ExonMutViz Code Project: Sequence Coverage Visualization

Purpose

This package is designed to visualize overlay the mutational probes identified in the NGS experiments specific to exonic region of genes in a given sample.

Overview

Design and implement code that produces one or more plots of sequence coverage for each gene region covered by the **ExonMutViz** assay. The code has capability to automatically generate the plot(s) from any text file in a pre-defined format (see *Input Data*, below). The scope of the plot(s) is for individual samples – i.e., an input file with data from five unique samples will produce five distinct sets of plots.

File Formats

Reference Gene Models. The genes in a file should be tab-delimited text file containing the models for each gene covered by the **ExonMutViz** assay. Although the **ExonMutViz** assay deliberately covers only a subset of these exons, the assay does cover at least one exon for each of the genes included in the file. The reference file lists the model data under the following headers:

```
gene, transcript, ccdsId, chrom, exonStart, exonEnd, exonNum
```

Input Data. The input file should be tab-delimited text file containing data under the following headers:

```
sample, geneRegion, gene, transcript, ccdsId, chrom, txStrand*,
readStrand**, hg19Start, hg19End, templates
```

* the orientation of the *transcript* relative to the genomic reference strand
** the orientation of the *sequence read* relative to the genomic reference strand

Output. The package automatically generate separate documents for each sample included in the input file. The plot(s) within each document should fit on one 8.5” by 11” page. Each document should include the name of the sample in the filename and be in PDF or Microsoft Word format.

Visualization

Present the sequence coverage data in a clear and logical manner. Sequence coverage of each **ExonMutViz** `geneRegion` should be plotted in relation to their respective gene model in a manner that allows the identity of the `geneRegion` to be identified (e.g., descriptive text or color labels). Any `geneRegion` within a sample with poor representation (>0 but <200 templates) should be highlighted or otherwise clearly distinguishable from those with adequate coverage (≥200 templates). Accounting for transcript and sequence read orientation is considered optional.

Acceptable Implementation Languages

Python 3 or Python 2.7