

GeneratedFigures

Venelin Mitov

6 June 2018

Collect inferred models

Here, we extract the inferred model and regimes and add them as columns to the testData_t2 data.table.
Manual execution.

Evaluate the 7 binary criterions on the inferred models

```
if(file.exists("../data-raw/ResultsTestData_t2/TestResultData_t2.RData")) {
  load("../data-raw/ResultsTestData_t2/TestResultData_t2.RData")
} else {
  testResultData <- NULL
}

testResultDataBhatt <- rbindlist(
  lapply(1:nrow(testData_t2), function(i) {

    resultFile <- paste0("../data-raw/ResultsTestData_t2/FinalResult_MixedGaussian_testData_t2_id_",
                          i)

    if(file.exists(resultFile)) {
      cat("Loading ", resultFile, "\n")
      load(resultFile)
      bestFitAIC <- RetrieveBestFitAIC(fitMappings)
      trueFromTestData <- RetrieveTrueFromTestData(testData_t2, i)

      bhatt <- bhattacharyya(trueFromTestData$tree, trueFromTestData$trueModel,
                               bestFitAIC$tree, bestFitAIC$inferredModel)

      cat("Bhatt: ", bhatt, "\n")
      data.table(i = i,
                 treeType=testData_t2[i, treeType],
                 treeSize=testData_t2[i, treeSize],
                 numClusters = testData_t2[i, numClusters],
                 crit = factor(c("Bhattacharyya dist.")),
                 tpr = NA,
                 fpr = NA,
                 AIC_Final = AIC(bestFitAIC$inferredModel),
                 AIC_True = AIC(trueFromTestData$trueModel),
                 inferred.x0 = bestFitAIC$inferredModel$X0[[1]],
                 inferred.y0 = bestFitAIC$inferredModel$X0[[2]],
                 true.x0 = trueFromTestData$trueModel$X0[[1]],
                 true.y0 = trueFromTestData$trueModel$X0[[2]],
                 value = bhatt
    }
  })
}
```

```

        )
    } else {
        NULL
    }

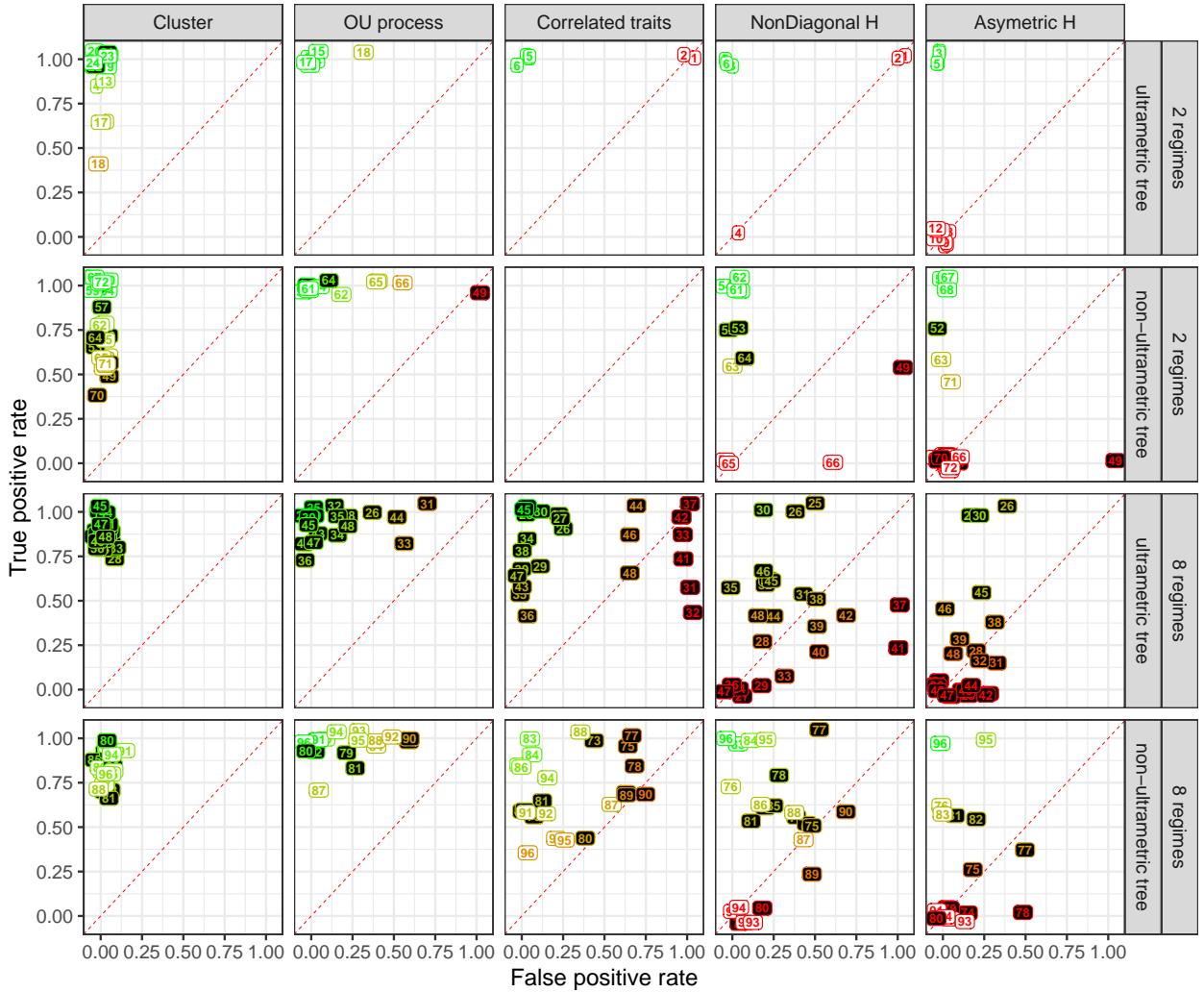
})
)

if(!is.null(testResultData)) {
    testResultData <- rbindlist(list(testResultData, testResultDataNew), fill = TRUE)
} else {
    testResultData <- testResultDataNew
}

testResultData[, crit2:=factor(crit, levels = c("Cluster",
                                                 "BM process",
                                                 "OU process",
                                                 "Uncorrelated traits",
                                                 "Correlated traits",
                                                 "NonDiagonal H",
                                                 "Asymmetric H",
                                                 "Bhattacharyya dist."))]

load("../data-raw/ResultsTestData_t2/TestResultData_t2.RData")
testResultData[, numClusters2:=paste0(numClusters, " regimes")]
testResultData[, treeType2:=factor(paste0(treeType, " tree"), levels=c("ultrametric tree", "non-ultrametric tree"))]
ggplot(testResultData[testSize == "small" & !(crit2 %in% c("BM process", "Uncorrelated traits"))]) +
    geom_abline(slope = 1, intercept = 0, linetype = 2, size = 0.2, color="red") +
    geom_label(aes(label = i, x = fpr, y = tpr,
                   #size = 0.5*apply(cbind((0.2+1-tpr), (0.2+fpr)), 1, max),
                   color = 0.5*apply(cbind((0.2+1-tpr), (0.2+fpr)), 1, max),
                   fill = AIC_Final<AIC_True ),
                   size = 2,
                   position = position_jitter(width=0.05, height = 0.05),
                   label.padding = unit(0.1, "lines"), fontface = "bold") +
    xlab("False positive rate") + ylab("True positive rate") +
    scale_color_continuous(low="green", high="red") +
    scale_fill_manual(values = c("TRUE"="white", "FALSE"="black")) +
    #scale_size_continuous(range = c(1.5, 3)) +
    scale_x_continuous(breaks=c(0, 0.25, 0.5, 0.75, 1)) +
    scale_y_continuous(breaks=c(0, 0.25, 0.5, 0.75, 1)) +
    facet_grid(numClusters2*treeType2~crit2) +
    theme_bw() +
    theme(legend.position = "none")

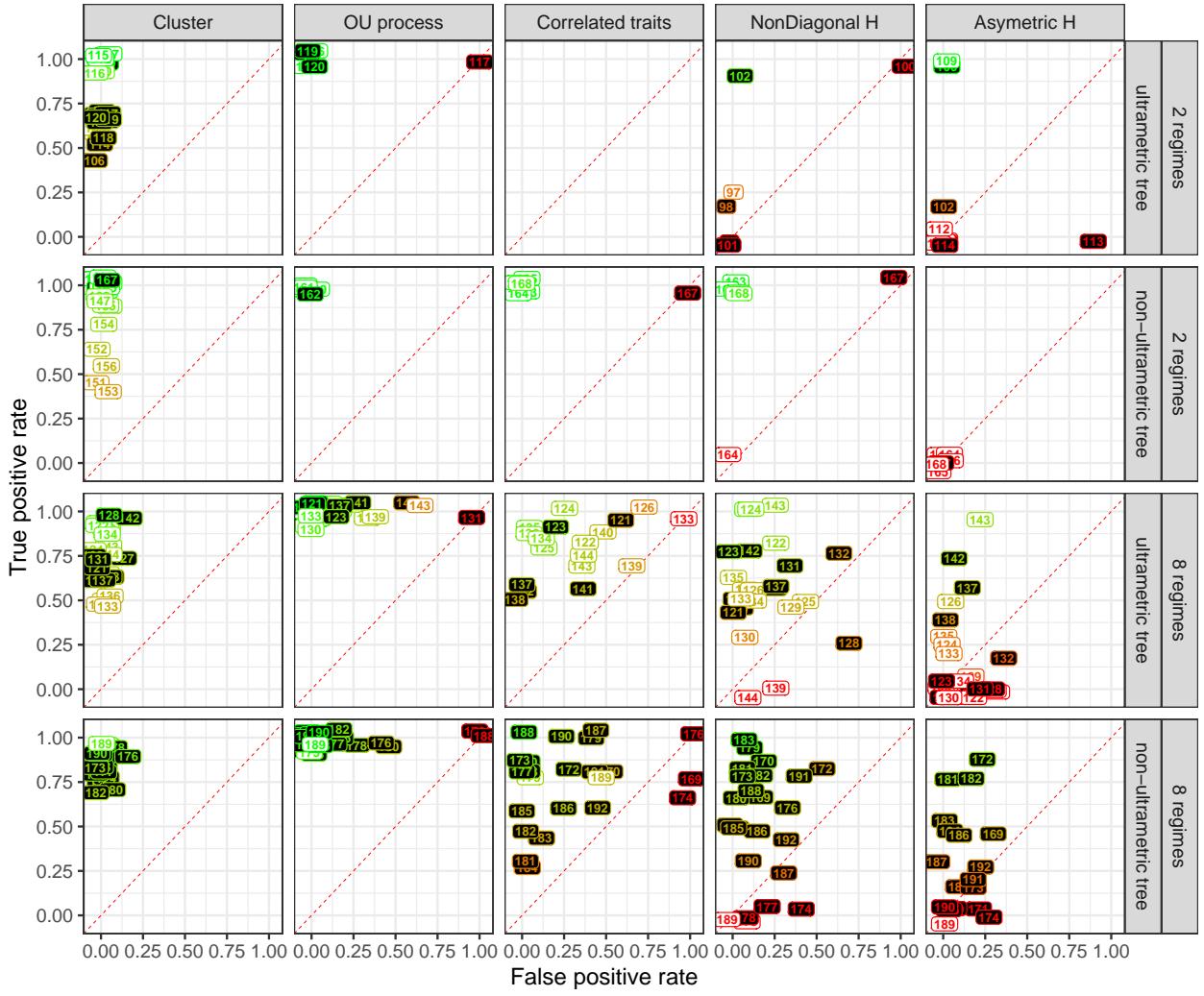
```



```

ggplot(testResultData[treeSize == "big" & !(crit2 %in% c("BM process", "Uncorrelated traits"))]) +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 0.2, color="red") +
  geom_label(aes(label = i, x = fpr, y = tpr,
                 #size = 0.5*apply(cbind((0.2+1-tpr), (0.2+fpr)), 1, max),
                 #color = 0.5*apply(cbind((0.2+1-tpr), (0.2+fpr)), 1, max),
                 fill = AIC_Final<AIC_True ),
                 size = 2,
                 position = position_jitter(width=0.05, height = 0.05),
                 label.padding = unit(0.1, "lines"), fontface = "bold") +
  xlab("False positive rate") + ylab("True positive rate") +
  scale_color_continuous(low="green", high="red") +
  scale_fill_manual(values = c("TRUE"="white", "FALSE"="black")) +
  #scale_size_continuous(range = c(1.5, 3)) +
  scale_x_continuous(breaks=c(0, 0.25, 0.5, 0.75, 1)) +
  scale_y_continuous(breaks=c(0, 0.25, 0.5, 0.75, 1)) +
  facet_grid(numClusters2*treeType2~crit2) +
  theme_bw() +
  theme(legend.position = "none")

```



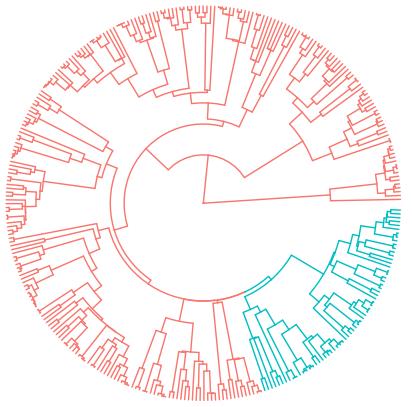
```

plList <- list(
  pl1 = PCMTreePlot(testData_t2$treeWithRegimes[[1]], layout="fan", open.angle=2, size=.25) +
    ggtitle("A. ultrametric / 2 regimes"),
  pl2 = PCMTreePlot(testData_t2$treeWithRegimes[[25]], layout="fan", open.angle=2, size=.25) +
    ggtitle("B. ultrametric / 8 regimes"),
  pl3 = PCMTreePlot(testData_t2$treeWithRegimes[[49]], layout="fan", open.angle=2, size=.25) +
    ggtitle("C. non-ultrametric / 2 regimes"),
  pl4 = PCMTreePlot(testData_t2$treeWithRegimes[[73]], layout="fan", open.angle=2, size=.25) +
    ggtitle("D. non-ultrametric / 8 regimes"),

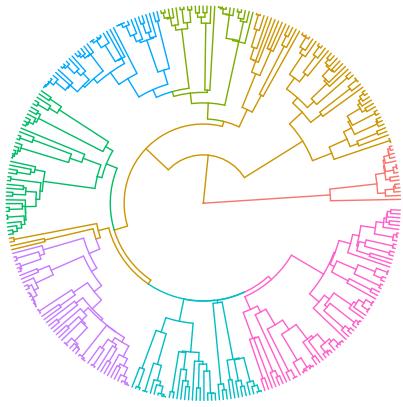
  pl5 = PCMTreePlot(testData_t2$treeWithRegimes[[97]], layout="fan", open.angle=2, size=.25) +
    ggtitle("A. ultrametric / 2 regimes"),
  pl6 = PCMTreePlot(testData_t2$treeWithRegimes[[121]], layout="fan", open.angle=2, size=.25) +
    ggtitle("B. ultrametric / 8 regimes"),
  pl7 = PCMTreePlot(testData_t2$treeWithRegimes[[145]], layout="fan", open.angle=2, size=.25) +
    ggtitle("C. non-ultrametric / 2 regimes"),
  pl8 = PCMTreePlot(testData_t2$treeWithRegimes[[169]], layout="fan", open.angle=2, size=.25) +
    ggtitle("D. non-ultrametric / 8 regimes")
)
cowplot:::plot_grid(plotlist = plList[1:4], nrow = 2, ncol=2)

```

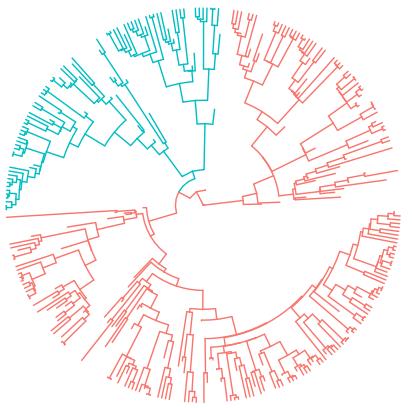
A. ultrametric / 2 regimes



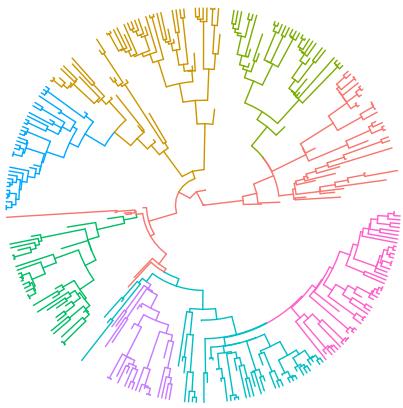
B. ultrametric / 8 regimes



C. non-ultrametric / 2 regimes

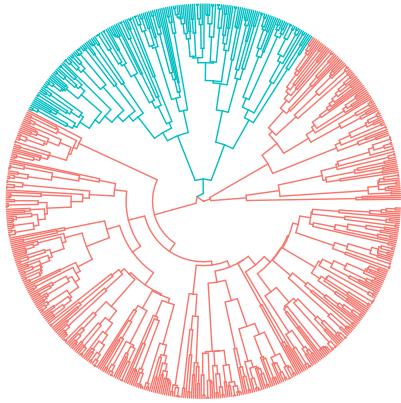


D. non-ultrametric / 8 regimes

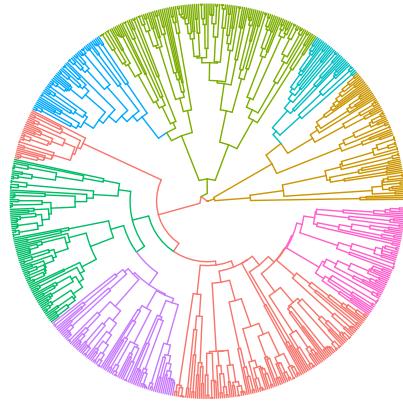


```
cowplot:::plot_grid(plotlist = plList[5:8], nrow = 2, ncol=2)
```

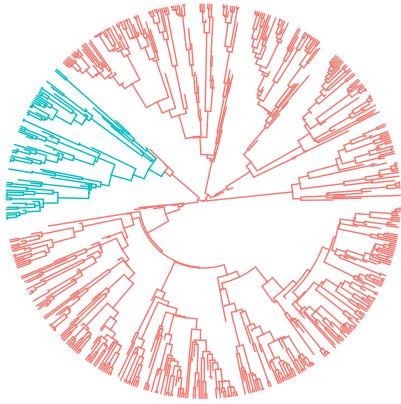
A. ultrametric / 2 regimes



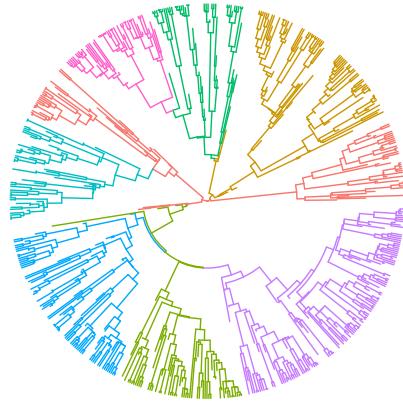
B. ultrametric / 8 regimes



C. non-ultrametric / 2 regimes



D. non-ultrametric / 8 regimes



```
data <- rbindlist(lapply(1:nrow(testData_t2), function(i) {
  pl <- PCMPlotTraitData2D(
    X = testData_t2$X[[i]][, 1:PCMTreeNumTips(testData_t2$treeWithRegimes[[i]])],
    tree = testData_t2$treeWithRegimes[[i]])
  cbind(pl$data, testData_t2[i], list(
    rowId = c(i, rep(as.integer(NA), nrow(pl$data)-1)),
    x0 = c(1.0, rep(as.double(NA), nrow(pl$data) - 1)),
    y0 = c(-1.0, rep(as.double(NA), nrow(pl$data) - 1)),
    treeType, treeSize, numClusters, clusterNodes,
    mapping,
    logLik = c(logLik[[1]], rep(as.double(NA), nrow(pl$data)-1)),
    AIC = c(AIC[[1]], rep(as.double(NA), nrow(pl$data)-1)),
    nobs = c(nobs[[1]], rep(as.integer(NA), nrow(pl$data)-1)),
    df = c(df[[1]], rep(as.integer(NA), nrow(pl$data)-1)),
    IdMappingForClustering, IdParamForMapping, IdSimulationForParam)))
}))
```



```
data[, labLogLik:=sapply(logLik, function(l1) if(is.na(l1)) as.character(NA) else paste0("L: ", round(l1, 2)))
```

```
data[, labAIC:=sapply(AIC, function(a) if(is.na(a)) as.character(NA) else paste0("AIC: ", round(a, 2))))
```

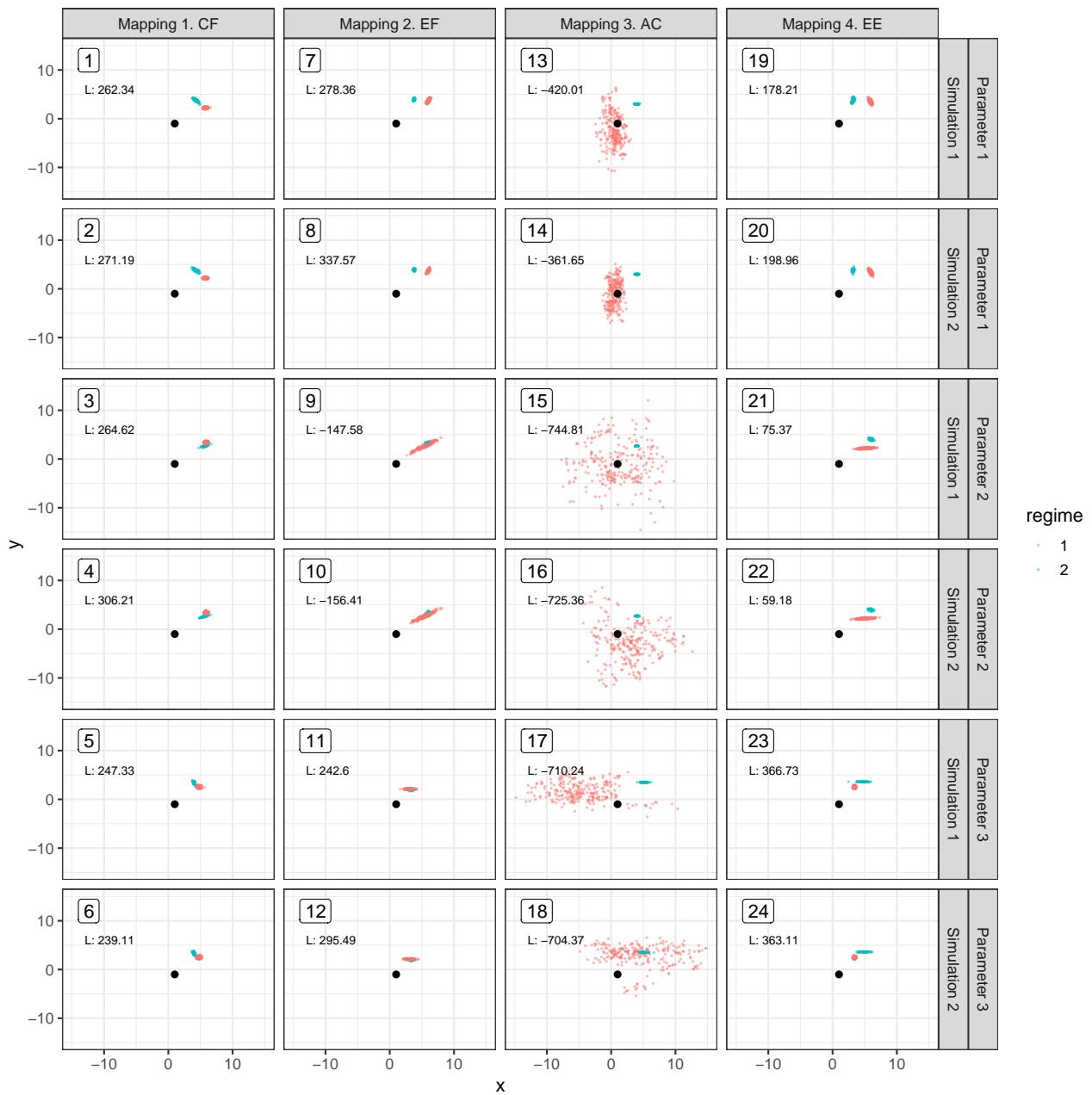
```

data[, IdMappingLETTERS:=
  paste0(IdMappingForClustering, ". ",
         sapply(mapping, function(m) do.call(paste0, as.list(LETTERS[m]))))]

xlim <- c(-15, 15)
ylim <- c(-15, 15)

ggplot(data[treeType == "ultrametric" & treeSize=="small" & numClusters==2], aes(x, y)) +
  geom_point(aes(color = regime), size = .1, alpha = .5) +
  geom_point(aes(x=x0, y=y0)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  #geom_text(aes(x=-12, y=2, label = labAIC), size = 3, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)~paste0
  theme_bw()

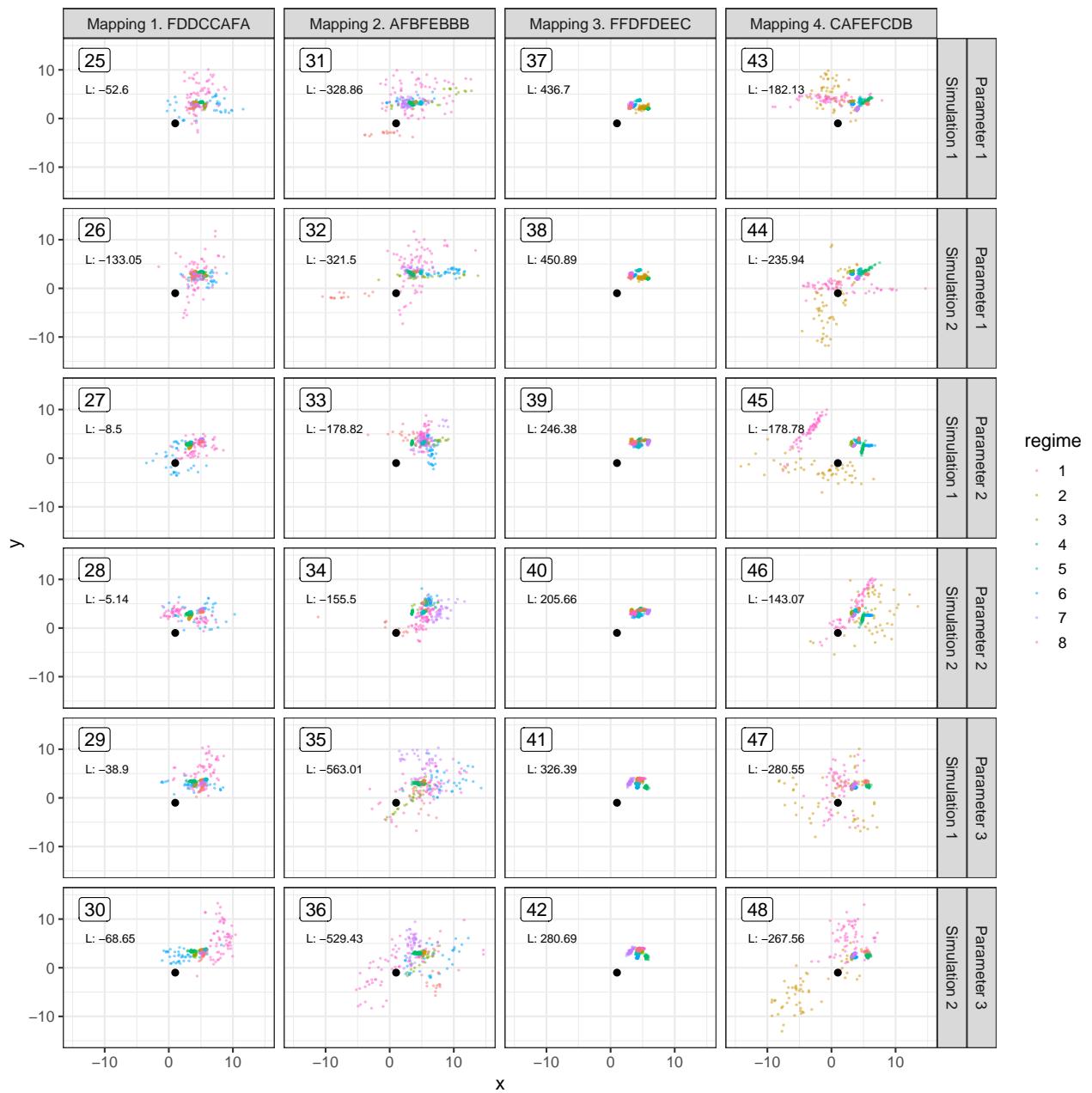
```



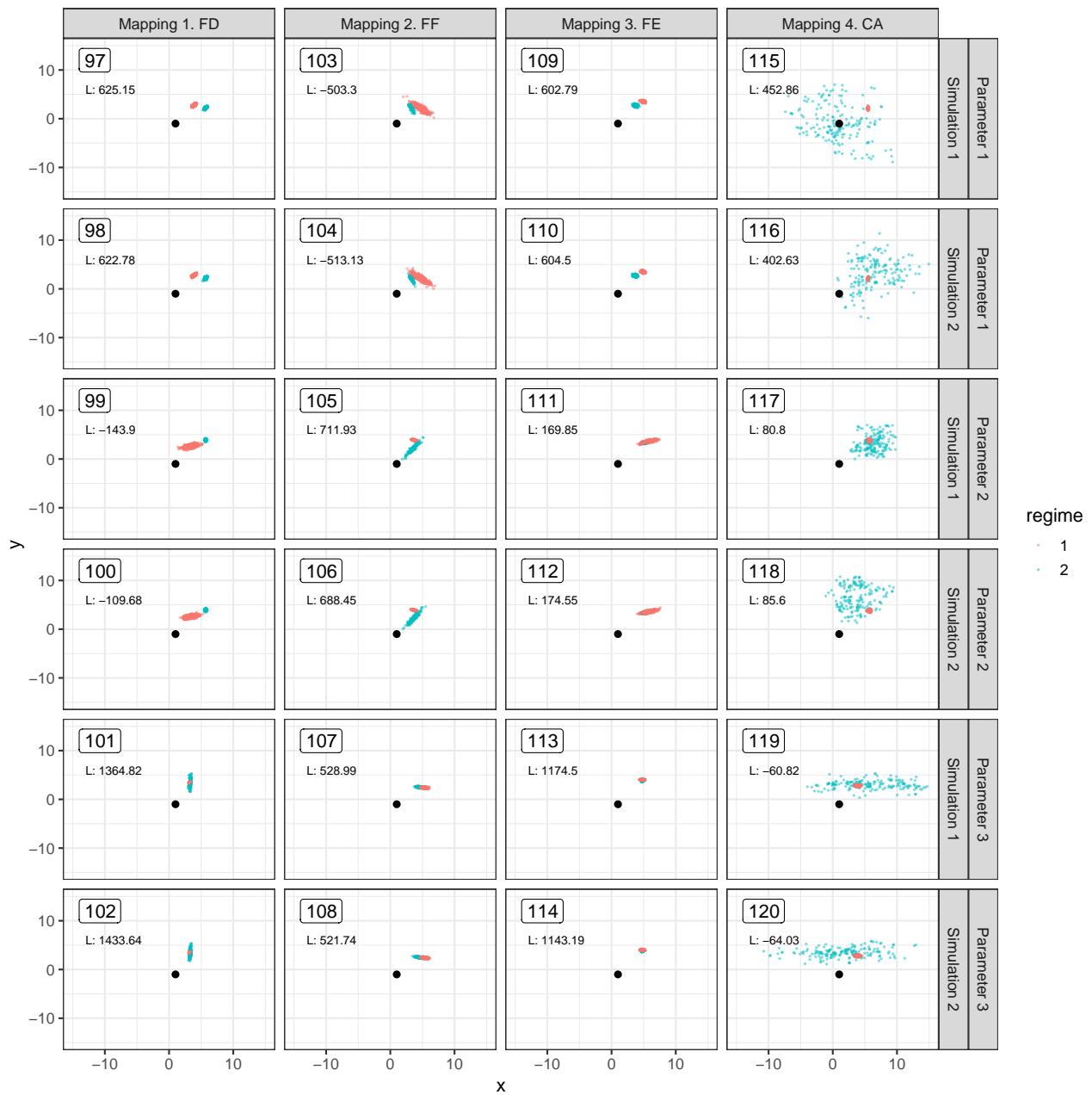
```

ggplot(data[treeType == "ultrametric" & treeSize=="small" & numClusters==8], aes(x, y)) +
  geom_point(aes(color = regime), size = .1, alpha = .5) +
  geom_point(aes(x=x0, y=y0)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)-paste0(
    theme_bw()
  )

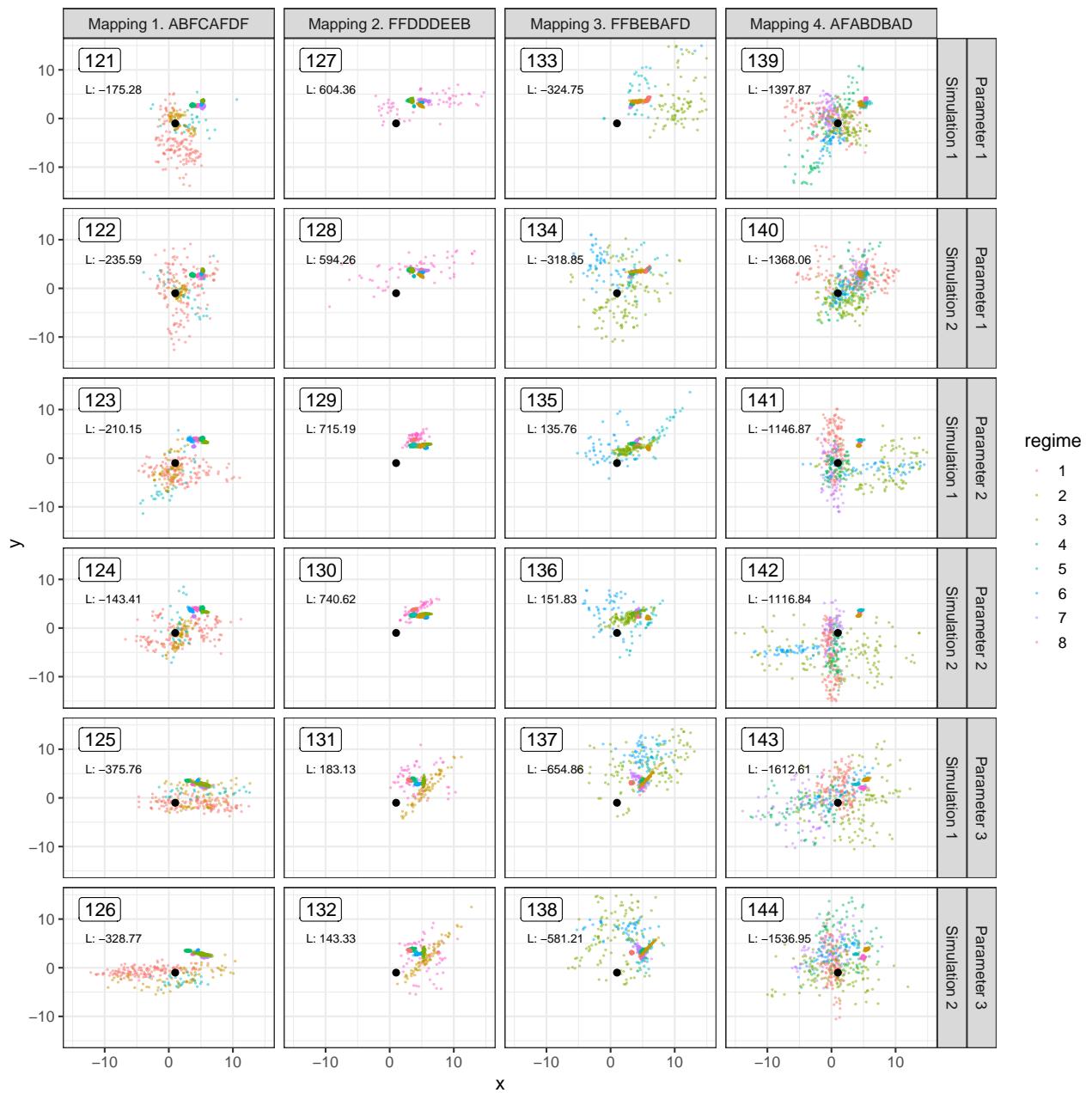
```



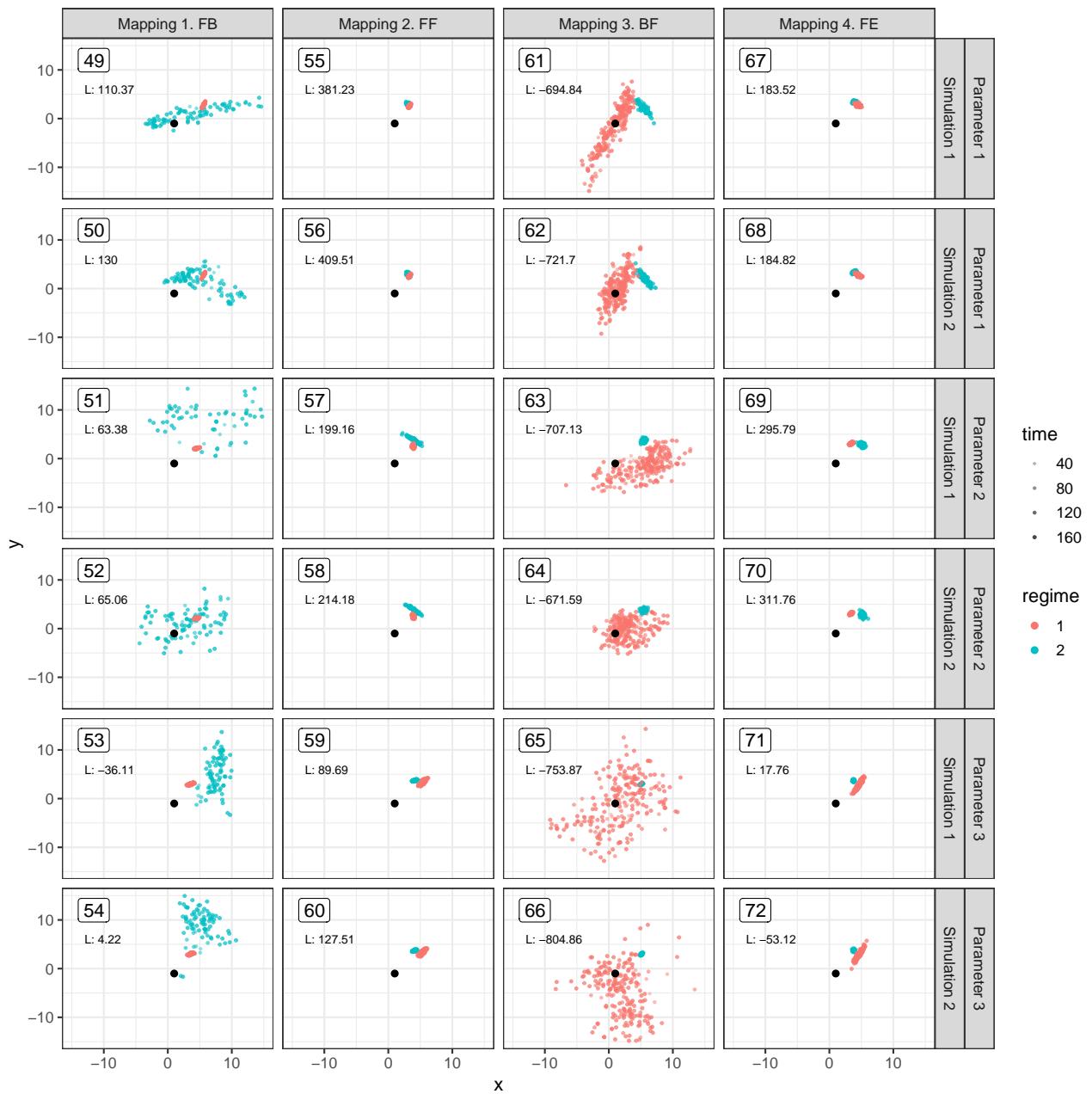
```
ggplot(data[treeType == "ultrametric" & treeSize=="big" & numClusters==2], aes(x, y)) +
  geom_point(aes(color = regime), size = .1, alpha = .5) +
  geom_point(aes(x=x0, y=y0)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)-paste0(
    theme_bw()
```



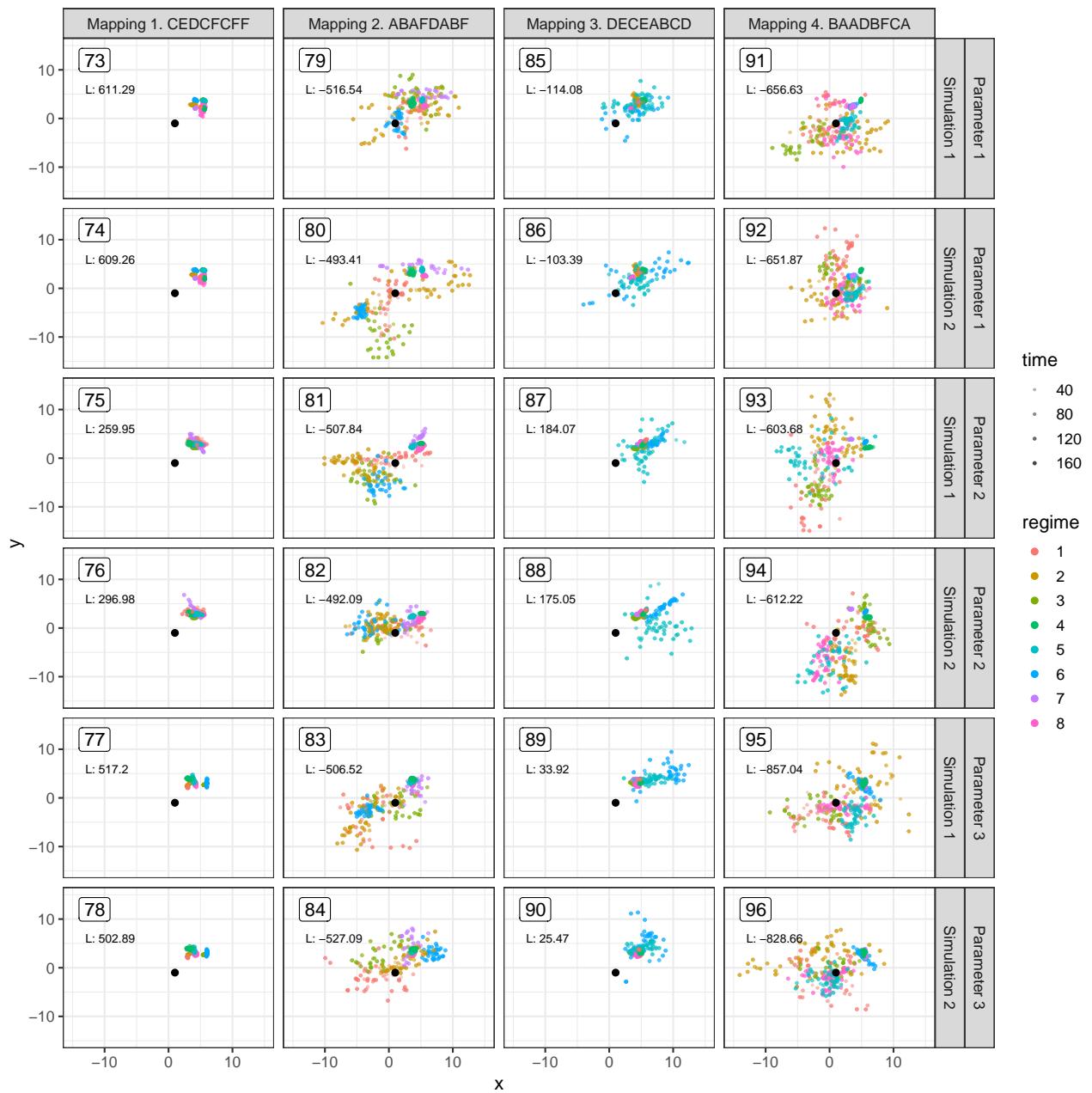
```
ggplot(data[data$treeType == "ultrametric" & treeSize=="big" & numClusters==8], aes(x, y)) +
  geom_point(aes(color = regime), size = .1, alpha = .5) +
  geom_point(aes(x=x0, y=y0)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)-paste0(
    theme_bw()
```



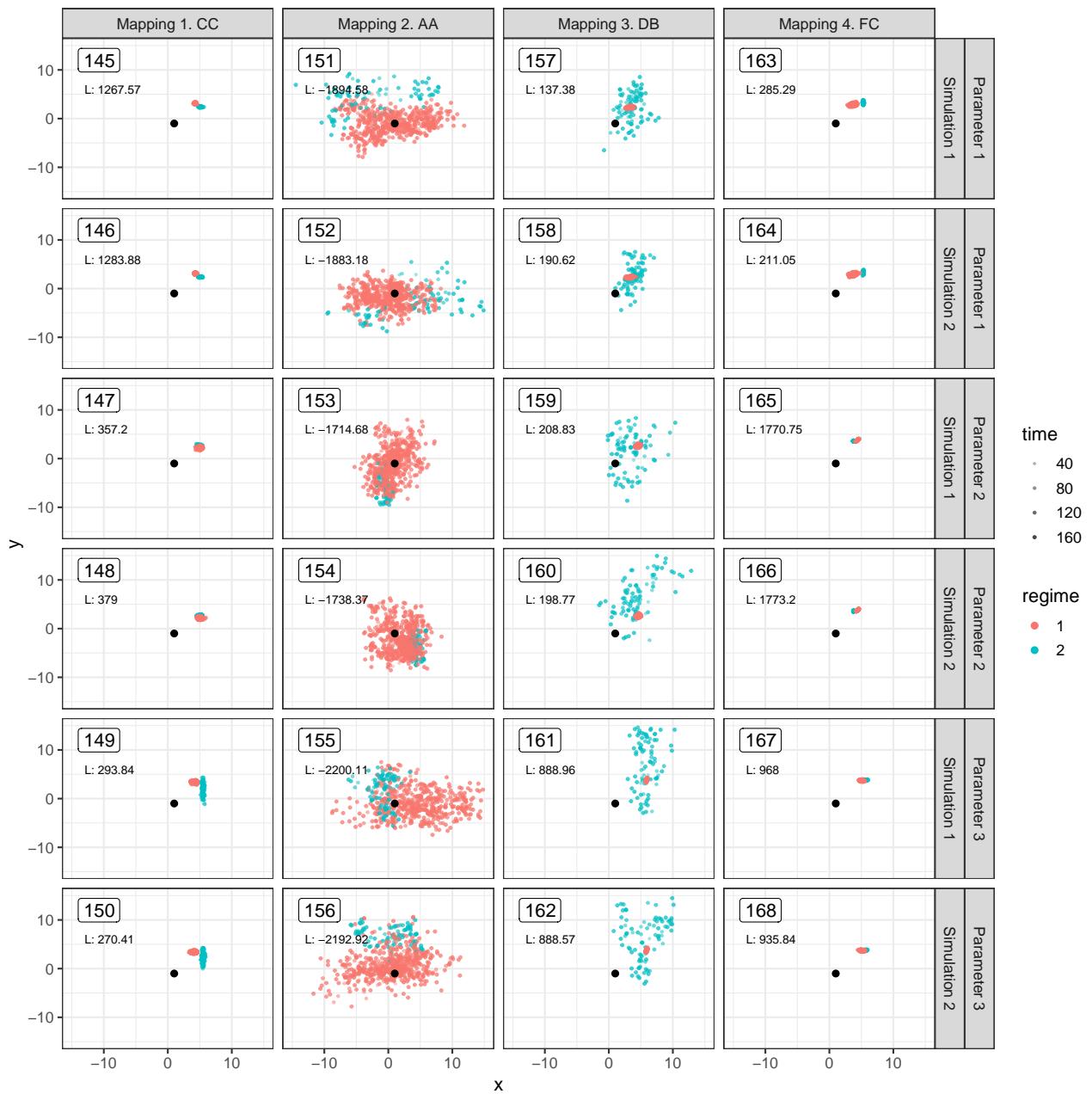
```
ggplot(data[treeType == "non-ultrametric" & treeSize=="small" & numClusters==2], aes(x, y)) +
  geom_point(aes(color = regime, size=time, alpha = time)) +
  geom_point(aes(x=x0, y=y0)) +
  scale_size_continuous(range = c(0.05, .5)) +
  scale_alpha_continuous(range = c(0.2, 0.75)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)~paste0
  theme_bw()
```



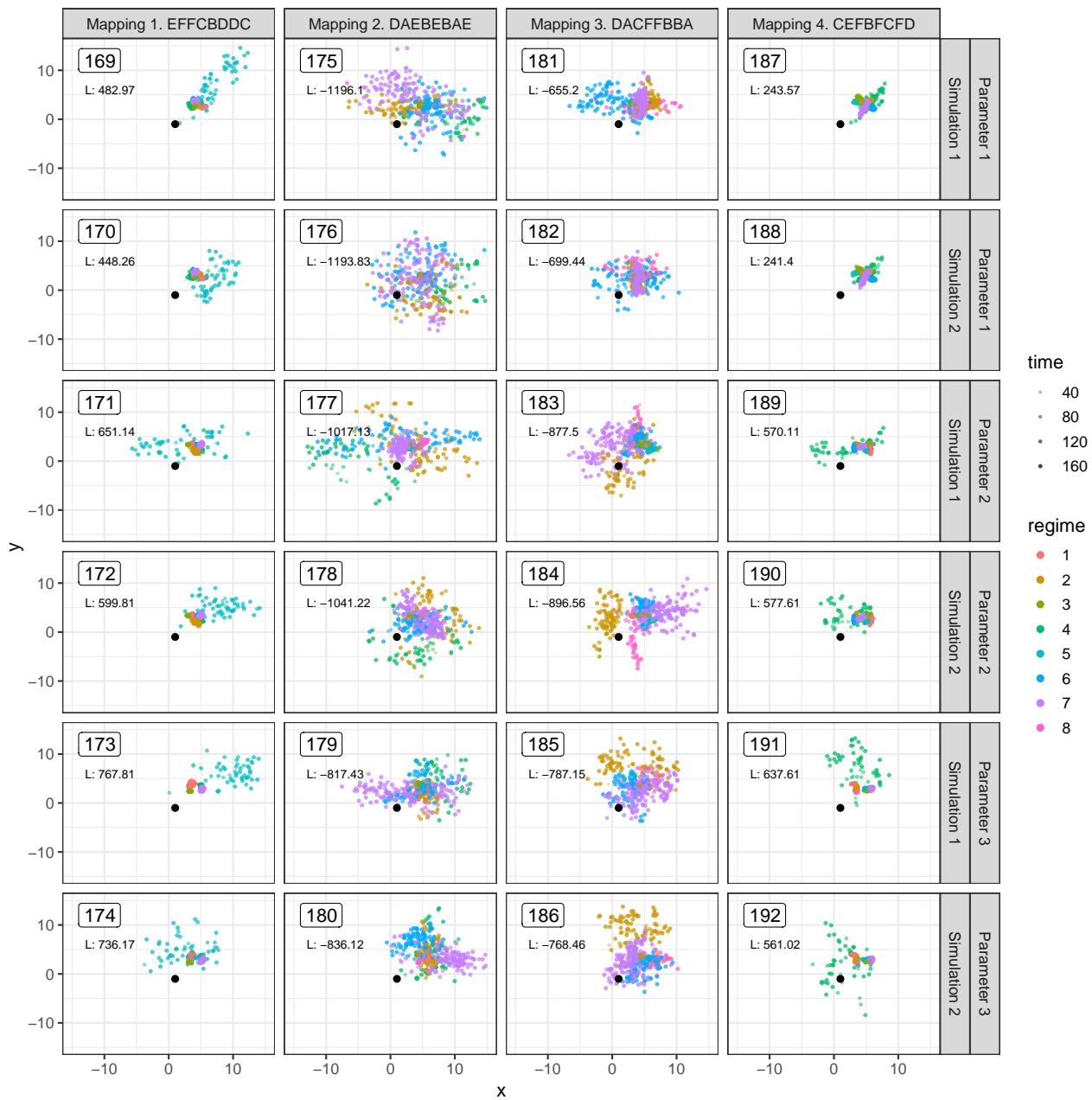
```
ggplot(data[treeType == "non-ultrametric" & treeSize=="small" & numClusters==8], aes(x, y)) +
  geom_point(aes(color = regime, size=time, alpha = time)) +
  geom_point(aes(x=x0, y=y0)) +
  scale_size_continuous(range = c(0.05, .5)) +
  scale_alpha_continuous(range = c(0.2, 0.75)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)~paste0
  theme_bw()
```



```
ggplot(data[treeType == "non-ultrametric" & treeSize=="big" & numClusters==2], aes(x, y)) +
  geom_point(aes(color = regime, size=time, alpha = time)) +
  geom_point(aes(x=x0, y=y0)) +
  scale_size_continuous(range = c(0.05, .5)) +
  scale_alpha_continuous(range = c(0.2, 0.75)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)~paste0
  theme_bw()
```



```
ggplot(data[treeType == "non-ultrametric" & treeSize=="big" & numClusters==8], aes(x, y)) +
  geom_point(aes(color = regime, size=time, alpha = time)) +
  geom_point(aes(x=x0, y=y0)) +
  scale_size_continuous(range = c(0.05, .5)) +
  scale_alpha_continuous(range = c(0.2, 0.75)) +
  scale_x_continuous(limits = xlim) +
  scale_y_continuous(limits = ylim) +
  geom_label(aes(x=-14, y=12, label = rowId), hjust = "left") +
  geom_text(aes(x=-13, y=6, label = labLogLik), size = 2.5, hjust = "left") +
  facet_grid(paste0("Parameter ", IdParamForMapping)+paste0("Simulation ", IdSimulationForParam)~paste0
  theme_bw()
```



```

load("../data-raw/ResultsTestData_t2/TestResultData_t2.RData")
summaryTable <- testResultData[, {
  maskTests <- !is.na(fpr) & !is.na(tpr)
  numTests <- sum(maskTests)
  list(
    `#tests` = numTests,
    `Better AIC` = sum(AIC_Final[maskTests] - AIC_True[maskTests] <= 0)/numTests,
    fpr = sum(fpr[maskTests])/numTests,
    #`SE(fpr)` = sd(fpr[maskTests])/sqrt(numTests),
    tpr = sum(tpr[maskTests])/numTests,
    #`SE(tpr)` = sd(tpr[maskTests])/sqrt(numTests)
  )
},
keyby=list(Crit.=crit2, N=factor(treeSize, levels=c("small", "big")), labels=c(318, 638)),

```

```
`#regimes` = as.integer(numClusters),  
`Tree-type` = factor(treeType, levels=c("ultrametric", "non-ultrametric")))] [Crit. %in% c("Clus  
print.xtable(xtable(summaryTable), comment=FALSE, include.rownames = FALSE)
```