

Canopy vignette

Yuchao Jiang
yuchaoj@upenn.edu

September 2, 2016

This is a demo for using the **Canopy** package in R. **Canopy** is a statistical framework and computational procedure for identifying subpopulations within a tumor, determining the mutation profiles of each subpopulation, and inferring the tumor's phylogenetic history. The input to **Canopy** are variant allele frequencies of somatic single nucleotide alterations (SNAs) along with allele-specific coverage ratios between the tumor and matched normal sample for somatic copy number alterations (CNAs). These quantities can be directly taken from the output of existing software. **Canopy** provides a general mathematical framework for pooling data across samples and sites to infer the underlying parameters. For SNAs that fall within CNA regions, **Canopy** infers their temporal ordering and resolves their phase. When there are multiple evolutionary configurations consistent with the data, **Canopy** outputs all configurations along with their confidence.

Below is an example on reconstructing tumor phylogeny of a transplantable metastasis model system derived from a heterogeneous human breast cancer cell line MDA-MB-231. Cancer cells from the parental line MDA-MB-231 were engrafted into mouse hosts leading to organ-specific metastasis. Mixed cell populations (MCPs) were in vivo selected from either bone or lung metastasis and grew into phenotypically stable and metastatically competent cancer cell lines. The parental line as well as the MCP sublines were whole-exome sequenced with somatic SNAs and CNAs profiled. **Canopy** is used to infer metastatic phylogeny.

Canopy's webpage is [here](#). A **demo code** for **Canopy** can be found [here](#). Script for dataset from the MDA231 study is attached below with step-by-step decomposition and explanation. Online **Q&A forum** for **Canopy** is available [here](#). If you've any questions regarding the software, you can also email us at canopy_phylogeny@googlegroups.com.

1. Installation

R package **Canopy** is available from CRAN (<https://cran.r-project.org/web/packages/Canopy/>):

```
> install.packages('Canopy')
```

A devel version can be installed from GitHub (<https://github.com/yuchaojiang/Canopy/>):

```
> install.packages("devtools")
> library(devtools)
> install_github("yuchaojiang/Canopy/package")
```

2. Canopy workflow

2.1 CNA and SNA input

The input to **Canopy** are variant allele frequencies of somatic SNAs along with allele-specific coverage ratios between the tumor and matched normal sample for somatic CNAs. For SNAs, let the matrices R and X be, respectively, the

number of reads containing the mutant allele and the total number of reads for each locus across all samples. The ratio R/X is the proportion of reads supporting the mutant allele, known as the variant allele frequency. For CNAs, Canopy directly takes output from FALCON, FALCON-X, or other allele-specific copy number estimation methods. These outputs are in the form of estimated major and minor copy number ratios, respectively denoted by W^M and W^m , with their corresponding standard errors ϵ^M and ϵ^m . Matrix Y specifies whether SNAs are affected by CNAs; matrix C specifies whether CNA regions harbor specific CNAs (this input is only needed if overlapping CNA events are observed).

Below is demo data input from project MDA231 (first case study in our paper).

```

> library(Canopy)
> data("MDA231")
> projectname = MDA231$projectname ## name of project
> R = MDA231$R; R ## mutant allele read depth (for SNAs)

      MCP1833_bone MCP1834_lung MCP2287_bone MDA-MB-231_parental MCP3481_lung
BRAF      155          59          136          77          49
KRAS      44          21          54          19          17
ALPK2     37          17          28          10          7
RYR1      44          0           26          0           0

> X = MDA231$X; X ## total depth (for SNAs)

      MCP1833_bone MCP1834_lung MCP2287_bone MDA-MB-231_parental MCP3481_lung
BRAF      157          111          177          146          71
KRAS      44          30           64          42          27
ALPK2     63          17           65          24          7
RYR1     107          56          165          55          43

> WM = MDA231$WM; WM ## observed major copy number (for CNA regions)

      MCP1833_bone MCP1834_lung MCP2287_bone MDA-MB-231_parental MCP3481_lung
chr7      2.998          2.002          2.603          2.000          2.001
chr12     1.998          1.998          1.603          1.001          1.999
chr18     1.000          2.992          1.000          1.002          2.996
chr19     2.000          2.000          2.000          2.000          2.000

> Wm = MDA231$Wm; Wm ## observed minor copy number (for CNA regions)

      MCP1833_bone MCP1834_lung MCP2287_bone MDA-MB-231_parental MCP3481_lung
chr7      0.002          0.998          0.397          1.000          0.999
chr12     0.002          0.998          0.397          1.000          0.999
chr18     1.000          0.004          1.000          0.999          0.002
chr19     1.000          1.000          1.000          1.000          1.000

> epsilonM = MDA231$epsilonM ## standard deviation of WM, pre-fixed here
> epsilonMm = MDA231$epsilonMm ## standard deviation of Wm, pre-fixed here
> ## Matrix C specifies whether CNA regions harbor specific CNAs
> ## only needed if overlapping CNAs are observed, specifying which CNAs overlap
> C = MDA231$C; C

      chr7_1 chr7_2 chr12_1 chr12_2 chr18 chr19
chr7      1      1      0      0      0      0
chr12     0      0      1      1      0      0
chr18     0      0      0      0      1      0
chr19     0      0      0      0      0      1

```

```
> Y = MDA231$Y; Y ## whether SNAs are affected by CNAs
```

	non-cna_region	chr7	chr12	chr18	chr19
BRAF	0	1	0	0	0
KRAS	0	0	1	0	0
ALPK2	0	0	0	1	0
RYR1	0	0	0	0	1

2.2 Phylogenetic tree (unknown)

Each sampled tree is modeled as a list by `Canopy`. Below are the tree elements of the most likely tree from the project MDA231 (first case study in the paper). This most likely tree is obtained from the posterior distribution in the tree space from the MCMC sampling (detailed in section 2.3). How to visualize/plot the sampled trees is in section 2.6.

```
> data('MDA231_tree')
```

```
> MDA231_tree$Z # Z matrix specifies the position of the SNAs along the tree branch
```

	clone1	clone2	clone3	clone4
BRAF	0	1	1	1
KRAS	0	1	1	1
ALPK2	0	1	1	1
RYR1	0	0	1	0

```
> MDA231_tree$cna.copy # major and minor copy number (interger values) for each CNA
```

	chr7_LOH	chr7_dup	chr12_dup	chr12_LOH	chr18_LOH	chr19_dup
major_copy	3	2	2	2	3	2
minor_copy	0	1	1	0	0	1

```
> MDA231_tree$CM # Major copy per clone for each CNA
```

	clone1	clone2	clone3	clone4
chr7	1	2	3	2
chr12	1	1	2	2
chr18	1	1	1	3
chr19	1	2	2	2

```
> MDA231_tree$Cm # Minor copy per clone for each CNA
```

	clone1	clone2	clone3	clone4
chr7	1	1	0	1
chr12	1	1	0	1
chr18	1	1	1	0
chr19	1	1	1	1

```
> MDA231_tree$Q # whether an SNA precedes a CNA
```

	chr7_LOH	chr7_dup	chr12_dup	chr12_LOH	chr18_LOH	chr19_dup
BRAF	1	1	0	0	0	0
KRAS	0	0	1	1	0	0
ALPK2	0	0	0	0	1	0
RYR1	0	0	0	0	0	0

```
> MDA231_tree$H # if an SNA precedes a CNA, whether it resides in the major copy
```

	chr7_LOH	chr7_dup	chr12_dup	chr12_LOH	chr18_LOH	chr19_dup
BRAF	1	1	0	0	0	0
KRAS	0	0	1	1	0	0
ALPK2	0	0	0	0	1	0
RYR1	0	0	0	0	0	0

```
> MDA231_tree$P # clonal composition for each sample
```

	MCP1833_bone	MCP1834_lung	MCP2287_bone	MDA-MB-231_parental	MCP3481_lung
clone1	0.000	0.000	0.000	0.000	0.000
clone2	0.006	0.000	0.399	0.992	0.003
clone3	0.991	0.001	0.600	0.001	0.001
clone4	0.003	0.999	0.001	0.007	0.996

```
> MDA231_tree$VAF # VAF based on current tree structure
```

	MCP1833_bone	MCP1834_lung	MCP2287_bone	MDA-MB-231_parental	MCP3481_lung
BRAF	0.997	0.667	0.867	0.667	0.667
KRAS	0.996	0.667	0.800	0.502	0.667
ALPK2	0.502	1.000	0.501	0.505	0.999
RYR1	0.330	0.000	0.200	0.000	0.000

2.3 MCMC sampling

Canopy samples in subtree space with varying number of subclones (denoted as K) by a Markov chain Monte Carlo (MCMC) method. A plot of posterior likelihood (pdf format) will be generated for each subtree space and we recommend users to refer to the plot as a sanity check for sampling convergence and to choose the number of burn-ins and thinning accordingly. Note that this step can be time-consuming, especially with larger number of chains (`numchain` specifies the number of chains with random initiations, a larger value of which is in favor of not getting stuck in local optima) and longer chains (`simrun` specifies number of iterations per chain). MCMC sampling is the most computationally heavy step in Canopy. It is recommended that jobs are run in parallel on high-performance cluster.

```
> K = 3:6 # number of subclones
> numchain = 20 # number of chains with random initiations
> sampchain = canopy.sample(R = R, X = X, WM = WM, Wm = Wm, epsilonM = epsilonM,
+   epsilonNm = epsilonNm, C = C, Y = Y, K = K, numchain = numchain,
+   simrun = 50000, writeskip = 200, projectname = projectname,
+   cell.line = TRUE, plot.likelihood = TRUE)
> save.image(file = paste(projectname, '_postmcmc_image.rda', sep=''),
+   compress = 'xz')

> length(sampchain) ## number of subtree spaces (K=3:6)

[1] 4

> length(sampchain[[which(K==4)]] ) ## number of chains for subtree space with 4 subclones

[1] 20

> length(sampchain[[which(K==4)]][[1]]) ## number of posterior trees in each chain

[1] 250
```

2.4 BIC for model selection

Canopy uses BIC as a model selection criterion to determine the optimal number of subclones.

```
> burnin = 100
> thin = 10
> bic = canopy.BIC(sampchain = sampchain, projectname = projectname, K = K,
+                 numchain = numchain, burnin = burnin, thin = thin, pdf = FALSE)

k = 3 : mean likelihood -17243.62 .
k = 4 : mean likelihood -548.2178 .
k = 5 : mean likelihood -597.4001 .
k = 6 : mean likelihood -586.8208 .

> optK = K[which.max(bic)]
```

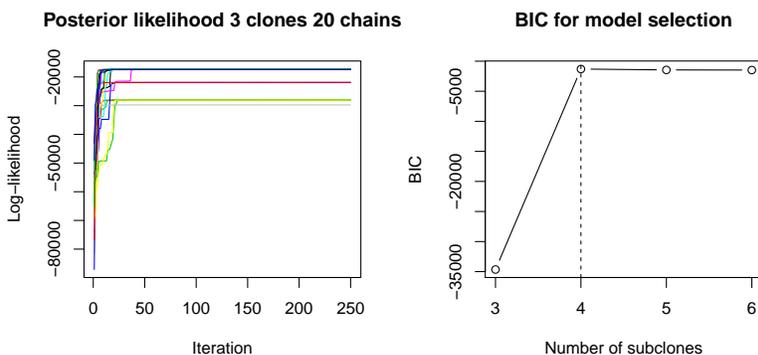


Figure 1: Posterior likelihood of MCMC (chains are colored differently) and BIC as a model selection method.

2.5 Posterior evaluation of sampled trees

Canopy then runs a posterior evaluation of all sampled trees by MCMC. If modes of posterior probabilities (second column of `config.summary`) aren't obvious, check if the algorithm has converged (and run sampling longer if not).

```
> post = canopy.post(sampchain = sampchain, projectname = projectname, K = K,
+                  numchain = numchain, burnin = burnin, thin = thin, optK = optK,
+                  C = C, post.config.cutoff = 0.05)
> samptreethin = post[[1]] # list of all post-burnin and thinning trees
> samptreethin.lik = post[[2]] # likelihoods of trees in samptree
> config = post[[3]] # configuration for each posterior tree
> config.summary = post[[4]] # configuration summary
> print(config.summary)
```

	Configuration	Post_prob	Mean_post_lik
[1,]	1	0.547	-556.90
[2,]	2	0.080	-557.00
[3,]	3	0.093	-557.19
[4,]	4	0.133	-556.51
[5,]	5	0.147	-555.87

```

> # first column: tree configuration
> # second column: posterior configuration probability in the entire tree space
> # third column: posterior configuration likelihood in the subtree space

```

2.6 Tree output and plotting

One can then use Canopy to output and plot the most likely tree (i.e., tree with the highest posterior likelihood). Mutations, clonal frequencies, and tree topology, etc., of the tree are obtained from the posterior distributions of subtree space with trees having the same configuration. In our MDA231 example, the most likely tree is the tree having configuration 3.

```

> config.i = config.summary[which.max(config.summary[,3]),1]
> cat('Configuration', config.i, 'has the highest posterior likelihood!\n')

```

Configuration 5 has the highest posterior likelihood!

```

> # plot the most likely tree in the posterior tree space
> output.tree = canopy.output(post, config.i, C)
> canopy.plottree(output.tree)
> # plot the tree with configuration 1 in the posterior tree space
> output.tree = canopy.output(post, 1, C)
> canopy.plottree(output.tree, pdf=TRUE, pdf.name =
+ paste(projectname, '_first_config.pdf', sep=' '))

```

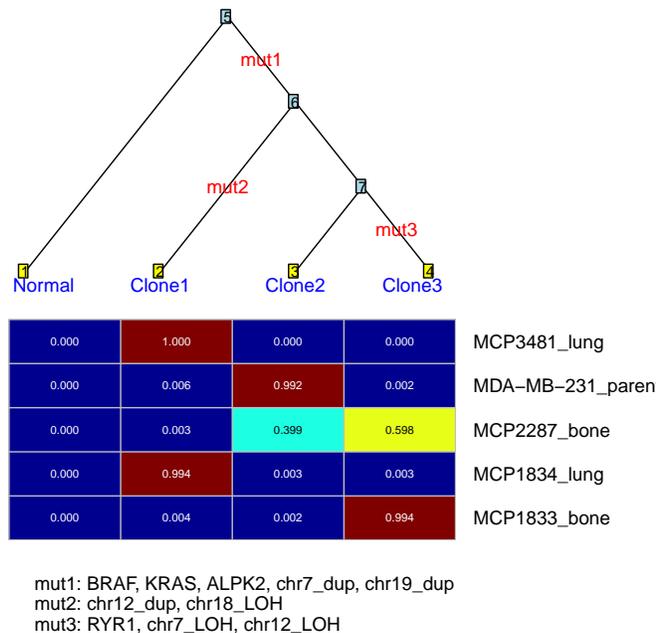


Figure 2: Most likely tree by Canopy for project MDA231.

3. Try it yourself

Now try Canopy yourself using the simulated toy dataset below! Note that no overlapping CNAs are used as input and thus matrix C doesn't need to be specified.

```
> library(Canopy)
> data(toy)
> projectname = 'toy'
> R = toy$R; X = toy$X; WM = toy$WM; Wm = toy$Wm
> epsilonM = toy$epsilonM; epsilonm = toy$epsilonm; Y = toy$Y
> K = 3:6; numchain = 10
> sampchain = canopy.sample(R = R, X = X, WM = WM, Wm = Wm, epsilonM = epsilonM,
+ epsilonm = epsilonm, C = NULL, Y = Y, K = K,
+ numchain = numchain, simrun = 50000, writeskip = 200,
+ projectname = projectname, cell.line = FALSE,
+ plot.likelihood = TRUE)
```

The most likely tree is shown below. There should be only one tree configuration from the posterior tree space. The code for this toy dataset analysis can be found [here](#).

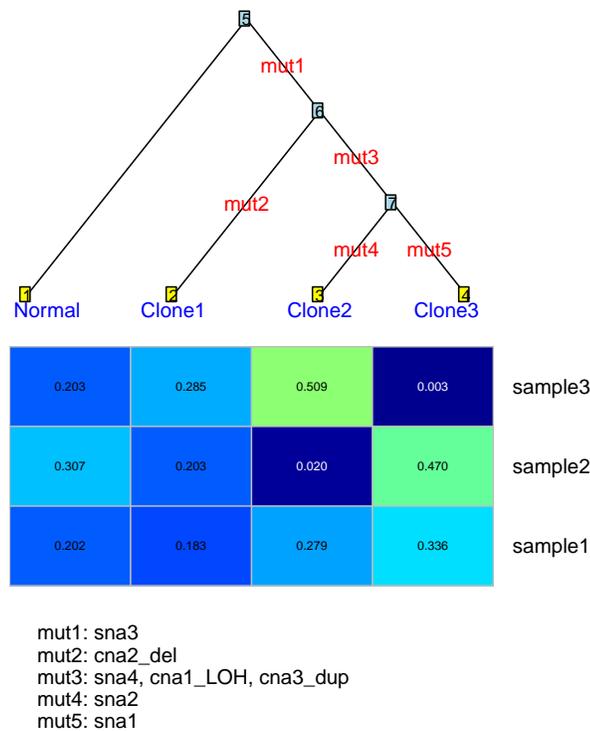


Figure 3: Most likely tree by Canopy for simulated toy dataset.

The second toy example has a different tree topology. Feel free to try Canopy on this dataset too! There should be also just one tree configuration as is shown below from the posterior tree space. The code for this toy dataset analysis can be found [here](#).

```
> library(Canopy)
> data(toy2)
> projectname = 'toy2'
> R = toy2$R; X = toy2$X; WM = toy2$WM; Wm = toy2$Wm
> epsilonM = toy2$epsilonM; epsilonNm = toy2$epsilonNm; Y = toy2$Y
> true.tree = toy2$true.tree
> K = 3:6; numchain = 10
> sampchain = canopy.sample(R = R, X = X, WM = WM, Wm = Wm, epsilonM = epsilonM,
+ epsilonNm = epsilonNm, C = NULL, Y = Y, K = K,
+ numchain = numchain, simrun = 50000, writeskip = 200,
+ projectname = projectname, cell.line = FALSE,
+ plot.likelihood = TRUE)
```

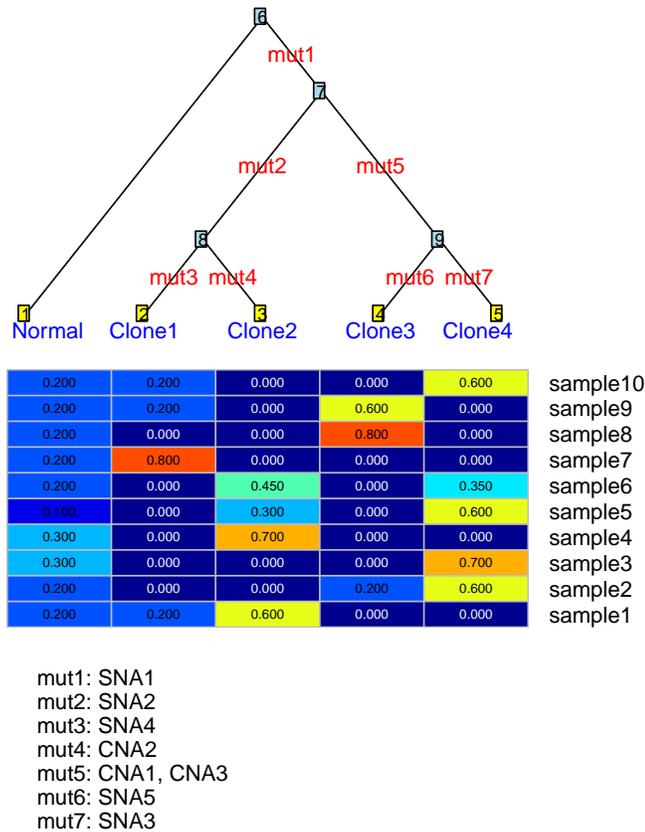


Figure 4: Most likely tree by Canopy for simulated toy dataset 2.

4. Citation

Assessing intra-tumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing, Yuchao Jiang, Yu Qiu, Andy J Minn, Nancy R zhang, Proceedings of the National Academy of Sciences, 2016. (html, pdf)

5. Session information:

Output of sessionInfo on the system on which this document was compiled:

- R version 3.2.3 (2015-12-10), x86_64-apple-darwin13.4.0
- Locale: C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, utils
- Other packages: Canopy 1.1.1, ape 3.5, fields 8.4-1, maps 3.1.1, spam 1.3-0
- Loaded via a namespace (and not attached): lattice 0.20-33, nlme 3.1-128, tools 3.2.3