Visualization Framework

A system to allow easy & flexible creation, configuration, and inclusion of visualizations made by you

Visualization Framework

A system to allow easy & flexible creation, configuration, and inclusion of visualizations made by you

Lowest layer first

Visualization Framework

You know best what questions need to be answered or explored

Visualization Framework

You know best what questions need to be answered or explored

Does not do your visualization programming for you

Visualization Framework

You know best what questions need to be answered or explored

Does not do your visualization programming for you

Does your Galaxy programming for you

Visualization Framework

You know best what questions need to be answered or explored

Does not do your visualization programming for you

Does your Galaxy programming for you

(we want to do your Galaxy programming then get out of your way)

Visualization

Your code + user's data

Visualization

Your code:

Visualization

Your code:

a VisualizationRegistry entry

Visualization

Your code:

- a VisualizationRegistry entry
- a .mako template

Visualization

Your code:

a VisualizationRegistry entry

a .mako template

(that's it –
no commits, controller methods, etc.)

Visualization

Your code:

- a VisualizationRegistry entry
- a .mako template

The Registry

A familiar pattern

```
visualizations_conf.xml.sample
  <!--
    Sample registry shipped with Galaxy
  -->

universe_wsgi.ini
...
#visualizations_conf_path = visualizations_conf.xml
...

visualizations_conf.xml
  <!--
    Include your visualizations and customize your server here
  -->
```

The Registry

A familiar pattern

Controls how Galaxy interacts with your visualization

```
<visualization name="Your Visualization">
  <!-- ... -->
</visualization>
<visualization name="Your Visualization 2">
  <!-- ... -->
</visualization>
<visualization name="Your Visualization 3">
  <!-- ... -->
</visualization>
```

The Registry

How will the user start my visualization?

/visualization/show/myvis?data=lots

The Registry

How will the user start my visualization?

/visualization/show/myvis?data=lots

When will the link be rendered?

when a data source passes your tests

The Registry

When will the link be rendered?

Some source of data

```
for dataset in history:
    visualization_links = registry.get_visualizations( dataset )
```

The Registry

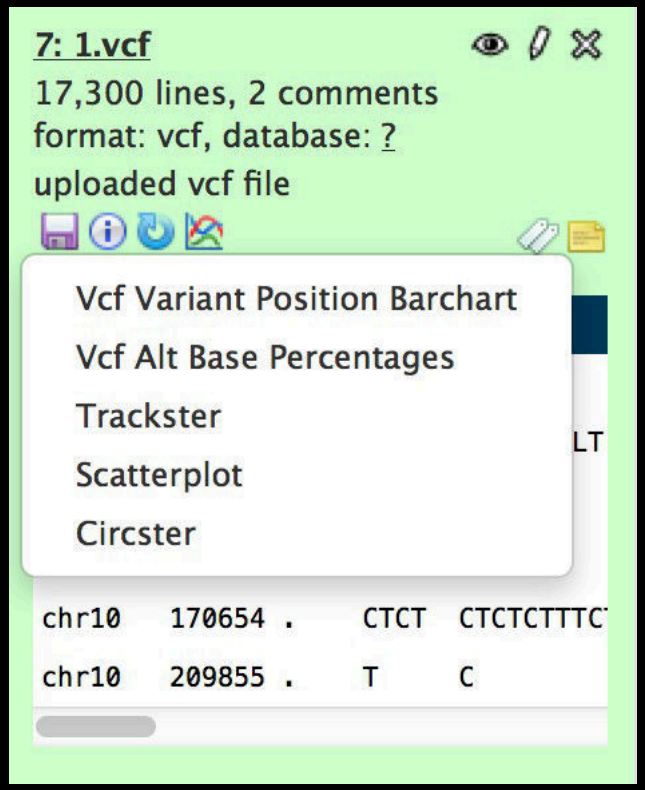
When will the link be rendered?

Some source of data

```
for dataset in history:
    visualization_links = registry.get_visualizations( dataset )
```

The Registry

When will the link be rendered?



The Registry

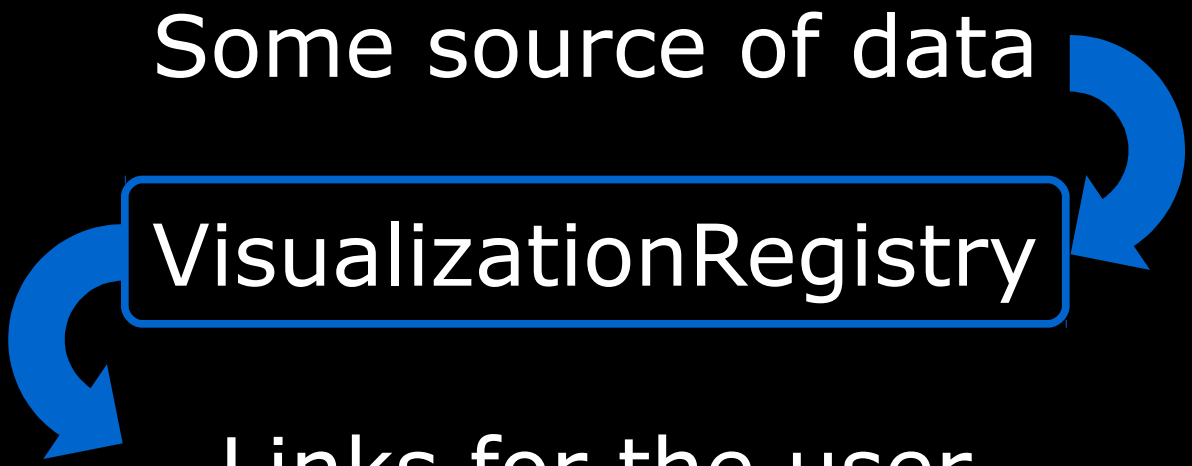
When will the link be rendered?

Some source of data

```
for dataset in history:
    visualization_links = registry.get_visualizations( dataset )

for dataset in library_folder:
    visualization_links = registry.get_visualizations( dataset )

for saved in user_visualizations:
    visualization_links = registry.get_visualizations( saved )

if user.is_admin():
    admin_vis_links = registry.get_visualizations( user )
```

The Registry

When will the link be rendered?

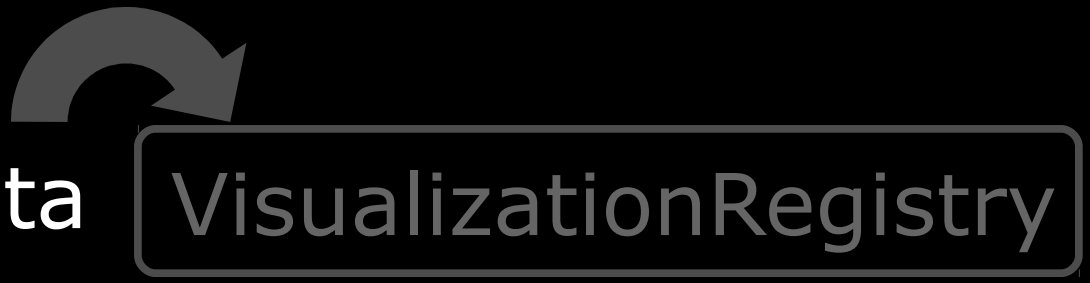


Outgoing Links

Outgoing Links

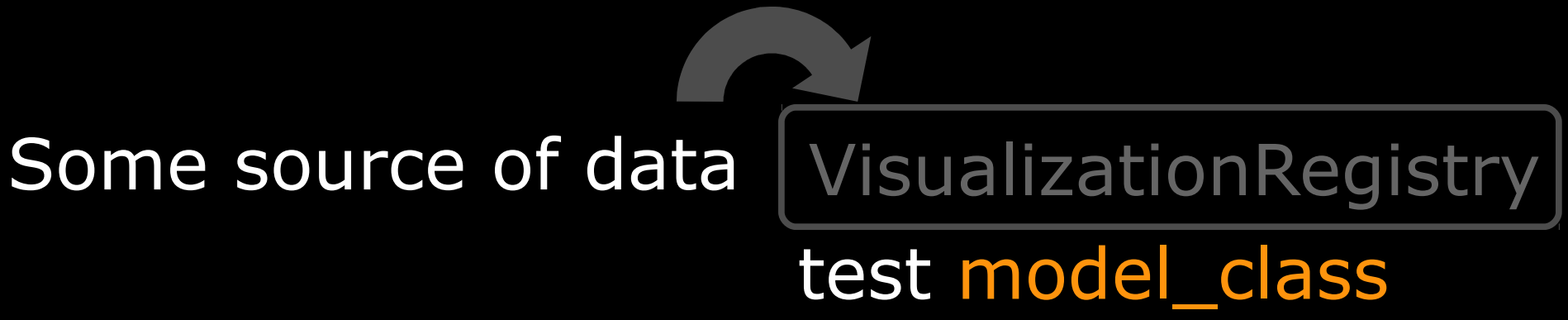

```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

Outgoing Links




```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

Outgoing Links




```
<model_class>HistoryDatasetAssociation</model_class>
```

Outgoing Links

Some source of data

VisualizationRegistry

test datatype

```
<model_class>HistoryDatasetAssociation</model_class>
<test type="isinstance" test_attr="datatype"
      result_type="datatype">tabular.Tabular</test>

HistoryDatasetAssociation( 1.vcf ).datatype == Tabular == Pass
```

Link for the user

Outgoing Links

Some source of data

VisualizationRegistry

test are OR'd

```
<model_class>HistoryDatasetAssociation</model_class>
<test type="isinstance" test_attr="datatype"
      result_type="datatype">binary.Bam</test>

HistoryDatasetAssociation( 1.vcf ).datatype == Bam == Fails

<test type="isinstance" test_attr="datatype"
result_type="datatype">tabular.Tabular</test>

HistoryDatasetAssociation( 1.vcf ).datatype == Tabular == Pass
```

Link for the user

Outgoing Links

Some source of data

VisualizationRegistry

test attr string

```
<visualization name="Coffee needed per day">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test test_attr="info"
            result_type="string">Grant submissions to do</test>
      <to_param param_attr="PI">someone_else_please</to_param>
    </data_source>
  </data_sources>
```

Link for the user

The Registry

How will the user start my visualization?

/visualization/show/myvis?data=lots

When will the link be rendered?

when a data source passes your tests

Can I pass more data through the link?

yes

The Registry


```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

Outgoing Links

Links for the user

to_param

```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

Outgoing Links

Links for the user

to_param

```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>(encoded)
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

Outgoing Links

Links for the user

to_param

```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
<!-- /visualization/show/scatterplot?dataset_id=01234567890ABCD -->
```

Outgoing Links

Links for the user

based on tests

```
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param assign="true">is_tab</to_param>
    </data_source>
<!-- link: /visualization/show/myvis?is_tab=true -->
```

Outgoing Links

Links for the user

based on tests

```
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param assign="true">is_tab</to_param>
    </data_source>
<!-- link: /visualization/show/myvis?is_tab=true -->
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">binary.Binary</test>
      <to_param assign="false">is_tab</to_param>
    </data_source>
<!-- link: /visualization/show/myvis?is_tab=false -->
```

Outgoing Links

Links for the user

1st passed : 1st rendered

```
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param assign="true">is_tab</to_param>
    </data_source>
<!-- rendered: /visualization/show/myvis?is_tab=true -->
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Vcf</test>
      <to_param assign="true">is_vcf</to_param>
    </data_source>
<!-- Never rendered -->
```

The Registry

How will the user start my visualization?

/visualization/show/myvis?data=lots

When will the link be rendered?

when a data source passes your tests

Can I pass more data through the link?

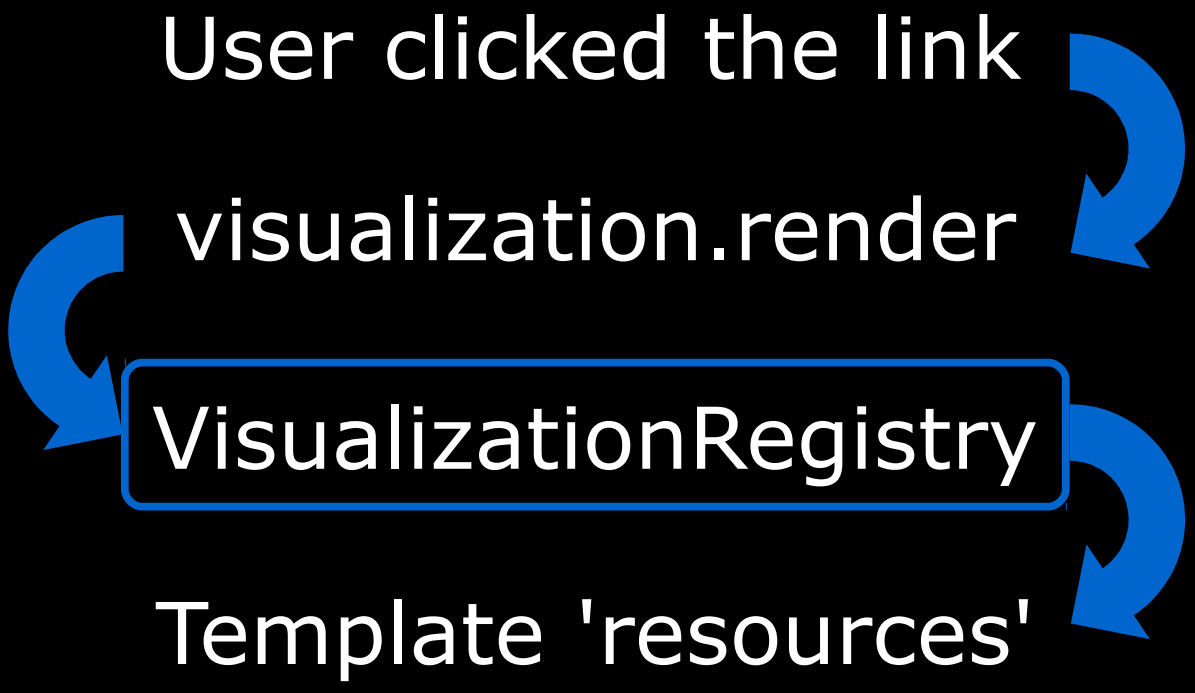
yes

How do I get at that data then?

it's parsed and sent to your template

The Registry

How do I get at that data then?



The Registry

User clicked the link visualization.render

```
def render( self, trans, visualization_name, embedded=None, **kwargs ):
    """
    Render the appropriate visualization template, parsing the
    query string `kwargs` into appropriate variables and resources
    (such as ORM models) based on this visualizations `param` data
    in visualizations_conf.xml.

    URL: /visualization/show/{visualization_name}
    """
    # validate name vs. registry
    registry = trans.app.visualizations_registry
    if not registry:
        raise HTTPNotFound( 'No visualization registry '
                          + '(possibly disabled in universe_wsgi.ini)' )
    if visualization_name not in registry.listings:
        raise HTTPNotFound( 'Unknown or invalid visualization: ' +
                            visualization_name )

    registry_listing = registry.listings[ visualization_name ]
```

The Registry

User clicked the link visualization.render

```
def render( self, trans, visualization_name, embedded=None, **kwargs ):
    """
    Render the appropriate visualization template, parsing the
    query string `kwargs` into appropriate variables and resources
    (such as ORM models) based on this visualizations `param` data
    in visualizations_conf.xml.

    URL: /visualization/show/{visualization_name}
    """
    # kwargs
    # validate name vs. registry
    registry = trans.app.visualizations_registry
    if not registry:
        raise HTTPNotFound( 'No visualization registry '
                          + '(possibly disabled in universe_wsgi.ini)' )
    if visualization_name not in registry.listings:
        raise HTTPNotFound( 'Unknown or invalid visualization: ' +
                            visualization_name )

    registry_listing = registry.listings[ visualization_name ]
```

The Registry

visualization.render VisualizationRegistry

```
def render( self, trans, visualization_name, embedded=None, **kwargs ):
    # ...
    registry_listing = registry.listings[ visualization_name ]
    # ...
    try:
        # convert query string to resources for template based
        # on registry config
        resources = registry.query_dict_to_resources( trans, self,
            visualization_name, kwargs )
        # ...
        returned = trans.fill_template( template_path,
            visualization_name=visualization_name,
            embedded=embedded, query_args=kwargs,
            shared_vars={}, **resources )
```

The Registry


```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

The Registry

template 'resources'

params

```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

The Registry

template 'resources'

to_param => param

```
<visualization name="scatterplot">
  <data_sources>
    <data_source>
      <model_class>HistoryDatasetAssociation</model_class>
      <test type="isinstance" test_attr="datatype"
            result_type="datatype">tabular.Tabular</test>
      <to_param param_attr="id">dataset_id</to_param>
    </data_source>
  </data_sources>
  <params>
    <param type="dataset" var_name_in_template="hda"
           required="true">dataset_id</param>
  </params>
  <template>scatterplot.mako</template>
</visualization>
```

The Registry


```
# some target object or other param
<HistoryDatasetAssociation>

# ... goes to the registry and gets serialized ...
<to_param param_attr="id">dataset_id</to_param>

# ... into a query string in a link that, when clicked, ...
/visualization/show/scatterplot?dataset_id=01234567890ABCD

# ... goes to the registry and gets un-serialized ...
<param type="dataset">dataset_id</param>

# ... into the target object for your template
<HistoryDatasetAssociation>
```

The Registry

template 'resources'

serialization

```
# some target object or other param
<HistoryDatasetAssociation>

# ... goes to the registry and gets serialized ...
<to_param param_attr="id">dataset_id</to_param>

# ... into a query string in a link that, when clicked, ...
/visualization/show/scatterplot?dataset_id=01234567890ABCD

# ... goes to the registry and gets un-serialized ...
<param type="dataset">dataset_id</param>

# ... into the target object for your template
<HistoryDatasetAssociation>
```

(yay! NAQSP)

The Registry

template 'resources'

template models

```
<param type="dataset" var_name_in_template="hda"
       required="true">dataset_id</param>

## ... into the target model! for your template
## <HistoryDatasetAssociation>
<p>Name: ${hda.name}</p>
## access to metadata
<p>Data: ${hda.metadata.data_lines}</p>
## access to 'eager loaded', related tables
<p>Job settings: <pre>${hda.creating_job.get_api_value()}</pre></p>
```

The Registry

template 'resources'

secure models

```
parsed_param = controller.get_dataset( trans, encoded_dataset_id,
    check_ownership=False, check_accessible=True )

returned = trans.show_error_message(
    "There was an error rendering the visualization. " +
    "Contact your Galaxy administrator if the problem persists." +
    "<br/>Details: " + str( exception ), use_panels=False )
```

The Registry

template 'resources'

many types

```
<param type="dataset">          ?d=<id>     HistoryDatasetAssociation
<param type="visualization">    ?v=<id>     Visualization(Revision)
<param type="hda_or_ldda">      ?d=<id>     HDA or LDDA
<param type="dbkey">            ?dbk=hg18   an existing genome build
<param type="json">             ?x={...}    JSON -> python structure
<param type="float">            ?x=1.3      parsed float
<param type="int">              ?x=-5       parsed int
<param type="bool">             ?x=true     True or False
... or lists of any of the above
<param csv="true"               ?d=<id1>,<id2>,<id3>
       type="dataset">                      a list of datasets [ ... ]
<param csv="true"               ?v=<id1>,<id2>,<id3>
       type="visualization">                a list of visualizations
<param csv="true"               ?ys=3,0,-9
       type="int">                          a list of parsed ints
```

The Registry

template 'resources'

multiple params

```
/visualization/show/complicated_visualization?
    allele_data=01234567890ABCD&
    ref_annotations=01234567890ABCE&
    ref_seq=hg18&
    chromosome=chr1&
    zoom_level=0.25&
    mark_positions=30033,35539,35542&
    quality_subgraph=F00FF0FFF8080ABE&
    view={start:25000,end:45000}
```

The Registry

How will the user start my visualization?

/visualization/show/myvis?data=lots

When will the link be rendered?

when a data source passes your tests

Can I pass more data through the link?

yes

How do I get at that data then?

(we want to do your Galaxy programming then get out of your way)

The Registry

How will the user start my visualization?

/visualization/show/myvis?data=lots

When will the link be rendered?

when a data source passes your tests

Can I pass more data through the link?

yes

How do I get at that data then?

it's parsed and sent to your template

Visualization

Your code:

- a VisualizationRegistry entry
- a .mako template

The Template

What libraries do I have to use?

(almost) any you'd like

The Template

What libraries do I have to use?

When to render?:

- Server side
- Client side
- A mix of both

The Template

What libraries do I have to use?

Server side – what to render?:

- SVG
- Images
- HTML

The Template

What libraries do I have to use?

Server side - rendering technology:

- Matplotlib
- Gnuplot + gnuplot
- Rpy2 + R
- Web Start + Java, etc.

The Template

What libraries do I have to use?

Server side:

- Matplotlib
- Gnuplot + gnuplot
- Rpy2 + R
- Web Start + Java

(must be installed/available)

The Template

What libraries do I have to use?

Server side SVG:

- Matplotlib -> SVG
- Gnuplot -> SVG
- SVGfig (installed)
- PySVG
- Pygal -> SVG
- (Mako) -> SVG

The Template

What libraries do I have to use?

(almost) any you'd like

Do I need to know JavaScript?

no (but js is good)

The Template

Do I need to know JavaScript?

Client side:

- SVG
- Canvas
- WebGL

The Template

Do I need to know JavaScript?

Client side:

- SVG
- Canvas
- WebGL
- Interactivity

The Template

Do I need to know JavaScript?

- Client side:
  - SVG
    - d3 (installed)
    - Raphaël
    - jQuery SVG
  - Canvas
  - WebGL
  - Interactivity

The Template

Do I need to know JavaScript?

- Client side:
  - SVG
  - Canvas
    - paper, fabric, easel, processingJS
  - WebGL
    - PhiloGL, xtk
  - Interactivity

The Template

Do I need to know JavaScript?

- Client side:
  - SVG
  - Canvas
  - WebGL
  - Interactivity (support/UI)
    - jQuery
    - backbone & Galaxy models/libs

The Template

What libraries do I have to use?

A mix of both:

- Matplotlib -> SVG + d3.js
- Gnuplot + JSON -> image + Canvas
- Mako -> HTML + jQuery

The Template

Do I need to know JavaScript?

A mix of both:

Matplotlib -> SVG + d3.js
Gnuplot + JSON -> image + Canvas
Mako -> HTML + jQuery

server = heavy lifting
client = interactivity

The Template

Do I need to know JavaScript?

A mix of both:

Matplotlib -> SVG + d3.js
Gnuplot + JSON -> image + Canvas
Mako -> HTML + jQuery

server = small,quick,light
client = expense on user side

The Template

What libraries do I have to use?

(almost) any you'd like

Do I need to know JavaScript?

no (but js is good)

(we want to do your Galaxy programming
then get out of your way)

The Template

What libraries do I have to use?

(almost) any you'd like

Do I need to know JavaScript?

no (but js is good)

How do I get the user's data?

(we want to do your Galaxy programming
then get out of your way)

Visualization

User's data + your code

Data

Possible sources of data:

- The database
- Dataset contents

Data

Possible sources of data:

- The database
  - resource APIs, the registry
- Dataset contents

Data

Possible sources of data:

- The database
- Dataset contents
  - bootstrapping into python/js
  - datasets API

Data

Possible sources of data:

- The database
- Dataset contents
- DataProviders

DataProviders

Get only the data you need
in the format you want

Data source + format desired + settings

DataProviders

Currently defined in datatypes

```
@dataproviders.decorators.has_dataproviders
class MyDataType( Tabular ):

    @dataproviders.decorators.dataprovider_factory(
            'mycolumns', ColumnarDataProvider.settings )
    # returns an iterator object
    def mycolumn_dataprovider( self, dataset, **settings ):
        return ColumnarDataProvider( dataset,
            indeces=[ 0, 6, 3, 3, 94 ],
            column_types=[ 'str', 'int', 'float', 'str' ],
            **settings )


---> [ [ 'KMFDM_0002', 8675309, 1.68, '1.68', None ],
       [ 'KMFDM_0003', 24,      0.02, '0.02', 'Maybe' ],
       ...
```

DataProviders

Currently defined in datatypes

```
@dataproviders.decorators.dataprovider_factory( 'mydict' )
def mydict_dataprovider( self, dataset, **settings ):
    return DictionaryDataProvider( dataset,
        indeces=[ 0, 6, 3, 3, 94 ],
        column_types=[ 'str', 'int', 'float', 'str' ],
        column_names=[ 'gene', 'count', 'qual',
                       'qual2', 'Notes' ],
        **settings )
```

```
---> [ { gene: 'KMFDM_0002',
         count: 8675309,
         qual:  1.68,
         qual2" '1.68',
         Notes: None
       }
       ...
```

DataProviders

Simple Filtering

(line) no blanks, no comments

DataProviders

Simple Filtering

Limiting

start at line 1000, return only 500

(pagination)

DataProviders

Simple Filtering

Limiting

More Filtering

- don't return lines matching regex
- (or inverse)

DataProviders

- Simple Filtering
- Limiting
- More Filtering
- Formatting
  - raw lines, arrays, dictionaries
  - specific columns
  - format with metadata

DataProviders

- Simple Filtering
- Limiting
- More Filtering
- Formatting
- Parsing
  - parse columns, key/values
  - parse using metadata.column_types

DataProviders

Multiple formats per source

```
@has_dataproviders
    @dataprovider_factory( 'column' )
    @dataprovider_factory( 'mycolumns' )
    @dataprovider_factory( 'genome-region' )
    @dataprovider_factory( 'interval-map' )
    @dataprovider_factory( 'seq-qal' )
```

Inherited formats

(no need to redefine)

DataProviders

(bootstrapping) into python:

```
bootstrapped_interval_list = list( dataset.datatype.dataprovider(
    dataset,
    'interval-map',
    limit=100,
    offset=200,
    regex_list=[ '^chr10' ], invert=True, ) )
# do something with the data
```

(bootstrapping) into js:

```
var interval_list =
    ${h.to_json_string( list( dataset.datatype.dataprovider(
                              dataset, 'interval-map' ) ) )};
    // do something with the data
```

DataProviders

(via the datasets API) into js:

```
var encodedId = '${query_args[ 'dataset_id' ]}',
    ajaxReq = jQuery.ajax( '/api/datasets/' + encodedId, {
        data : {
            // 'raw_data' tells the datasets API
            //    to return file contents instead of
            //    data about the dataset itself
            data_type: 'raw_data',
            // name the provider
            provider : 'interval-map'
        }
    });
    ajaxReq.then( function( intervalList ){
        // do something with the data
    });
```

Visualization

Your code + user's data

(we want to do your Galaxy programming
then get out of your way)

Persistence & Sharing

The Visualizations models and API

save
load
versioning
links by slugline

Persistence & Sharing

The Visualizations model and API

save
    "Here's a link to the mumps epidemiology..."
load
    "Got it. Good – so no one will be affected?"
versioning/'bookmarking'
    "Oh. I set the points to transparent. Try this one."
links by slugline
    "Well... that's probably more publishable."

In the Future

A work in progress
Multiple target data sources
Higher layers: UI elements
Higher layers: base templates
Embedding
Run tools & create new data
Combine & separate visualizations

Tusen takk! Thanks!

You, the Galaxy community!
UiO
Dave Clements
The Galaxy Team