geneCNV Documentation

Release 0.0.2

Velina Kozareva, Nigel Delaney

Feb 07, 2018

Contents

GeneCNV is a command-line software package and Python library designed to use short-read targeted sequencing data to identify copy number across targets. It is intended for analysis across a subset of genes using parameters derived from a predefined set of reference (normal) samples.

CHAPTER 1

Introduction

This introduction will demonstrate usage of the main commands required to run geneCNV and step you through an example analysis using data provided in the package.

1.1 Install geneCNV

Go to GitHub to download the source code and see full installation instructions.

Briefly, to install the package and any unsatisfied dependencies:

```
git clone https://github.com/GenePeeks/geneCNV.git
cd geneCNV
pip install -r requirements.txt
python setup.py install
```

1.2 Get coverage counts

To get started, generate coverage counts across relevant targets and samples using the `create-matrix` command.

You must first provide a BED file of relevant targets in this format:

```
X   32867834    32867947    Ex3 DMD
X   33038245    33038327    Ex2 DMD
X   33229388    33229673    Ex1 DMD
```

The first four fields (chromosome, start position, end position, label) are required, while the fifth is optional. An example BED file for the DMD gene (`example_dmd_baseline.bed`) is provided in the `test_data` directory of the package.

Note: Baseline target specification If you are using baseline targets, these targets should have `Baseline` as part of the target label.

In addition to a BED file, you must provide a text file of paths to the sample BAM files in this format:

```
/path/to/file1.bam
/path/to/file2.bam
```

An example `create-matrix` command looks like:

```
genecnv create-matrix test_data/example_dmd_baseline.bed training_samples.fofn \
training_sample_coverage.csv --targetArgfile dmd_baseline_targets.pickle
```

Serialized target/argument files (targetArgfiles) can be optionally produced with this command. You only need to produce a target/argument file once for a specific set of targets. An example output CSV for this command is provided in `test_data`. This can be used to run the subsequent `train-model` command.

1.3 Train the model with normal samples

Next you'll estimate the model hyperparameters (train the model) using the samples included in the coverage count matrix. These should be "normal" samples without known CNVs in the (non-baseline) targets of interest.

Note: How should I select "normal" samples?

- If you are unsure which of your samples should be considered "normal", you can generate a coverage matrix and then examine the coverage distributions across samples using a dataframe analysis tool. You can then produce another coverage matrix after removing any noisy or problematic samples.
- **It is important that all sequencing data used for analysis with geneCNV has been produced with the same sequencing pipeline.**

To train the model, run the following:

```
genecnv train-model dmd_baseline_targets.pickle test_data/training_sample_coverage.
↪csv \
dmd_baseline_params.pickle --use_baseline_sum
```

Baseline autosomal targets are used to identify absolute copy number when no CNVs are present, and help provide more accurate results overall. Including baseline targets can also allow you to identify the sex of a sample when targets on the X chromosome are being tested. Baseline targets are not analyzed for copy number and are assumed to have copy number of 2.

If you are using a large number of baseline targets (>20), it's recommended to use the optional `--use_baseline_sum` argument when calling `train-model`. This reduces the total number of baseline targets to one during training.

1.4 Evaluate samples for CNVs

Once parameters have been estimated from an appropriate set of training samples, they can be used to perform copy number analysis for the relevant targets on a test sample with the `evaluate-sample` command. Here you can pass simply a sample BAM file or a coverage matrix CSV (generated using the same targets).

To evaluate the first test sample in the file `test_data/test_female_sample_coverage.csv` use the following command:

```
genecnv evaluate-sample test_data/test_female_sample_coverage.csv dmd_baseline_params.
↪pickle \
normal_female_results
```

This command will produce three output files with the provided prefix, `normal_female_results.txt`, which provides the posterior probabilities and copy numbers for all relevant targets, `normal_female_results_summary.txt` which provides a summary of any CNVs detected, and `normal_female_results.pdf`, which provides a visualization of the copy numbers and posterior probabilities across targets.

Depending on the number of total targets and MCMC iterations needed for convergence, the sample evaluation may take up to 10-12 minutes to complete. By default it takes advantage of multiple cores, but this can be turned off with the option `--use_single_process`.

For further detail and options, see the *CLI documentation*.

CHAPTER 2

Command Line Interface

2.1 create-matrix

2.2 train-model

2.3 evaluate-sample

CHAPTER 3

Indices and tables

- genindex
- modindex
- search