

**GBrowse** recognizes the concept of a "group" of related features that are connected by dotted lines. The canonical example is a pair of ESTs that are related by being from the two ends of the same cDNA clone. However many feature databases, including the older Bio::DB::GFF database backends, do not allow for arbitrary hierarchical grouping. To work around this, you may specify a feature name-based regular expression that will be used to trigger grouping.

## The `group_pattern` Option

It works like this. Say you are working with EST feature pairs and they follow the nomenclature 501283.5 and 501283.3, where the suffix is "5" or "3" depending on whether the read was from the 5' or 3' ends of the insert. To group these pairs by a dotted line, specify the "group\_pattern" option in the appropriate track section:

```
group_pattern = /\.[53]$/
```

At render time, GBrowse will strip off this pattern from the names of all features in the EST track and group those that have a common base name. Hence 501283.5 and 501283.3 will be grouped together by a dotted line, because after the pattern is removed, they will share the same common name "501283".

This works for all embedded pattern, provided that stripping out the pattern results in related features sharing the same name. For example, if the convention were "est.for.501283" and "est.rev.501283", then this grouping pattern would have the desired effect:

```
group_pattern = /\.(for|rev)\./
```

Don't forget to escape regular expression meta-characters and to consider the various ways in which the regular expression might break. It is entirely possible to create an invalid regular expression, in which case gbrowse will crash until you comment out the offending option.

## The `group_on` Option

The Bio::DB::SeqFeature::Store database backend has the opposite problem. It allows features to be hierarchically grouped to an unlimited extent. However, this is problematic with quantitative data such as tiling array data in which a feature might extend from one chromosome to another. For performance reasons, you may not wish to read the entire chromosome-length feature into memory in order to display a small portion of it. The **group\_on** option allows you to create a database in which each quantitative data point is a separate feature, but at run time related features are grouped together to create a composite feature that can be graphed. The syntax is:

```
group_on = grouping_field
```

Where *grouping\_field* is one of the feature's methods. Typically, you will group on the **display\_name** field in order to group features together by their human-readable name. In this case, the **group\_on** field would read:

```
group_on = display_name
```

For very dense quantitative data, you should use [wiggly format](#). Please see [GBrowse/Uploading Wiggle Tracks](#) for instructions.