CONTRA User Guide, version 2.0

Richard Lupat
Jason Li
Peter Maccallum Cancer Centre, East Melbourne, Victoria
Richard.Lupat@petermac.org
Jason.Li@petermac.org

17 October 2011

Contents

1. Introduction

CONTRA is a tool for copy number variation (CNV) detection for targeted resequencing data such as those from whole-exome capture data. CONTRA calls copy number gains and losses for each target region with key strategies include the use of base-level log-ratios to remove GC-content bias,

correction for an imbalanced library size effect on log-ratios, and the estimation of log-ratio variations via binning and interpolation.. It takes standard alignment formats (BAM/SAM) and output in variant call format (VCF 4.0) for easy integration with other next generation sequencing analysis package.

2. Requirements

To run CONTRA, you need the following programs:

1. Python 2.6+
   Most of the scripts for CONTRA are written in Python. It requires version 2.6 in order to use the multiprocessing module. (http://www.python.org/)
2. R
   http://www.r-project.org/
3. BEDTools (Included in CONTRA package. See Installation Guide)
   The original source of the BEDTools can be found in http://code.google.com/p/bedtools/
4. SAMtools
   http://samtools.sourceforge.net/
5. [Optional] DNACopy (R-library that will be used for predicting large CNV)
   http://www.bioconductor.org/packages/2.8/bioc/html/DNAcopy.html

3. Installation Guide

Download CONTRA tarball and decompress it with the following command:

```
tar –xvzf contra.<version>.tar.gz
```

For users who DO NOT have BEDTools installed, follow these steps:

```
cd contra.<version>
tar -zxvf BEDTools.tar.gz
cd BEDTools
make clean
make all
sudo cp bin/* /usr/local/bin/
```

4. Format of Input Files

The analysis itself requires several files as following:

1. BAM or SAM files for the test and control samples. User can also provide a baseline file (in BED format) as the control sample. Please refer to “Section 9: Baseline” for example how to use the baseline script.

2. Target File in BED format [http://genome.ucsc.edu/FAQ/FAQformat])
   A tab-delimited file specifying the target/capture regions. Four columns must be specified: Chromosome, ChromStart, ChromEnd and Name. Name can be anything, and is used for annotation only. If there is no name specified, user will see "unknown" for the gene name/symbol. No header line is expected.

   The ChromStart is 0 base (the first base in a chromosome is numbered 0)
   The ChromEnd base is not included in the display of the feature.
   For example, chromStart= 0 and chromEnd = 100 is implying the first 100 bases of a chromosome (0-99).

   Below is the example of first few lines of a target.BED file:

1	*357512*	*357632*	*uc001aaw.1*
1	*357571*	*357691*	*uc001aaw.1*
1	*357633*	*357753*	*uc001aaw.1*
1	*357694*	*357814*	*uc001aaw.1*
1	*357756*	*357876*	*uc001aaw.1*
1	*357816*	*357936*	*uc001aaw.1*

3. Fasta file for reference genome [e.g. human_g1k_v37.fasta]

5. CONTRA Workflow

Step-1
CONTRA takes all the required files from the user [target, test BAM/SAM, control BAM/SAM/baseline file, fasta], and do all the pre-processing steps to ensure all the input files are compatible with the CONTRA script (such as files need to be in sorted order). If maxRegionSize is specified, large regions will be broken down into smaller regions depending on user specified parameters.

Step-2
Short-read alignment information (BAM or SAM formats) from a test and a control sample is converted into read count per base pair for a list of target regions (BED format) using BEDTools. Target regions with too few reads in the control sample (by default, < 10 base pairs with read count > 10) are excluded from further analysis. The remaining read counts are then scaled based on the geometric mean of the total read counts of the two samples

Step-3
For each target region, a set of base-level log-ratios between test and control is calculated based on the scaled read counts, the mean of which is used to estimate the region's log-ratio. Library size bias is then removed based on a linear relationship between log-ratio and log-coverage estimated from the data. The regions are then binned based on their similarity in log-coverage.

Step-4
Significance is then computed for each region and is adjusted to reduce false discovery rates. Results are reported with other details in either tab-delimited or the VCF4.0 format (Variant Call Format; see www.1000genomes.org).

Step-5
If large deletion option is specified, circular binary segmentation will be performed on region log-ratios, using different parameters to achieve different resolutions of segmentation. Segmentation results from different resolutions are combined to make the final call.

6. Output Format

6.1. VCF

VCF4.0 format

Note:
QUAL = 10Log_{10} Adjusted p-value
Other columns are described in the VCF file header.

The details explanation for each column can be found in
[http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/VCF%20%28Variant%20Call%20Format%29%20version%204.0/encoding-structural-variants]

6.2. Tab-delimited

Three tab-delimited files will be generated:

1. Full details of the analysis, excluding target regions that do not pass the minimum read depth & number of bases thresholds (See Step 2 in CONTRA workflow)
2. Including only target regions that pass the p-value threshold
3. If largeDeletion option is specified, a summary of large CNV prediction with significant copy number changes will be presented in the tab-delimited file (Step 5 in CONTRA workflow)

The tab-delimited output for the contains:
target region ID, exon number, gene symbol, chromosome, original start coordinate, original end coordinate, mean, standard deviation and median of the log-ratios, number of bases included in the analysis for that target region, p-value, adjusted p-value, gain/loss, average test sample's scaled read depth, average control sample's scaled read depth, average test sample's original read depth, average control sample's original read depth, minimum and maximum log ratios on that target regions and the bin number.

6.3. Plot

If --plot option is specified when running the code, plot(s) for the distribution of log ratios will be included in the output folder.

7. Examples

7.1. Example 1: Running CONTRA with default optional parameters

We assume we have a target file *target_test.BED*, two BAM files *test_sample.BAM* and *control_sample.BAM* and a reference file *human_ref.fasta*. Our intended output folder is in *~/ContraTest/sampleName/* .

To run the analysis on this sample, the command line argument is:

```
python contra.py --target target_test.BED --test test_sample.BAM
--control control_sample.BAM --fasta human_ref.fasta
--outfolder ~/ContraTest/sampleName/
```

This will create a folder call *sampleName* inside *~/ContraTest/*. Inside the folder, there will be two subfolders, plot and table. The table folder will contain the VCF file and the analysis' details.

Note: CONTRA will always attempt to create the folder specified last (i.e. *sampleName* in this example). If the folder exists, there will be an error message when running the script. It is setup this way to avoid the data in the existing folder being overwrite. **However,** it will not create the parents folder (i.e. *ContraTest* in this example) with the assumption this folder has already existed. An attempt to put the result folders in a directory that has not been created will generate an error message.

7.2. Example 2: Running CONTRA with other optional parameters

We assume we have a target file *target_test.BED*, two BAM files *test_sample.BAM* and *control_sample.BAM* and a reference file *human_ref.fasta*. Our intended output folder is in *~/ContraTest/sampleName/* .

The options we want to change for this example:

- Number of bins to : 1,5,10,15 and 20 bins
- Minimum read depth : 5
- Minimum number of bases : 20
- SampleName : sample123
- Remove multi mapped reads

To run the analysis on this sample, the command line argument is:

```
python contra.py --target target_test.BED --test test_sample.BAM
--control control_sample.BAM --fasta human_ref.fasta --outfolder
~/ContraTest/sampleName/ --numBin 1,5,10,15,20 --minReadDepth 5
--minNBases 20 --sampleName sample123 --nomultimapped
```

Note1: The --nomultimapped option will use samtools to filter out alignment with mapping quality = 0.

Note2: With the --sampleName option, all the results' filename will be appended with the sample name in front of the default filename.

8. List of CONTRA parameters

8.1. Required

-t, --target	Target region definition file [BED format]
-s, --test	Alignment file for the test sample [BAM/SAM]
-c, --control	Alignment file for the control sample [BAM/SAM/BED* – baseline file] *--bed option has to be supplied for control with baseline file.
-f, --fasta	Reference genome [FASTA]
-o, --outFolder	the folder name (and its path) to store the output of the analysis (this new folder will be created – error message occur if the folder exists)

8.2. Optional

--numBin	Numbers of bins to group the regions. User can specify multiple experiments with different numbers of bins (comma separated). [Default: 20]
--minReadDepth	The threshold for minimum read depth for each bases (see Step 2 in CONTRA workflow) [Default: 10]
--minNBases	The threshold for minimum number of bases for each target regions (see Step 2 in CONTRA workflow) [Default: 10]
--sam	If the specified test and control samples are in SAM format. [Default: False] (It will always take BAM samples as default)
--bed	If specified, control will be a baseline file in BED format. [Default: False] *Please refer to the Baseline Script section for instruction how to create baseline files from set of BAMfiles. A set of baseline files from different platform has also been provided in the CONTRA download page.
--pval	The p-value threshold for filtering. [Default: 0.05]
--sampleName	The name to be appended to the front of the default output name. By default, there will be nothing appended. [Default:””]
--nomultimapped	The option to remove multi-mapped reads (using SAMtools with mapping quality > 0). [Default: FALSE]
-p, --plot	If specified, plots of log-ratio distribution for each bin will be included in the output folder [default: FALSE]

--minExon	Minimum number of exons in one bin (if less than this number, bin that contains small number of exons will be merged to the adjacent bins) [Default : 2000]
--minControlRdForCall	Minimum Control ReadDepth for call [Default: 5]
--minTestRdForCall	Minimum Test ReadDepth for call [Default: 0]
--minAvgForCall	Minimum average coverage for call [Default: 20]
--maxRegionSize	Maximum region size in target region (for breaking large regions into smaller regions. By default, maxRegionSize=0 means no breakdown). [Default : 0]
--targetRegionSize	Target region size for breakdown (if maxRegionSize is non-zero) [Default: 200]
-l, --largeDeletion	If specified, CONTRA will run large deletion analysis (CBS). User must have DNAcopy R-library installed to run the analysis. [False]
--smallSegment	CBS segment size for calling large variations [Default : 1]
--largeSegment	CBS segment size for calling large variations [Default : 25]
--lrCallStart	Log ratios start range that will be used to call CNV [Default : -0.3]
--lrCallEnd	Log ratios end range that will be used to call CNV [Default : 0.3]
--passSize	Size of exons that passed the p-value threshold compare to the original exons size [Default: 0.5]

9. Baseline Script

Creating a baseline control from multiple samples is can be useful when a matched control is not available. In the CONTRA download page, we have provided several baseline files for some of the platforms that we have tried. Alternatively, the “baseline.py” script that comes with CONTRA can be used to generate a custom baseline file.

9.1. List of parameters

-t, --target	Target region definition file [REQUIRED] [BED format]
-f, --files	Files to be converted to baselines [REQUIRED] [BAM]
-o, --output	Output folder [REQUIRED]
-c, --trim	Portion of outliers to be removed before calculating average [Default: 0.2]

-n, --name Output baseline name [Default: baseline]

9.2. Example

We assume we have a target file *target_test.BED* and four BAM files that we want to turn into baseline file *test1.BAM*, *test2.BAM* , *test3.BAM* and *test4.BAM*. Our intended output folder is in *~/Baseline/sampleBaseline/* with the final baseline file name *baseline_test*.

To run the baseline script on this sample, the command line argument is:

`python baseline.py --target` *target_test.BED* `--files` *test1.BAM test2.BAM test3.BAM test4.BAM* `--output` *~/Baseline/sampleBaseline/* `--name` *baseline_test*