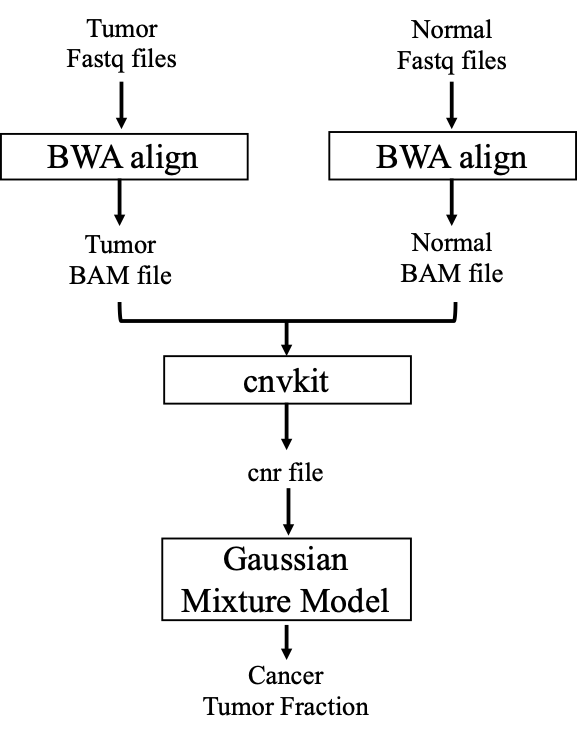
**Tumor fraction inference from CNV**

**General purpose**

This script is used to calculate tumor fraction from copy number variation profile of bulk tumor tissue sequencing data.



**Required input**

1. Sample name (for output)
2. CNVkit output ( *\*.cnr*  ) file.

The CNVkit package, including its documentation, could be found at: <https://cnvkit.readthedocs.io/en/stable/>

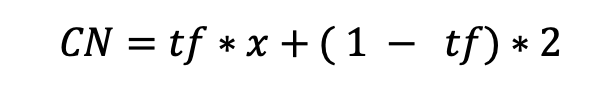
**Execution**

In shell: use Rscript 2.Bulk.Panel.Sequencing.Tumor.fraction.from.CNV.R $sample\_name $sample\_cnr, where

**Algorithm**



Genomic regions of different copy number state are commonly observed in tumor. Capture-based sequencing of such regions is subjected to technical and biological variation, and thus give rise to a noisy readout. We assume that the log2-depth-ratio between tumor and normal samples on a set of genomic regions at similar copy number state in tumor would follow Gaussian distribution. We then have:

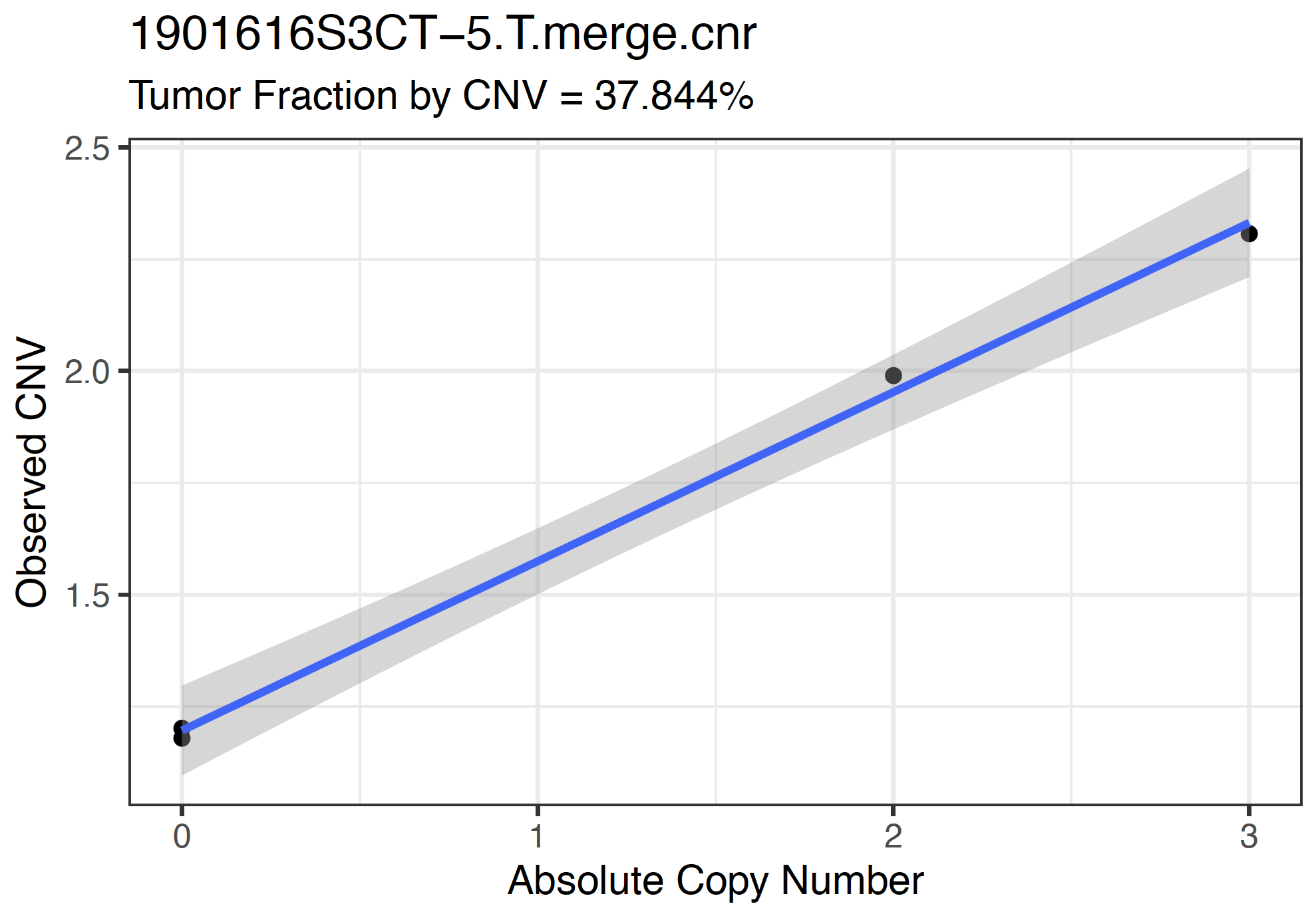


Where CN = the observed copy number (raw), tf = give tumor fraction of the sample, and x = the actual copy number state. Because copy number state of any given genomic region should be an integer, we have x as an integer. Noticing tf is constant across a number of paired CN:x combinations, we could infer tf from linear regression.

The script firstly takes in a cnr file which tells log2-depth-ratio of given tumor sample and the normal sample on a captured region. Then, model-based clustering based on parameterized finite Gaussian (Mclust) is applied to the log2-depth-ratios. This give rise to a number of mean-of-log2-depth-ratio (mu) for each cluster of genomic regions, which should be at similar copy number state. The script iterate through a series of combination of x in order to have a best linear fit.

**Outputs**

1. A scatter plot of observed (y axis) and inferred (x axis) copy number states, such as:



1. A json file output recording the linear regression output, in which:
2. coefficients = the linear regression coefficient, which could be inferred as tumor fraction.
3. r2 = the goodness-of-fit of linear regression
4. sample = the input sample name
5. cn\_stat\_mod = the inferred copy number state for each cluster. Sometimes there could be multiple clusters with similar inferred CN, which is possible.
6. k = tested knn-clustering number
7. Intercept = linear regression intercept
8. Intercept\_info = 1-intercept/2 . This should be the ‘tumor fraction at CN=0’ and it should be close to `coefficients`. If abs(Intercept\_info-coefficients) > 0.05 then it flags for unsatisfactory fit and the result could not be used. It might be necessary to re-run the script, since knn clustering is innately random. Usually abs(Intercept\_info-coefficients) <= 0.05 is considered a good fit.

|  |
| --- |
| [  {  "sample": "1901616S3CT-5.T.merge.cnr",  "coefficients": 0.3784,  "r2": 0.9965,  "cn\_stat\_mod": "0,0,2,3",  "k": 5,  "Intercept": 1.1962,  "Intercept\_info": 0.4019  }  ] |

1. A txt file output which is a tab-delimited version of the json file.

**Environment**

|  |
| --- |
| > sessionInfo()  R version 3.6.2 (2019-12-12)  Platform: x86\_64-pc-linux-gnu (64-bit)  Running under: CentOS Linux 7 (Core)  Matrix products: default  BLAS: /gpfs/bin/R-3.6.2/lib/libRblas.so  LAPACK: /gpfs/bin/R-3.6.2/lib/libRlapack.so  locale:  [1] LC\_CTYPE=en\_US.UTF-8 LC\_NUMERIC=C  [3] LC\_TIME=en\_US.UTF-8 LC\_COLLATE=en\_US.UTF-8  [5] LC\_MONETARY=en\_US.UTF-8 LC\_MESSAGES=en\_US.UTF-8  [7] LC\_PAPER=en\_US.UTF-8 LC\_NAME=C  [9] LC\_ADDRESS=C LC\_TELEPHONE=C  [11] LC\_MEASUREMENT=en\_US.UTF-8 LC\_IDENTIFICATION=C  attached base packages:  [1] parallel stats4 stats graphics grDevices utils datasets  [8] methods base  other attached packages:  [1] mclust\_5.4.6 mixtools\_1.2.0 cowplot\_1.1.0  [4] ggthemes\_4.2.0 RColorBrewer\_1.1-2 ggforce\_0.3.2  [7] bezier\_1.1.2 GenomicRanges\_1.38.0 GenomeInfoDb\_1.22.1  [10] IRanges\_2.20.2 S4Vectors\_0.24.4 BiocGenerics\_0.32.0  [13] readr\_1.4.0 tidyr\_1.1.2 reshape2\_1.4.4  [16] dplyr\_1.0.2 ggplot2\_3.3.5  loaded via a namespace (and not attached):  [1] Rcpp\_1.0.5 pillar\_1.4.6 compiler\_3.6.2  [4] plyr\_1.8.6 XVector\_0.26.0 bitops\_1.0-6  [7] tools\_3.6.2 zlibbioc\_1.32.0 lattice\_0.20-41  [10] lifecycle\_0.2.0 tibble\_3.0.4 gtable\_0.3.0  [13] pkgconfig\_2.0.3 rlang\_0.4.11 Matrix\_1.2-18  [16] GenomeInfoDbData\_1.2.2 withr\_2.3.0 stringr\_1.4.0  [19] generics\_0.0.2 vctrs\_0.3.4 hms\_0.5.3  [22] segmented\_1.2-0 grid\_3.6.2 tidyselect\_1.1.0  [25] glue\_1.4.2 R6\_2.4.1 survival\_3.2-7  [28] polyclip\_1.10-0 kernlab\_0.9-29 farver\_2.0.3  [31] tweenr\_1.0.1 purrr\_0.3.4 magrittr\_1.5  [34] splines\_3.6.2 MASS\_7.3-53 scales\_1.1.1  [37] ellipsis\_0.3.1 colorspace\_1.4-1 stringi\_1.5.3  [40] RCurl\_1.98-1.2 munsell\_0.5.0 crayon\_1.3.4 |