Canopy vignette

Yuchao Jiang
yuchaoj@wharton.upenn.edu

October 29, 2015

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 R script 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 and/or yuchaoj@wharton.upenn.edu.

1. Installation

R package Canopy is currently under review at Bioconductor. Meanwhile, you can download the source Canopy_0.99.0.tar.gz and install following the instructions below.

```
> install.packages('ape')
> install.packages('fields')
> install.packages('Canopy_0.99.0.tar.gz', repos = NULL, type="source")
```

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 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 data input from project MDA231.

```
> 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$epsilonM ## standard deviation of WM, pre-fixed here
> epsilonm = MDA231$epsilonm ## standard deviation of Wm, pre-fixed here
> ## Matrix C specifices whether CNA regions harbor specific CNAs
> ## only needed if overlapping CNAs are observed
> 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 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). Jobs can be run in parallel with different K's 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,
+             epsilonm = epsilonm, 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
```

2.3 BIC for model selection

`Canopy` uses BIC as a model selection criterion to determine to 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 -17243.58 .
k = 4 : mean likelihood -548.1847 .
k = 5 : mean likelihood -576.1433 .
k = 6 : mean likelihood -586.7338 .

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

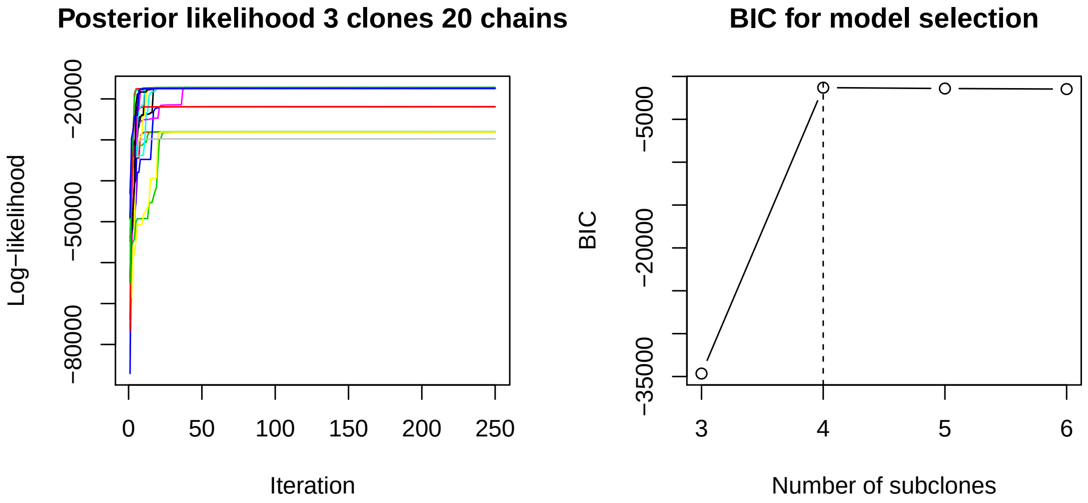


Figure 1: Posterior likelihood of MCMC (chains are colored differently) and BIC as a model selection method.

2.4 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     0.429       -548.64
[2,]             2     0.429       -548.49
[3,]             3     0.143       -548.09

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

2.5 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 3 has the highest posterior likelihood!
```

```
> output.tree = canopy.output(post, config.i, C)
> canopy.plottree(output.tree, pdf = FALSE)
```

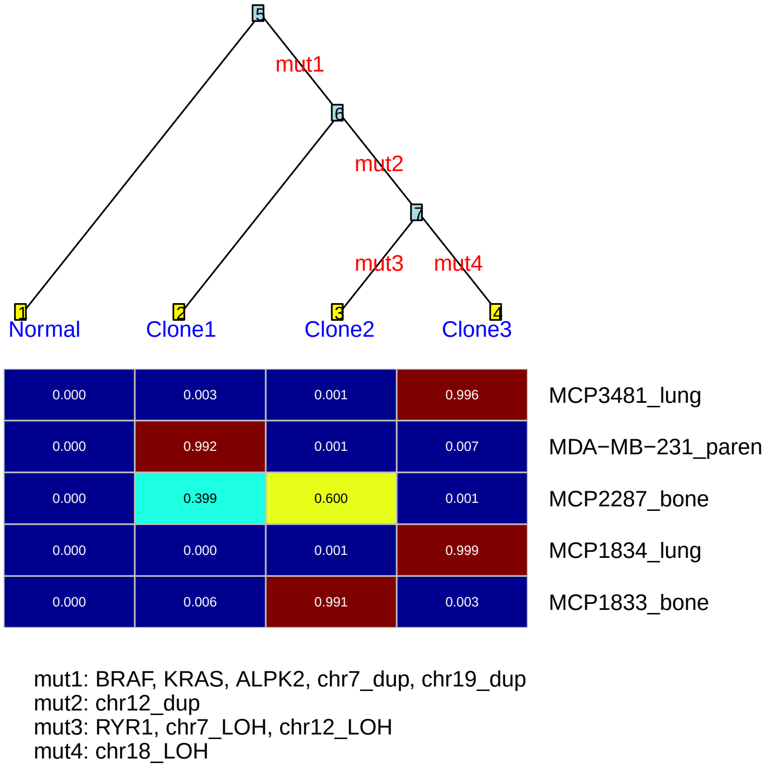

5
mut1
6
mut2
7
mut3
mut4
1
2
3
4
Normal
Clone1
Clone2
Clone3
0.000 0.003 0.001 0.996 MCP3481_lung
0.000 0.992 0.001 0.007 MDA−MB−231_paren
0.000 0.399 0.600 0.001 MCP2287_bone
0.000 0.000 0.001 0.999 MCP1834_lung
0.000 0.006 0.991 0.003 MCP1833_bone
mut1: BRAF, KRAS, ALPK2, chr7_dup, chr19_dup
mut2: chr12_dup
mut3: RYR1, chr7_LOH, chr12_LOH
mut4: chr18_LOH

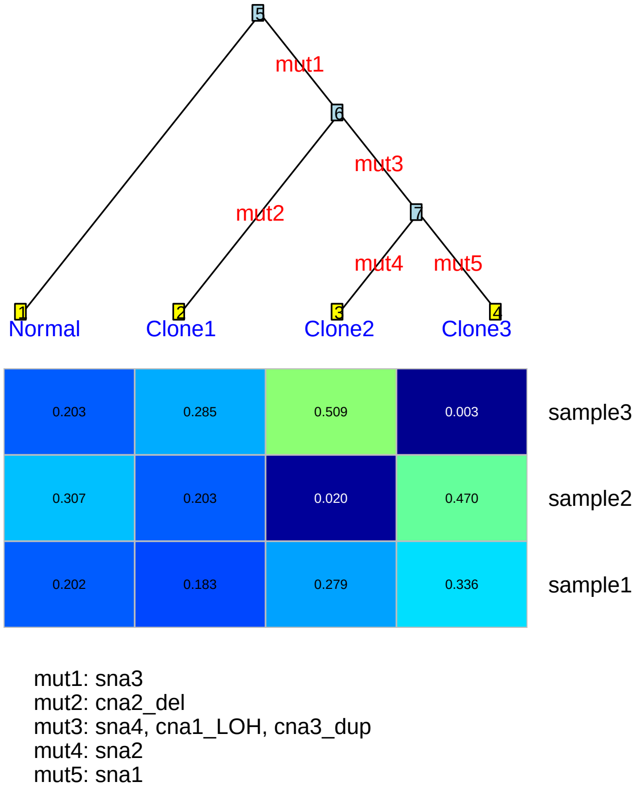
Figure 2: Most likely tree by Canopy for project MDA231.

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.

```
> 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)
```

There should be only one mode in the posterior tree space, with the most likely tree shown below.



Normal	Clone1	Clone2	Clone3	
0.203	0.285	0.509	0.003	sample3
0.307	0.203	0.020	0.470	sample2
0.202	0.183	0.279	0.336	sample1

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

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, submitted to Genome Biology, 2015.

5. Session information:

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

- R version 3.2.0 (2015-04-16), `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 1.0.0, ape 3.3, fields 8.3-5, maps 3.0.0-2, spam 1.2-1
- Loaded via a namespace (and not attached): lattice 0.20-33, nlme 3.1-122, tools 3.2.0