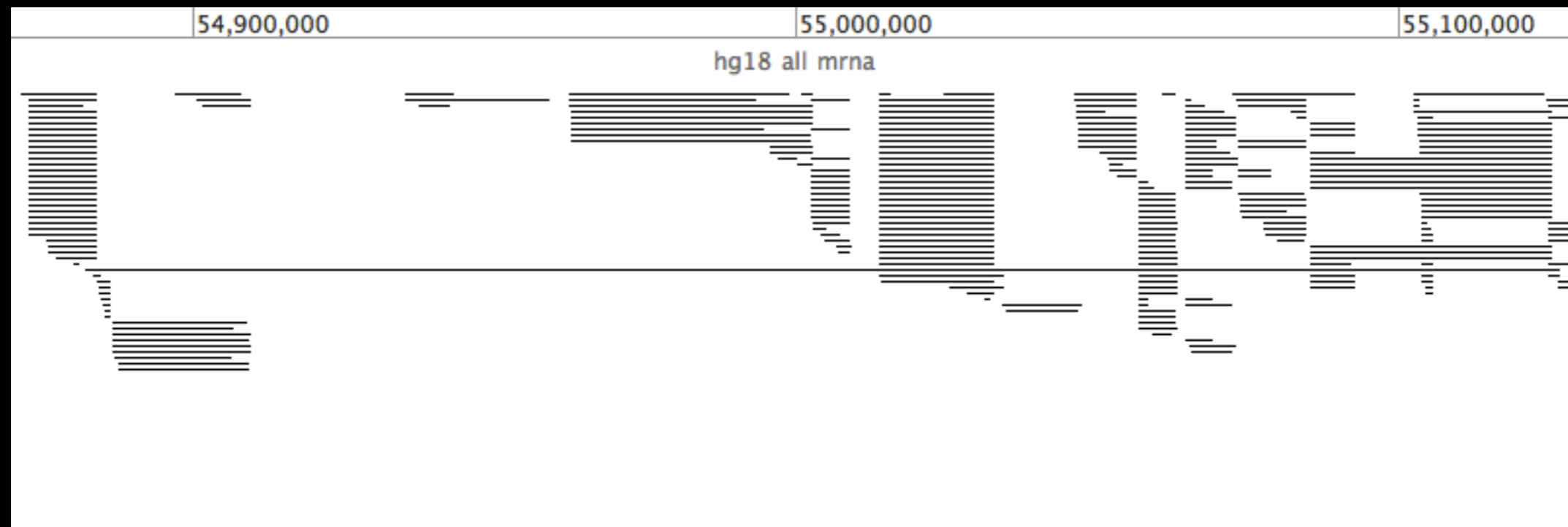


Intensity display (shades of gray)



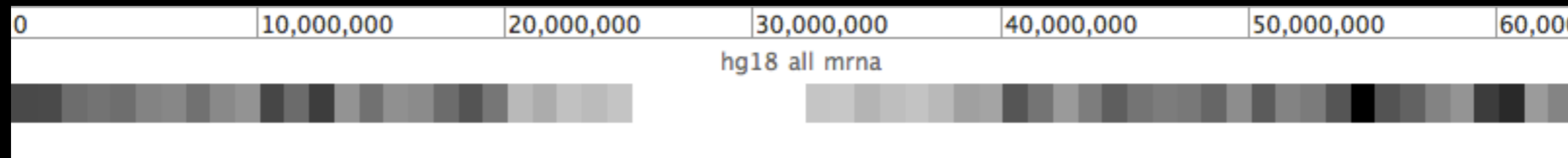
Filled line graph display

# Bed (Feature Track)

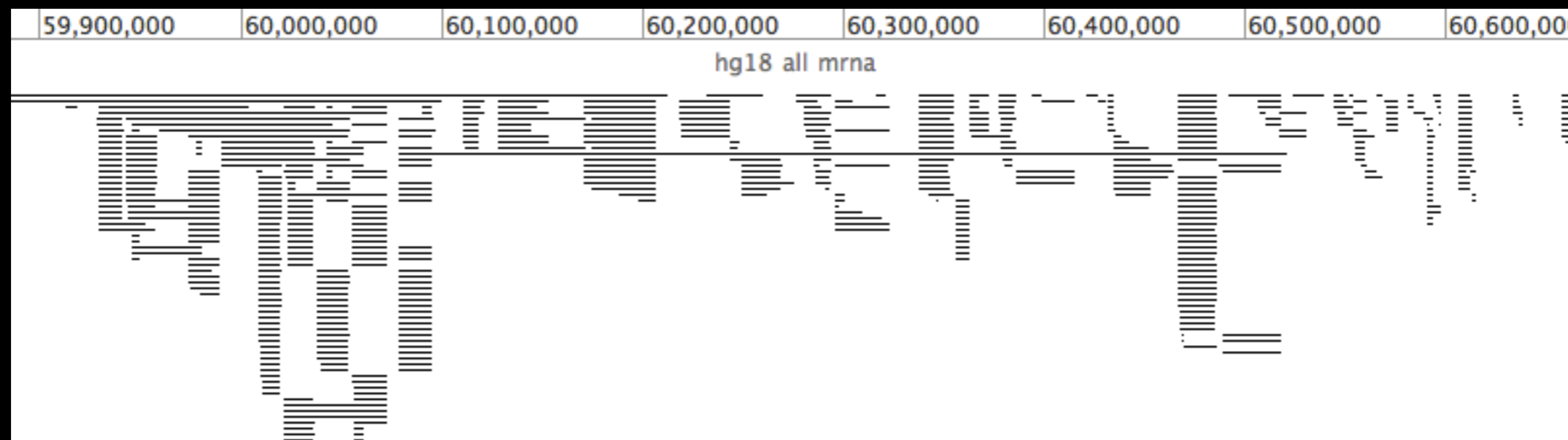


Snippet of hg18 all\_mrna feature track

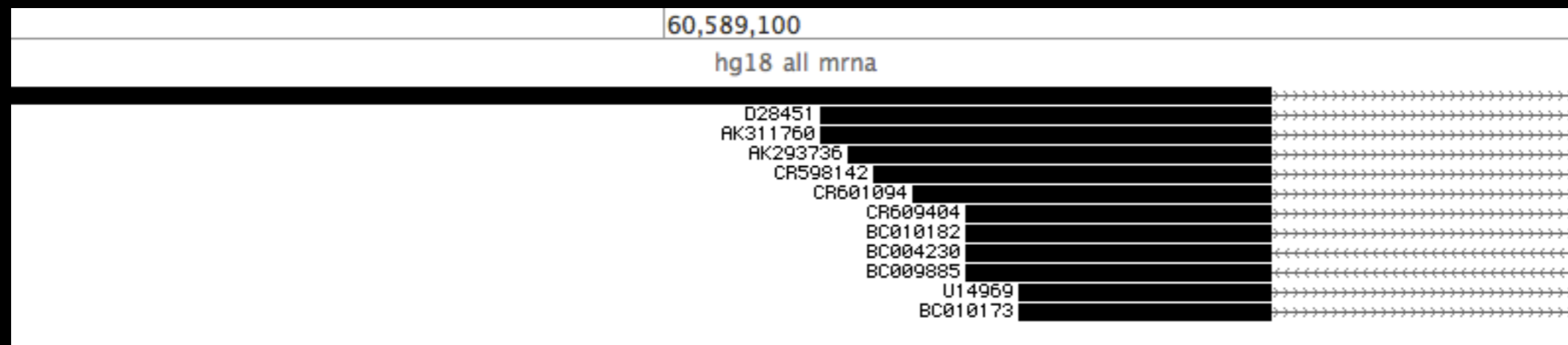
- 3 levels of detail: automatically adjusts based on what can fit on the screen



High level density display

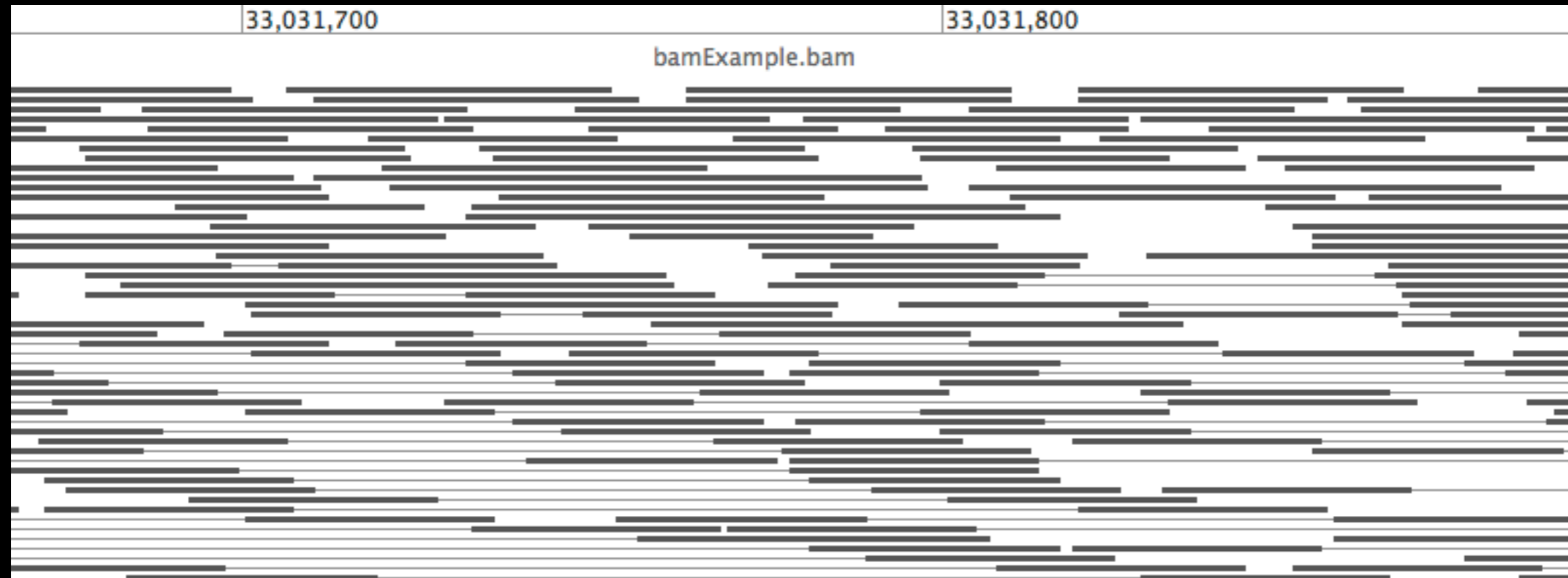


Feature display with no labels/detail

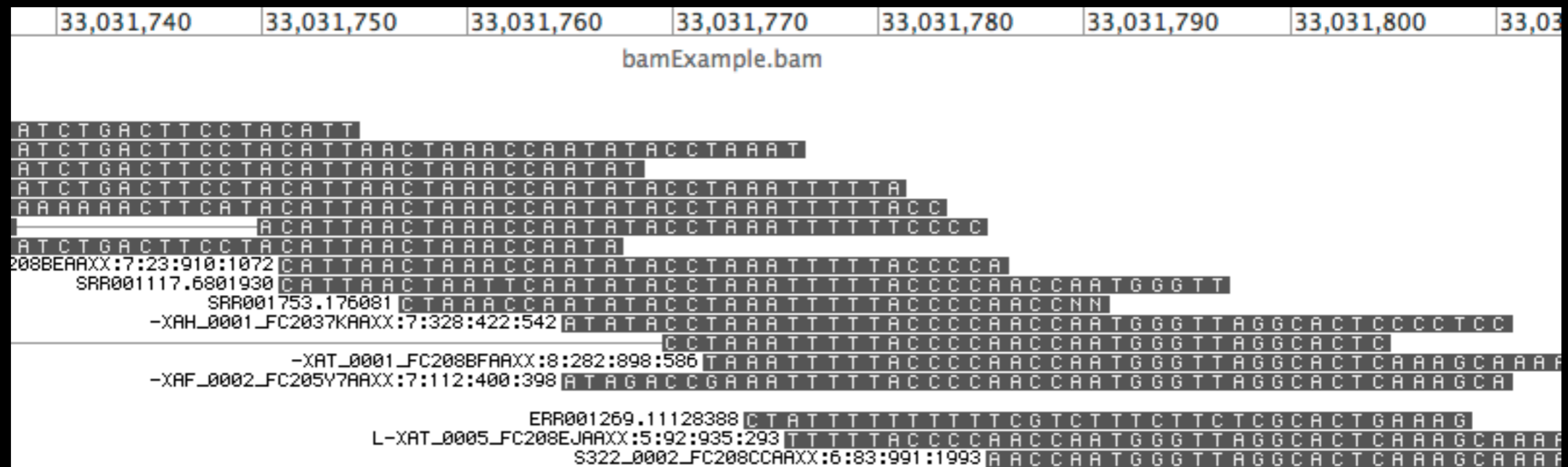


Feature display with labels, intron indicators, exon indicators





Paired-ends display with no labels



Paired-ends display with read labels and sequence



Galaxy Analyze Data Workflow Data Libraries Visualization Help User

hg18 all mrna  
No data for this chrom/contig.

phyloP44wayPrimate (chr19:1-63811651)  
No data for this chrom/contig.

Configuration

phyloP44wayPrimate (chr19:1-63811651)

Min value:  
-1

Max value:  
0.986000001431

Display mode:  
Filled

hg1  
Bl  
black

Text color:  
black

Show summary counts

Refresh Save Add Tracks

Select Chrom/Contig 0 : 0

- Add tracks, remove tracks, reorder tracks, resize tracks, edit track settings

# Well-Integrated




Analyze Data   Workflow   Data Libraries   **Visualization**   Help   User

**New Track Browser**

Browser name:






Reference genome build (dbkey):

- Human Apr. 2003 (hg15)
- Human July 2003 (hg16)
- Human Mar. 2006 (hg18)
- Human May 2004 (hg17)

**1: hg18 all mrna**   


15,905 regions, format: bed,  
database: hg18

[Info: UCSC Main on Human: all\\_mrna](#)  
[Visualize in Trackster](#)

| display at UCSC [main test](#) | BX [main](#)  
| view in [GeneTrack](#)

1. Chrom	2. Start	3. End	4. Name	5	6. St
chr19	11950	21966	AK308635	0	-
chr19	13111	13426	DQ786201	0	-
chr19	13114	21944	AK024448	0	-
chr19	15802	21944	BC067227	0	-
chr19	19402	20143	BC058830	0	+
chr19	20166	20971	DQ786265	0	+



# 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

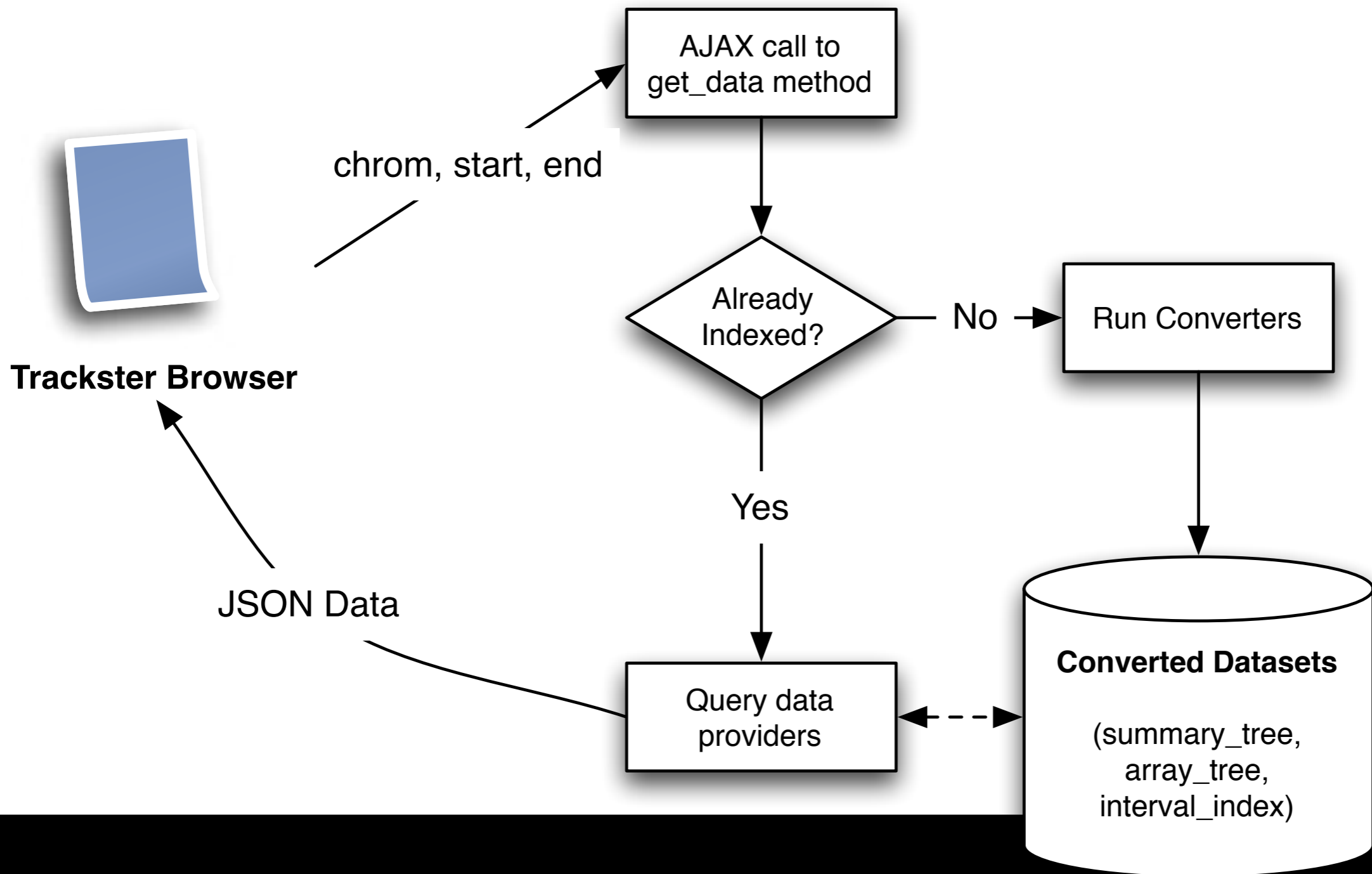
```
<script type="text/javascript" charset="utf-8">
  function draw() {
    var ctx = document.getElementById('canvas').getContext('2d');
    ctx.translate(75,75);

    for (var j = 0; j < 16; j++) {
      ctx.fillStyle = 'rgb('+(20*j)+'','+(255-20*j)+'',80)';
      ctx.rotate(Math.PI*2/(24));
      ctx.beginPath();
      ctx.arc(30,60,6,0,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>
```

Javascript code



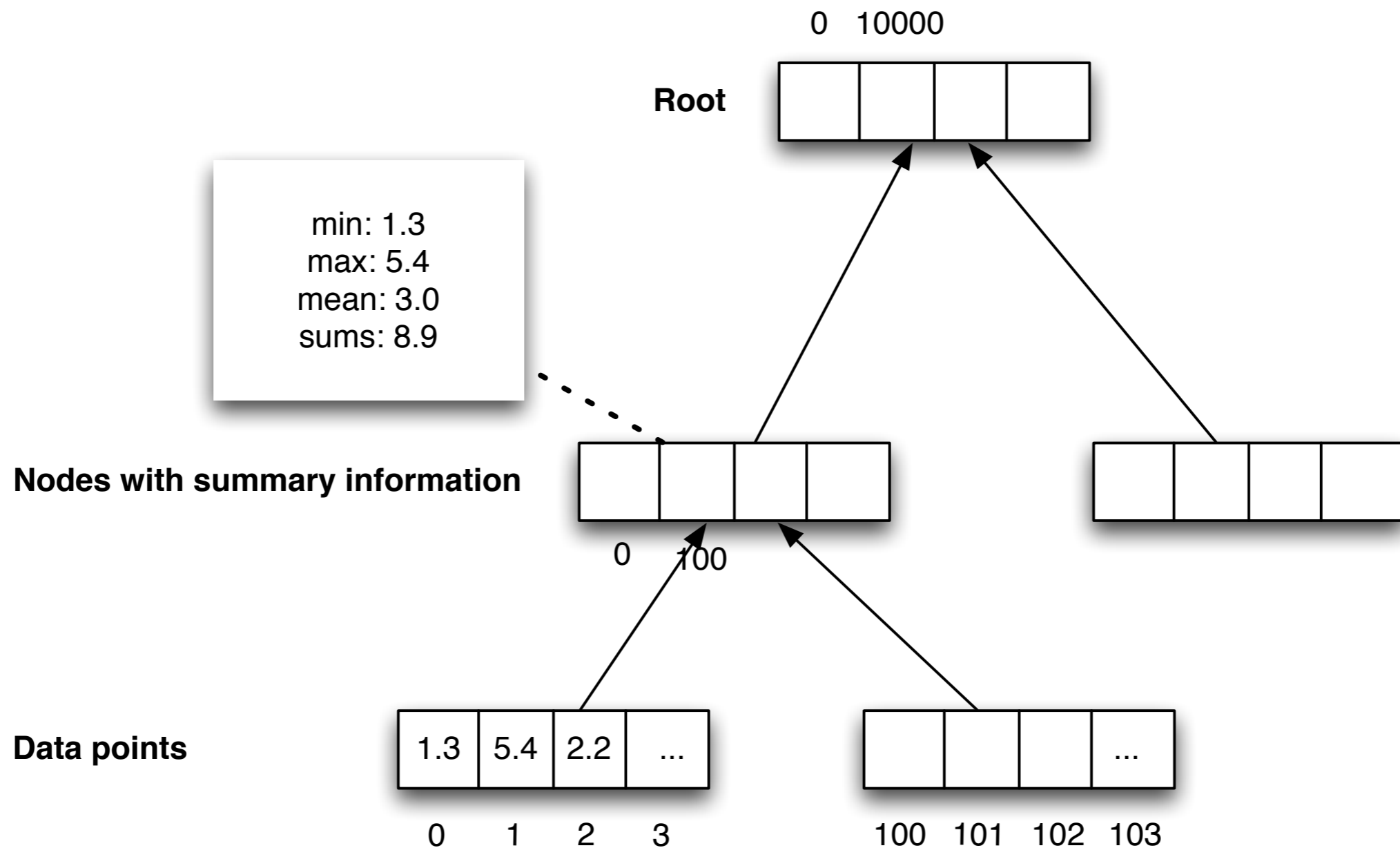
Result



- The drawing module of trackster makes a simple AJAX call to fetch data, and gets JSON data in return

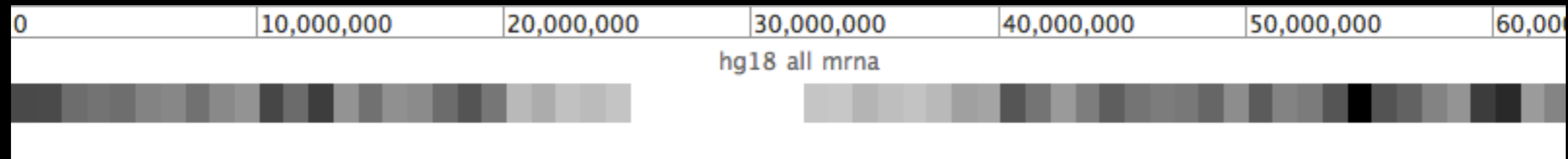
# Data Structures

array\_tree diagram for block size = 100

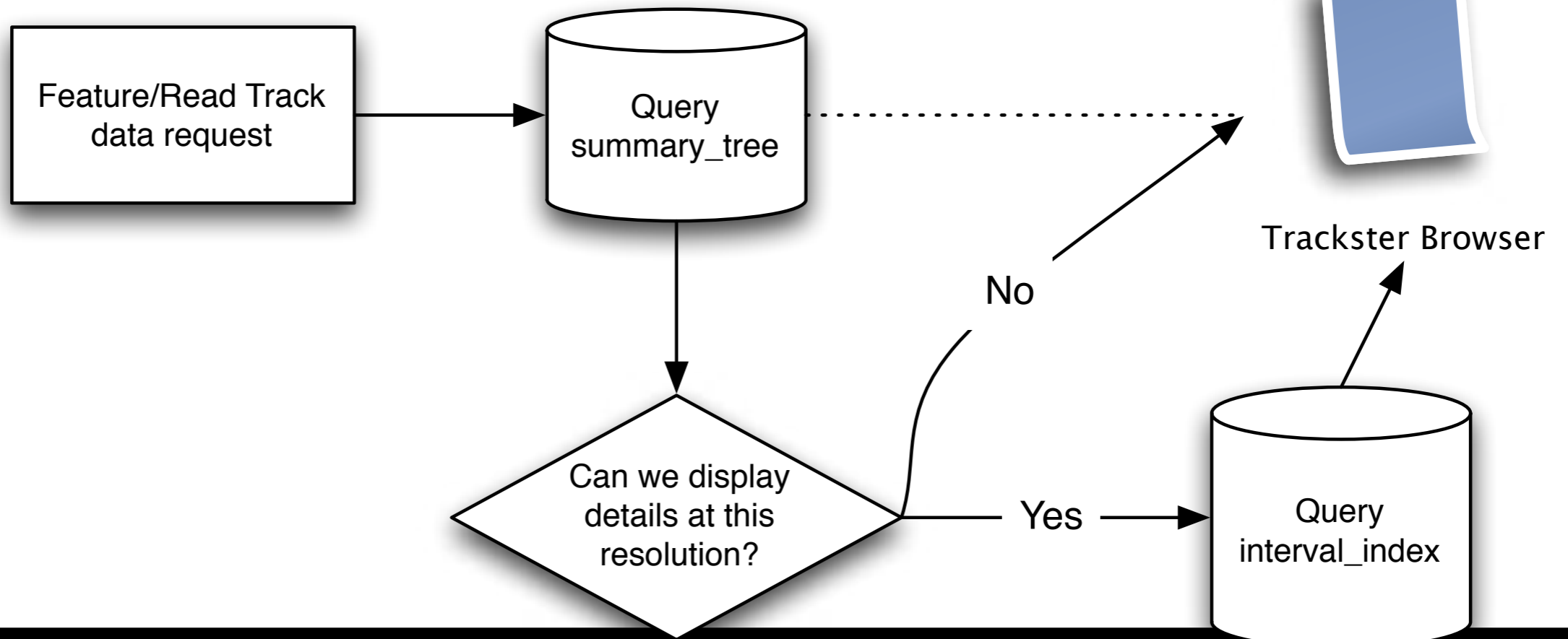


# Data Structures

- summary\_tree



## FeatureTrack / ReadTrack Decision Diagram





# 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

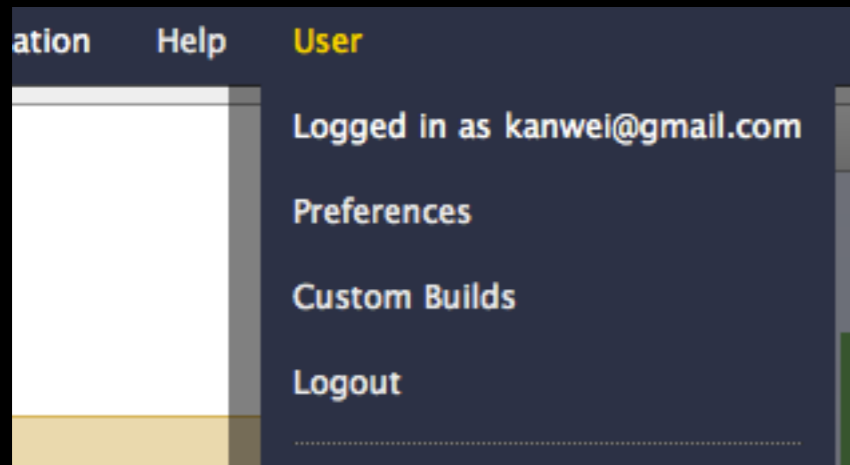
# Datatype Conversion

- Automatically run converters when needed
- Converters can now depend on other converters
  - BAM → summary\_tree requires BAM → BAI first

```
<datatype extension="bam" type="galaxy.datatypes.binary:Bam" 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

### Custom Database/Builds

You may specify your own database/builds here.

Name	Key	Chroms/Lengths	
ant	ant	<a href="#">37 entries</a>	<input type="button" value="Delete"/>

#### Add a Build

Name (eg: Human Chromosome):

Key (eg: hg18):

Chromosome Length file upload (.len file):  
 no file selected

Alternatively, paste length info:

### Edit Attributes

Name:

Info:

Annotation / Notes:

Add an annotation or notes to a dataset; annotations are available for all datasets.

Database/Build:

- ant (ant) [Custom]
- Bacillus anthracis str. 'Ames Ancestor' (10784)
- Bacillus anthracis str. Ames (baciAnth\_AMES)
- Bacillus anthracis str. Sterne (10878)
- Chimp Mar. 2006 (panTro2)
- Chimp Nov. 2003 (panTro1)
- Chloroflexus aurantiacus J-10-fl (59)

3. Set your datasets to the new build

# Sharing

The screenshot displays the Galaxy web interface. At the top, the 'Galaxy' logo is on the left, and navigation links for 'Analyze Data', 'Workflow', 'Data Libraries', 'Visualization', 'Admin', 'Help', and 'User' are on the right. Below the navigation bar, the page title is 'Published Visualizations | kanwei | bamExample'. The main content area shows a visualization titled 'Galaxy Visualization ' bamExample''. The visualization consists of two tracks: the top track is labeled 'all\_est' and shows a series of horizontal lines representing estimated values; the bottom track is labeled 'bamExample.bam' and shows a dense collection of horizontal bars representing genomic data. On the right side, there is a sidebar titled 'About this Visualization'. It includes a profile picture of the author 'kanwei', a section for 'Related Visualizations' with links to 'All published visualizations' and 'Published visualizations by kanwei', a 'Tags' section, and a 'Community: none' status. At the bottom of the sidebar, there is a 'Yours:' section with a small icon.

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

# 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