

9.1) Unconstrained Bayesian topology inference (RevBayes).

```
#####
# Reading in the Data #
#####

filenames <- v("18sublist.aln.fasta", "16sublist.aln.fasta")

n_data_subsets <- filenames.size()
for (i in 1:n_data_subsets) {
  data[i] = readDiscreteCharacterData(filenames[i])
}

taxa <- data[1].taxa()
n_species <- data[1].ntaxa()
n_branches <- 2 * n_species - 3

mvi = 0
mni = 0

#####
# Substitution Model #
# Loop over each data subset #
#####

for (i in 1:n_data_subsets) {
  er_prior[i] <- v(1,1,1,1,1,1)
  er[i] ~ dnDirichlet(er_prior[i])
  moves[++mvi] = mvSimplexElementScale(er[i], alpha=10, tune=true, weight=3)

  pi_prior[i] <- v(1,1,1,1)
  pi[i] ~ dnDirichlet(pi_prior[i])
  moves[++mvi] = mvSimplexElementScale(pi[i], alpha=10, tune=true, weight=2)

  Q[i] := fnGTR(er[i],pi[i])

  alpha_prior_mean[i] <- 5.0
  alpha_prior_sd[i] <- 0.587405
  alpha[i] ~ dnLognormal( alpha_prior_mean[i], alpha_prior_sd[i] )
  gamma_rates[i] := fnDiscretizeGamma( alpha[i], alpha[i], 4, false )

  moves[++mvi] = mvScale(alpha[i],weight=2)

  pinvar[i] ~ dnBeta(1,1)
  moves[++mvi] = mvScale(pinvar[i], lambda=0.1, tune=true, weight=2.0)
  moves[++mvi] = mvSlide(pinvar[i], delta=0.1, tune=true, weight=2.0)
}

#####
# Tree model #
#####
```

```

topology ~ dnUniformTopology(taxa=taxa)

moves[++mvi] = mvNNI(topology, weight=1.0)
moves[++mvi] = mvSPR(topology, weight=1.0)

for (i in 1:n_branches) {
  br_lens[i] ~ dnExponential(10.0)
  moves[++mvi] = mvScale(br_lens[i])
}

TL := sum(br_lens)

# unite topology and branch length vector into phylogeny object
phylogeny := treeAssembly(topology, br_lens)

#####
# PhyloCTMC Model #
#####

part_rate_mult ~ dnDirichlet( rep(1.0, n_data_subsets) )
moves[++mvi] = mvSimplexElementScale(part_rate_mult, alpha=10, tune=true, weight=2)

for (i in 1:n_data_subsets) {
  part_rate[i] := part_rate_mult[i] * n_data_subsets
}

for (i in 1:n_data_subsets) {
  phyloSeq[i] ~ dnPhyloCTMC(tree=phylogeny, Q=Q[i], branchRates=part_rate[i],
siteRates=gamma_rates[i], plnv=pinvar[i], type="DNA")
  phyloSeq[i].clamp(data[i])
}

#####
# Analysis #
#####

mymodel = model(phylogeny)

# add monitors
monitors[++mni] = mnModel(filename="output/gene.log",printgen=10)
monitors[++mni] = mnFile(phylogeny, filename="output/gene.trees", printgen=100)
monitors[++mni] = mnScreen(alpha,printgen=100)

# run the analysis
mymcmc = mcmc(mymodel, moves, monitors)
mymcmc.burnin(10000,1000)
mymcmc.run(300000)

# summarize output
treetrace = readTreeTrace("output/gene.trees")
treetrace.summarize()

map_tree = mapTree(treetrace,"output/gene_map.tre")
q()

```

9.2) Constrained Bayesian topology inference (RevBayes).

```
#####
# Reading in the Data #
#####

filenames <- v("18sublist.aln.fasta", "16sublist.aln.fasta")

n_data_subsets <- filenames.size()
for (i in 1:n_data_subsets) {
  data[i] = readDiscreteCharacterData(filenames[i])
}

# Get some useful variables from the data. We need these later on.
taxa <- data[1].taxa()
n_species <- data[1].ntaxa()
n_branches <- 2 * n_species - 3

mvi = 0
mni = 0

#####
# Substitution Model #
# Loop over each data subset #
#####

for (i in 1:n_data_subsets) {

  # exchangeability rates for partition i
  er_prior[i] <- v(1,1,1,1,1,1)
  er[i] ~ dnDirichlet(er_prior[i])
  moves[++mvi] = mvSimplexElementScale(er[i], alpha=10, tune=true, weight=3)

  # stationary frequencies for partition i
  pi_prior[i] <- v(1,1,1,1)
  pi[i] ~ dnDirichlet(pi_prior[i])
  moves[++mvi] = mvSimplexElementScale(pi[i], alpha=10, tune=true, weight=2)

  # rate matrix for partition i
  Q[i] := fnGTR(er[i],pi[i])

  # +Gamma for partition i
  alpha_prior_mean[i] <- 5.0
  alpha_prior_sd[i] <- 0.587405
  alpha[i] ~ dnLognormal( alpha_prior_mean[i], alpha_prior_sd[i] )
  gamma_rates[i] := fnDiscretizeGamma( alpha[i], alpha[i], 4, false )

  # add moves for the alpha parameter
  moves[++mvi] = mvScale(alpha[i],weight=2)

  # the probability of a site being invariable
  pinvar[i] ~ dnBeta(1,1)
  moves[++mvi] = mvScale(pinvar[i], lambda=0.1, tune=true, weight=2.0)
```

```

moves[++mvi] = mvSlide(pinvar[i], delta=0.1, tune=true, weight=2.0)
}

#####
# Tree model #
#####

#Constraints on clades
clade_A = clade("Resomia_ornicephala", "Frillagalma_vityazi")
clade_B = clade("Cordagalma_ordinatum", "Forskalia_formosa", "Forskalia_asymmetrica",
"Forskalia_edwardsii", "Forskalia_tholoides")
clade_C = clade("Athorybia_rosacea", "Agalma_clausi", "Agalma_elegans", "Agalma_okenii",
"Nanomia_bijuga", "Halistemma_rubrum")
clade_D = clade("Lychnagalma_utricularia", "Physophora_gilmeri", "Physophora_hydrostatica",
clade_C)
clade_BD = clade( clade_B, clade_D )
clade_ABC = clade( clade_A, clade_B, clade_D )
clade_E = clade( clade_ABC, "Erenna_richardi", "Erenna_sirena", "Erenna_laciniata",
"Stephalia_dilata", "Thermopalia_sp", "Marrus_claudanielis", "Physonect_sp",
"Stephanomia_amphytridis" )
clade_constraints = [ clade_A, clade_B, clade_C, clade_D, clade_BD, clade_ABC, clade_E ]

topology ~ dnUniformTopology( taxa=taxa, constraints=clade_constraints )

moves[++mvi] = mvNNI(topology, weight=1.0)
moves[++mvi] = mvSPR(topology, weight=1.0)

for (i in 1:n_branches) {
  br_lens[i] ~ dnExponential(10.0)
  moves[++mvi] = mvScale(br_lens[i])
}

TL := sum(br_lens)

phylogeny := treeAssembly(topology, br_lens)

#####
# PhyloCTMC Model #
#####

part_rate_mult ~ dnDirichlet( rep(1.0, n_data_subsets) )
moves[++mvi] = mvSimplexElementScale(part_rate_mult, alpha=10, tune=true, weight=2)

for (i in 1:n_data_subsets) {
  part_rate[i] := part_rate_mult[i] * n_data_subsets
}

for (i in 1:n_data_subsets) {
  phyloSeq[i] ~ dnPhyloCTMC(tree=phylogeny, Q=Q[i], branchRates=part_rate[i],
siteRates=gamma_rates[i], plnv=pinvar[i], type="DNA")
  phyloSeq[i].clamp(data[i])
}
#####

```

```
# Analysis #
#####

mymodel = model(phylogeny)

# add monitors
monitors[++mni] = mnModel(filename="output/gene_ct.log",printgen=10)
monitors[++mni] = mnFile(phylogeny, filename="output/gene_ct.trees", printgen=100)
monitors[++mni] = mnScreen(alpha,printgen=100)

# run the analysis
mymcmc = mcmc(mymodel, moves, monitors)
mymcmc.burnin(10000,1000)
mymcmc.run(300000)

# summarize output
treetrace = readTreeTrace("output/gene_ct.trees")
treetrace.summarize()

map_tree = mapTree(treetrace,"output/gene_map_ct.tre")

q()
```

9.3) Bayesian timetree with fixed topology, with relaxed molecular clock.

```
#####
#   Timetree full model specification and MCMC set up for estimating
#   time-calibrated phylogenies under a strict clock model
#####

setOption("useScaling", "true")

### Load the sequence alignment
D <- readDiscreteCharacterData(file="concatenate.fasta")

### initialize an iterator for the moves vector
mi = 1

n_sites <- D.nchar()

#####
#   Birth-death, rho=55/186, fixed topology
#####

## read in the tree
backbone <- readTrees("timetree_rev_constrained/backbone_rooted_ultram.tre")[1]

n_taxa <- backbone.ntips()
taxa <- backbone.taxa()

#####
##### Birth-death process model #####
#####

diversification ~ dnExponential(10.0)
turnover ~ dnBeta(2.0, 2.0)

denom := abs(1.0 - turnover)
birth_rate := diversification / denom
death_rate := (turnover * diversification) / denom

### fix rho to 0.2957, since there are ~186 described species and we have sampled 55
rho <- 0.2957

### the root age is an independent stochastic node with a lognormal prior
#arbitrary numbers since we don't have time calibrations from fossils
mean_ra <- 0.5
stdv_ra <- 0.001
mu_ra <- ln(mean_ra) - ((stdv_ra*stdv_ra) * 0.5)
root_time ~ dnLnorm(mu_ra, stdv_ra, offset=1)

#####
#Constraints on clades#
#####
```

```

### the time tree is a stochastic node modeled by the constant rate birth-death process
(dnBDP)
tree_prior = dnBDP(lambda=birth_rate, mu=death_rate, rho=rho, rootAge=root_time,
samplingStrategy="uniform", condition="nTaxa", taxa=taxa)

timetree ~ dnConstrainedTopology(tree_prior, backbone = backbone)
timetree.setValue(backbone)

moves[mi++] = mvScale(diversification,lambda=1.0,tune=true,weight=3.0)
moves[mi++] = mvSlide(turnover,delta=1.0,tune=true,weight=3.0)

##### Tree Moves #####

### add moves on the tree node times, including the root time, which is outside of the timetree
moves[mi++] = mvNodeTimeSlideUniform(timetree, weight=30.0)

### and moves for the tree topology
moves[mi++] = mvNNI(timetree, weight=8.0)
moves[mi++] = mvNarrow(timetree, weight=8.0)
moves[mi++] = mvFNPR(timetree, weight=8.0)

#####
#       Timetree UCLN relaxed-clock model specification
#####

n_branches <- 2 * n_taxa - 3
ucln_mean ~ dnExponential(2.0)
ucln_sigma ~ dnExponential(3.0)
ucln_var := ucln_sigma * ucln_sigma
ucln_mu := ln(ucln_mean) - (ucln_var * 0.5)

moves[mi++] = mvScale(ucln_mean, lambda=1.0, tune=true, weight=4.0)
moves[mi++] = mvScale(ucln_sigma, lambda=0.5, tune=true, weight=4.0)

for(i in 1:n_branches){
  branch_rates[i] ~ dnLnorm(ucln_mu, ucln_sigma)
  moves[mi++] = mvScale(branch_rates[i],lambda=1.0,tune=true,weight=2.0)
}
moves[mi++] = mvVectorScale(branch_rates,lambda=1.0,tune=true,weight=2.0)
moves[mi++] = mvVectorSingleElementScale(branch_rates,lambda=30.0,tune=true,weight=1.0)

mean_rt := mean(branch_rates)
n_species <- D.ntaxa()
er_prior <- v(1,1,1,1,1,1)
er ~ dnDirichlet(er_prior)
moves[mi++] = mvSimplexElementScale(er, alpha=10, tune=true, weight=3)

pi_prior <- v(1,1,1,1)
pi ~ dnDirichlet(pi_prior)
moves[mi++] = mvSimplexElementScale(pi, alpha=10, tune=true, weight=2)

Q := fnGTR(er,pi)

```

```

phySeq ~ dnPhyloCTMC(tree=timetree, Q=Q, branchRates=mean_rt, nSites=n_sites,
type="DNA")

phySeq.clamp(D)

#####
##### MCMC #####
#####

mymodel = model(er)

### set up the monitors that will output parameter values to file and screen
monitors[1] = mnModel(filename="output/TimeTree_siphs_mcmc.log", printgen=10)
monitors[2] = mnFile(filename="output/TimeTree_siphs_mcmc.trees", printgen=10, timetree)
#monitors[3] = mnScreen(printgen=100, root_time)

### workspace mcmc ###
mymcmc = mcmc(mymodel, monitors, moves)

### pre-burnin to tune the proposals ###
mymcmc.burnin(generations=5000,tuningInterval=100)

### run the MCMC ###
mymcmc.run(generations=10000)

### display proposal acceptance rates and tuning ###
mymcmc.operatorSummary()

### summarize the trees ###
tt = readTreeTrace("output/TimeTree_siphs_mcmc.trees", "clock")
tt.summarize()

### write MAP tree to file
mapTree(tt, "output/TimeTree_siphs_mcmc_MAP.tre")

q()

```


9.4) Backbone tree topology used in 9.3.

((((((((((Abylopsis_tetragona:0.1920090572,Diphyes_dispar:0.1920090572):0.07038825201,
(((Lensia_conoidea:0.12776145,Sulculeolaria_quadriavlis:0.12776145):
0.0724035863,Muggiaea_atlantica:0.2001650363):0.01772580604,Chelophyes_appendiculata:
0.2178908423):0.0445064669):0.1207186703,(Sphaeronectes_christiansonae:
0.02137209896,Sphaeronectes_koellikeri:0.02137209896):0.3617438805):0.06838539141,
((Chuniphyes_moserae:0.1070447673,Chuniphyes_multidentata:0.1070447673):
0.1400302115,Kephyes_ovata:0.2470749788):0.1096919361,Crystallophyes_amygdalina:
0.3567669149):0.09473445601):0.05336762283,Tottonophyes_enigmatica:0.5048689937):
0.03684738417,((((Hippopodius_hippopus:0.1629358231,Vogtia_pentacantha:0.1629358231):
0.07817927636,Vogtia_glabra:0.2411150995):0.2274944812,Gymnopraia_lapislazula:
0.4686095807):0.01821777298,Rosacea_flaccida:0.4868273537):0.02139388915,
(Desmophyes_haematogaster:0.19046821,Lilyopsis_fluoracantha:0.19046821):0.3177530328):
0.03349513508):0.0250076704,Craseoa_lathetica:0.5667240483):0.02922570111,
((Nectadamas_diomedeeae:0.2000425839,Nectopyramis_natans:0.2000425839):
0.1163169974,Praya_dubia:0.3163595813):0.2795901681):0.0235863006,((((((((Agalma_okenii:
0.03947243462,Athorybia_rosacea:0.03947243462):0.01150594871,Agalma_elegans:
0.05097838333):0.00782366888,Agalma_clausi:0.05880205221):
0.03987968999,Halistemma_rubrum:0.0986817422):0.1363393864,Nanomia_bijuga:
0.2350211286):0.1574236295,((Physophora_gilmeri:0.1996894035,Physophora_hydrostatica:
0.1996894035):0.04503863,Lychnagalma_utricularia:0.2447280335):0.1477167245):
0.01217998655,((((Forskalia_edwardsii:0.06686802613,Forskalia_tholoides:0.06686802613):
0.05276556323,Forskalia_formosa:0.1196335894):0.04922444644,Forskalia_asymmetrica:
0.1688580358):0.2108026519,Cordagalma_ordinatum:0.3796606877):0.02496405696):
0.002302330604,(Frillagalma_vityazi:0.3758143426,Resomia_ornicephala:0.3758143426):
0.03111273266):0.1491307595,((((Erenna_richardi:0.02029803167,Erenna_sirena:
0.02029803167):0.009412289298,Erenna_laciniata:0.02971032097):
0.05264891477,Stephanomia_amphytridis:0.08235923574):0.09325613073,
((Marrus_claudanielis:0.08423382499,Physonect_sp:0.08423382499):0.05121942472,
(Stephalia_dilata:0.03092743461,Thermopalia_sp:0.03092743461):0.1045258151):
0.04016211677):0.3804424682):0.06347821529):0.01324242897,(Bargmannia_amoena:
0.09229156571,Bargmannia_elongata:0.09229156571):0.5404869133):0.0277360691,
(Apolemia_lanosa:0.04834004414,Apolemia_rubriversa:0.04834004414):0.6121745039):
0.1083139728,((Rhizophysa_eysenhardtii:0.1600232864,Rhizophysa_filiformis:0.1600232864):
0.2114603422,Physalia_physalis:0.3714836287):0.3973448922):0.2311714792,
(((Clytia_hemisphaerica:0.7494295556,Hydra_circumcincta:0.7494295556):
0.1469102,Ectopleura_dumortieri:0.8963397556):0.1036602444,((Porpita_porpita:
0.1556570118,Velella_velella:0.1556570118):0.4634878517,Staurocladia_wellingtoni:
0.6191448635):0.3808551365):-4.884981308e-15);