

Generated Figures

Venelin Mitov

3 Jan 2018

Collect inferred models

Here, we extract the inferred model and regimes and add them as columns to the testData_t5 data.table.
Manual execution.

Evaluate the 7 binary criterions on the inferred models

```
if(file.exists("../data-raw/ResultsTestData_t5/TestResultData_t5.RData")) {
  load("../data-raw/ResultsTestData_t5/TestResultData_t5.RData")
} else {
  testResultData <- NULL
}

testResultDataBhatt <- rbindlist(
  lapply(1:nrow(testData_t5), function(i) {

    resultFile <- paste0("../data-raw/ResultsTestData_t5/FinalResult_MixedGaussian_testData_t5_id_",
                          i)

    if(file.exists(resultFile)) {
      cat("Loading ", resultFile, "\n")
      load(resultFile)
      bestFitAIC <- RetrieveBestFitAIC(fitMappings)
      trueFromTestData <- RetrieveTrueFromTestData(testData_t5, i)

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

      cat("Bhatt: ", bhatt, "\n")
      data.table(i = i,
                 treeType=testData_t5[i, treeType],
                 treeSize=testData_t5[i, treeSize],
                 numClusters = testData_t5[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_t5/TestResultData_t5.RData")
testResultData[, numClusters2:=paste0(numClusters, " regimes")]
testResultData[, treeType2:=factor(paste0(treeType, " tree"), levels=c("ultrametric tree", "non-ultrametric tree"))]
ggplot(testResultData[testSize == "N=80" & !(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[testSize == "N=159" & !(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")

ids <- testData_t5[1:256, list(id = min(.I), .N), keyby=list(treeType, treeSize, numClusters)][, sort(id)

for(id in ids) {
  print(data.table(
    id = id,
    type = testData_t5$treeType[[id]],
    size = testData_t5$treeSize[[id]],
    numRegimes = testData_t5$numClusters[[id]]))
  tree <- testData_t5$treeWithRegimes[[id]]
  print(PCMTreeDtNodeRegimes(tree)[endNode <= PCMTreeNumTips(tree), .N, keyby=regime])
}

##      id      type size numRegimes
## 1: 1 ultrametric N=80          2
##   regime N
## 1: 1 52
## 2: 2 28
##      id      type size numRegimes
## 1: 129 ultrametric N=80          3
##   regime N
## 1: 1 31
## 2: 2 21
## 3: 3 28

ids <- testData_t5[1:256, list(id = min(.I), .N), keyby=list(treeType, treeSize)][, sort(id)]

dtNumAllPartitions <- rbindlist(lapply(ids, function(id) {
  data.table(
    id = id,
    type = testData_t5$treeType[[id]],
    size = testData_t5$treeSize[[id]],
    P18 = data.table(a=sapply(PCMTreeListAllPartitions(
      testData_t5$tree[[id]], 18), length) + 1)[
        , .N, keyby=a][, sum(N*6^a)],
    P19 = data.table(a=sapply(PCMTreeListAllPartitions(
      testData_t5$tree[[id]], 19), length) + 1)[
        , .N, keyby=a][, sum(N*6^a)],
    P20 = data.table(a=sapply(PCMTreeListAllPartitions(
      testData_t5$tree[[id]], 20), length) + 1)[
        , .N, keyby=a][, sum(N*6^a)],
    P21 = data.table(a=sapply(PCMTreeListAllPartitions(
      testData_t5$tree[[id]], 21), length) + 1)[
        , .N, keyby=a][, sum(N*6^a)],
    P22 = data.table(a=sapply(PCMTreeListAllPartitions(

```

```

    testData_t5$tree[[id]], 22), length) + 1)[
    , .N, keyby=a] [, sum(N*6^a)],
P23 = data.table(a=sapply(PCMTreeListAllPartitions(
    testData_t5$tree[[id]], 23), length) + 1)[
    , .N, keyby=a] [, sum(N*6^a)]
)
}))

save(dtNumAllPartitions, file = "dtNumAllPartitions.RData")

load("dtNumAllPartitions.RData")
dtNumAllPartitions

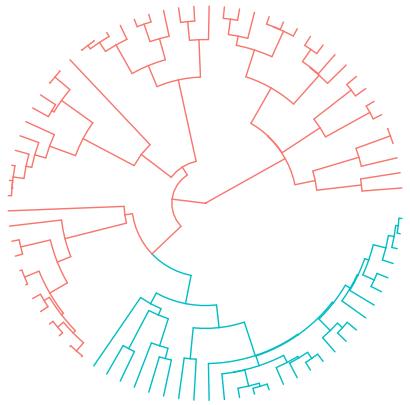
##      id      type  size      P18      P19      P20      P21      P22
## 1:   1 ultrametric N=80  24054  17358  1050  582  114
## 2:  65 non-ultrametric N=80   5550   4902  4254  3822 2922
## 3: 129      ultrametric N=159 85719534 20672574 3862086 3862086 688110
## 4: 193 non-ultrametric N=159 203397990 89043666 56089842 10719654 3108858
##      P23
## 1:    78
## 2:  2238
## 3: 62718
## 4: 547278

ids <- testData_t5[, list(id = min(.I), .N), keyby=list(treeType, treeSize, numClusters)][, sort(id)]

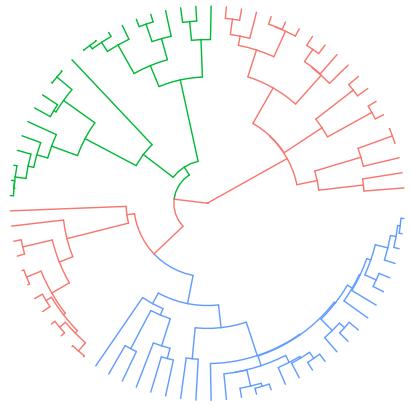
plList <- lapply(ids, function(id) {
  PCMTreePlot(testData_t5$treeWithRegimes[[id]], layout="fan", open.angle=2, size=.25) +
    ggtitle(paste0(LETTERS[match(id, ids)], ".",
                  testData_t5$treeType[[id]], " / ",
                  testData_t5$treeSize[[id]], " / ",
                  testData_t5$numClusters[[id]], " regimes"))
})
cowplot:::plot_grid(plotlist = plList[1:4], nrow = 2, ncol=2)

```

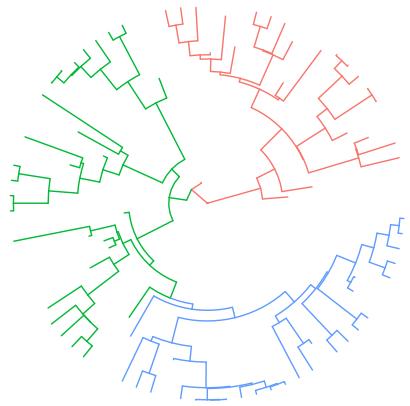
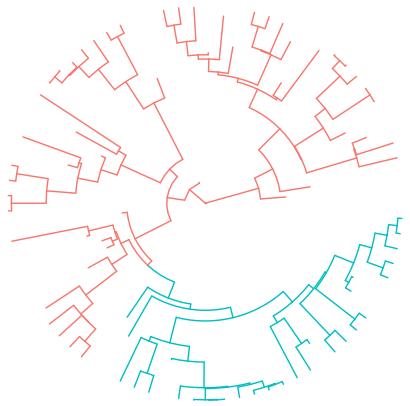
A. ultrametric / N=80 / 2 regimes



B. ultrametric / N=80 / 3 regimes

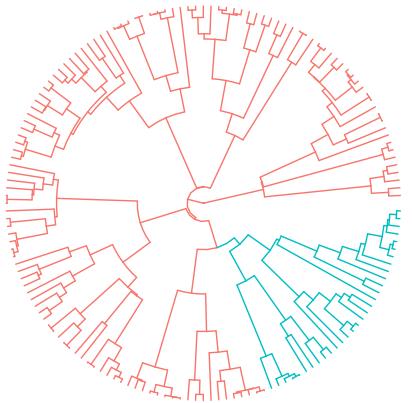


C. non-ultrametric / N=80 / 2 regime D. non-ultrametric / N=80 / 3 regime

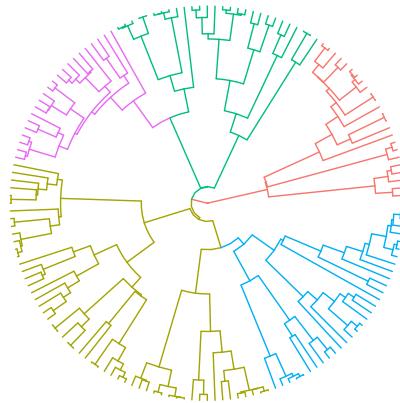


```
cowplot:::plot_grid(plotlist = plList[5:8], nrow = 2, ncol=2)
```

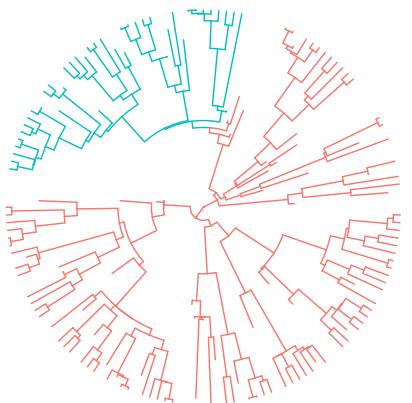
E. ultrametric / N=159 / 2 regimes



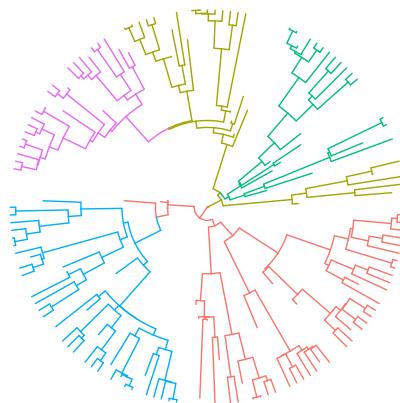
F. ultrametric / N=159 / 5 regimes



G. non-ultrametric / N=159 / 2 regim

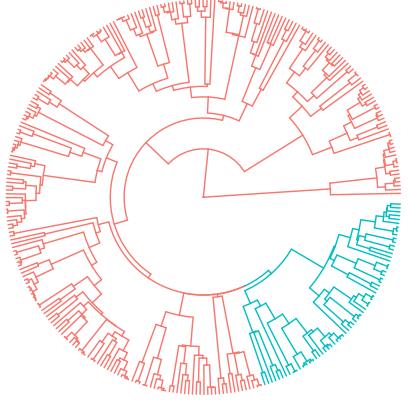


H. non-ultrametric / N=159 / 5 regi

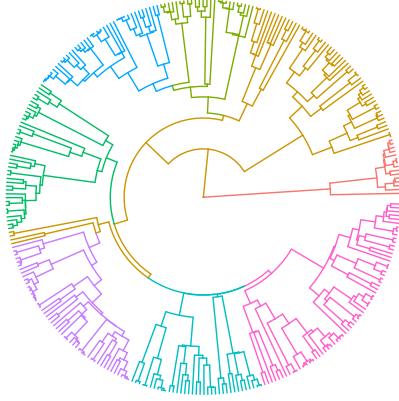


```
cowplot:::plot_grid(plotlist = plList[9:12], nrow = 2, ncol=2)
```

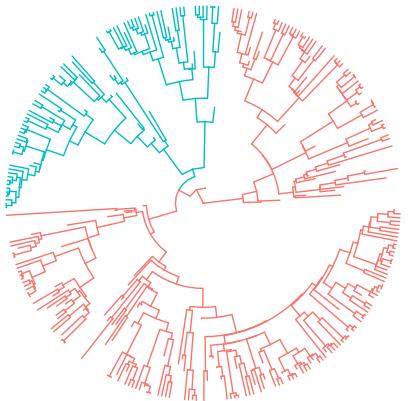
I. ultrametric / N=318 / 2 regimes



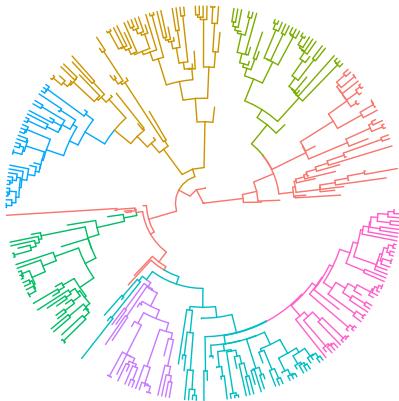
J. ultrametric / N=318 / 8 regimes



K. non-ultrametric / N=318 / 2 regimes

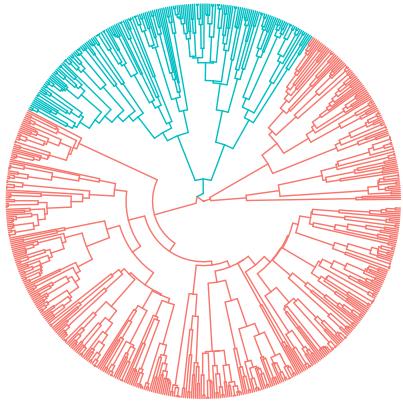


L. non-ultrametric / N=318 / 8 regimes

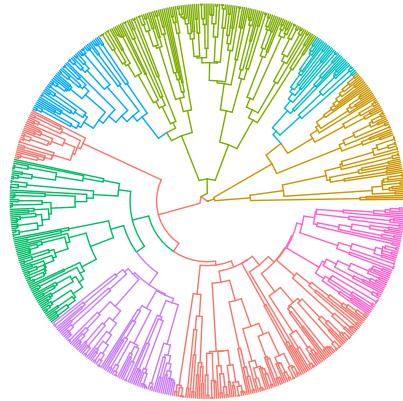


```
cowplot:::plot_grid(plotlist = plList[13:16], nrow = 2, ncol=2)
```

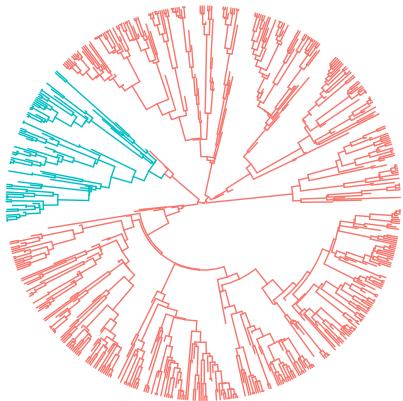
M. ultrametric / N=638 / 2 regimes



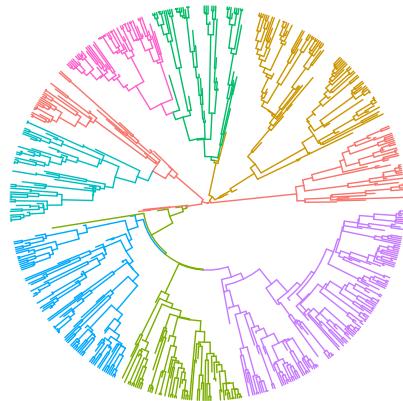
N. ultrametric / N=638 / 8 regimes



O. non-ultrametric / N=638 / 2 regim



P. non-ultrametric / N=638 / 8 regim



```

data <- rbindlist(lapply(1:nrow(testData_t5), function(i) {
  pl <- PCMPlotTraitData2D(
    X = testData_t5$X[[i]][, 1:PCMTreeNumTips(testData_t5$treeWithRegimes[[i]])],
    tree = testData_t5$treeWithRegimes[[i]])
  cbind(pl$data, testData_t5[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)))
}))
```

```

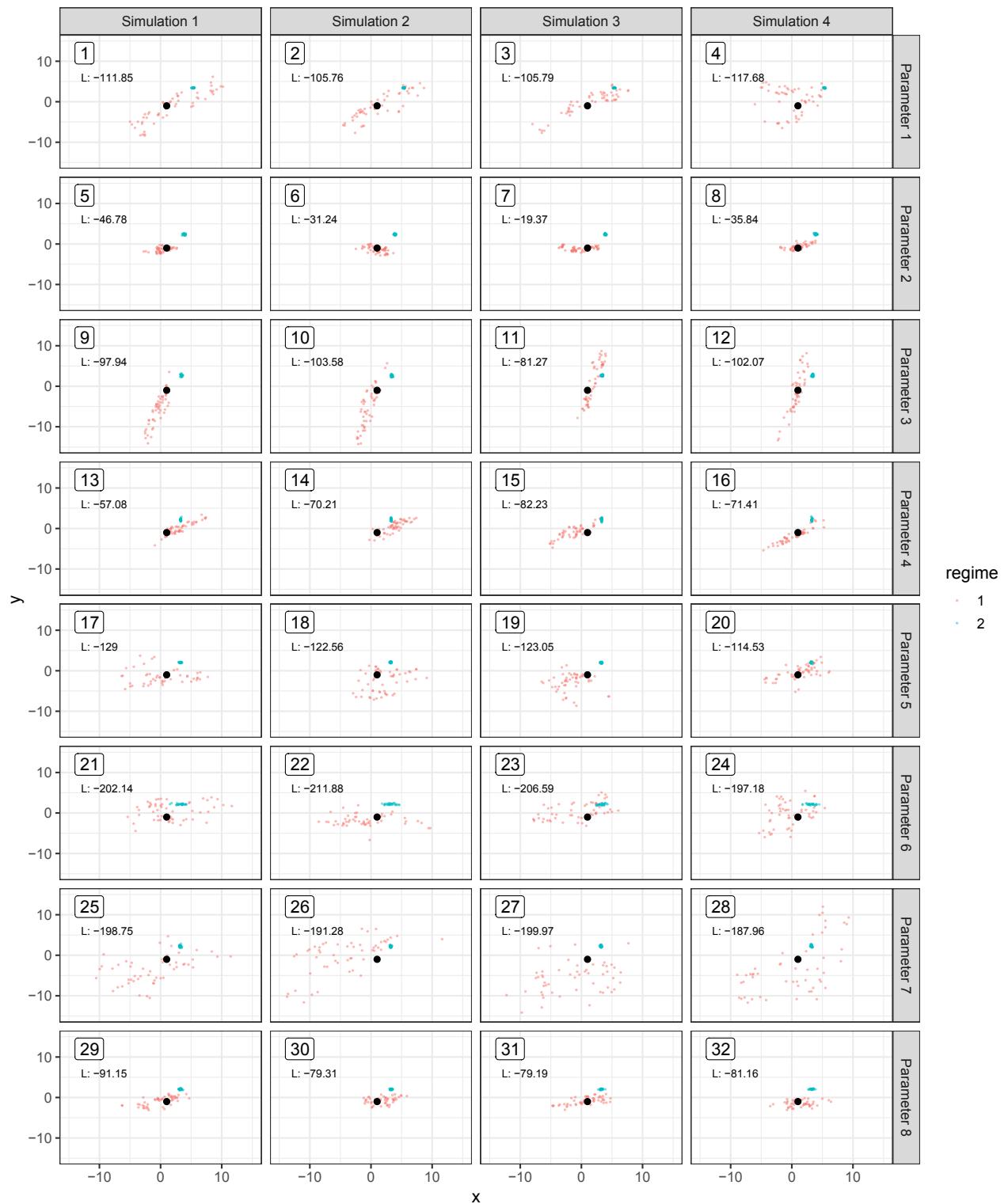
data[, labLogLik:=sapply(logLik, function(l1) if(is.na(l1)) as.character(NA) else paste0("L: ", round(l1, 2)))
```

```

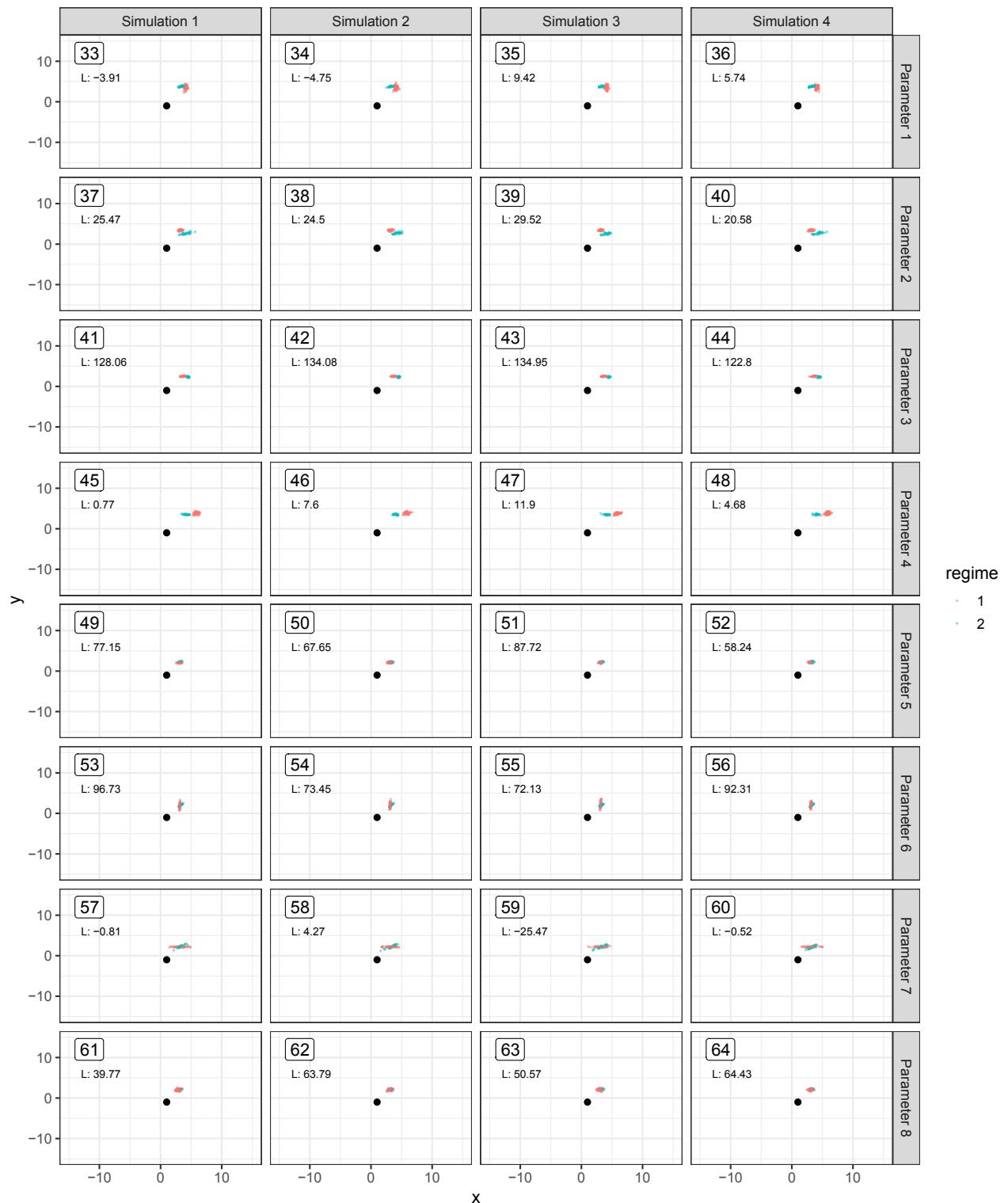
data[, labAIC:=sapply(AIC, function(a) if(is.na(a)) as.character(NA) else paste0("AIC: ", round(a, 2))))
```

```
data[, IdMappingLETTERS:=  
  paste0(IdMappingForClustering, ". ",  
    sapply(mapping, function(m) do.call(paste0, as.list(LETTERS[m]))))]  
  
xlim <- c(-15, 15)  
ylim <- c(-15, 15)
```

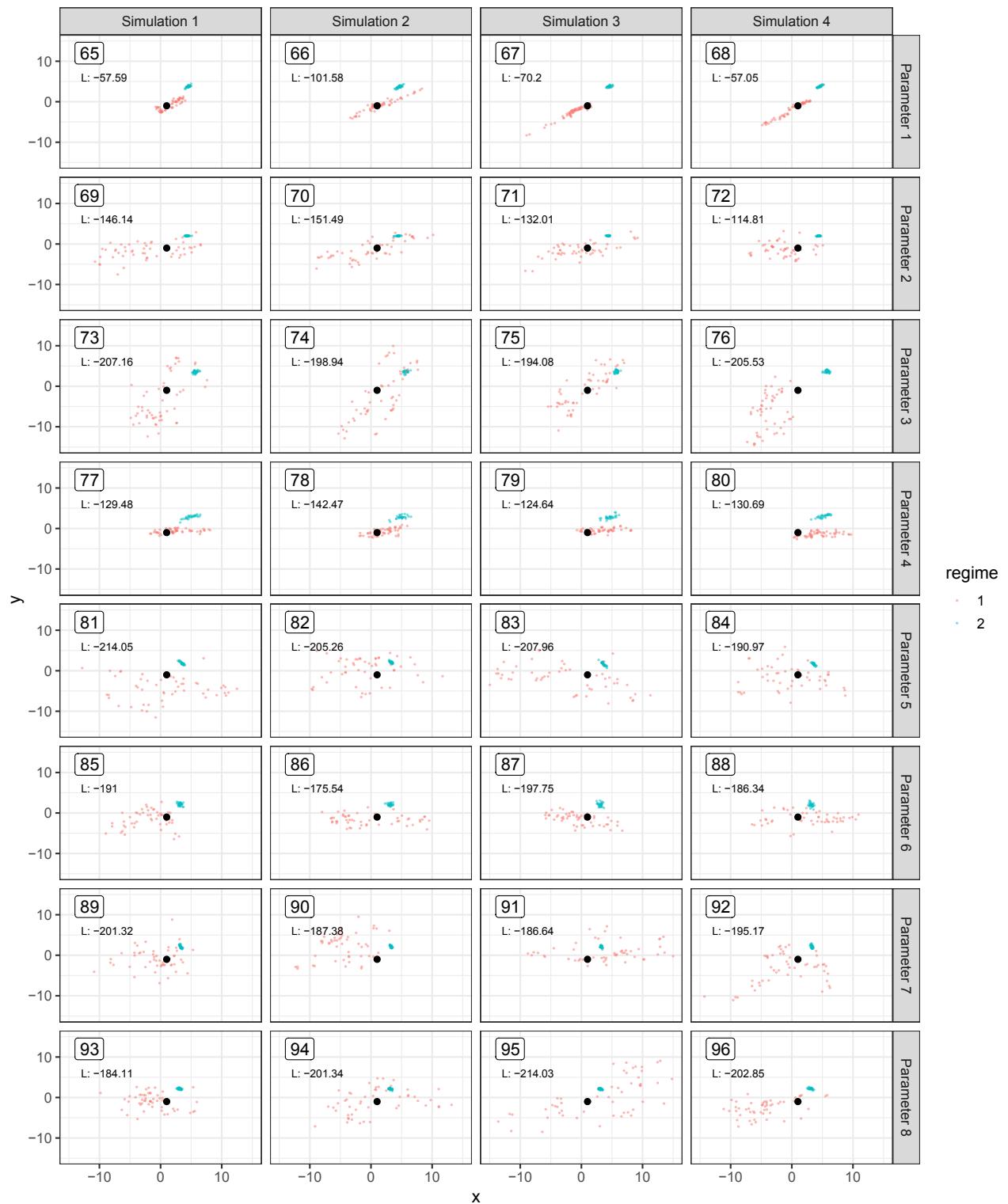
Tree-type ultrametric / N=80 / 2 regimes / Mapping 1. BC



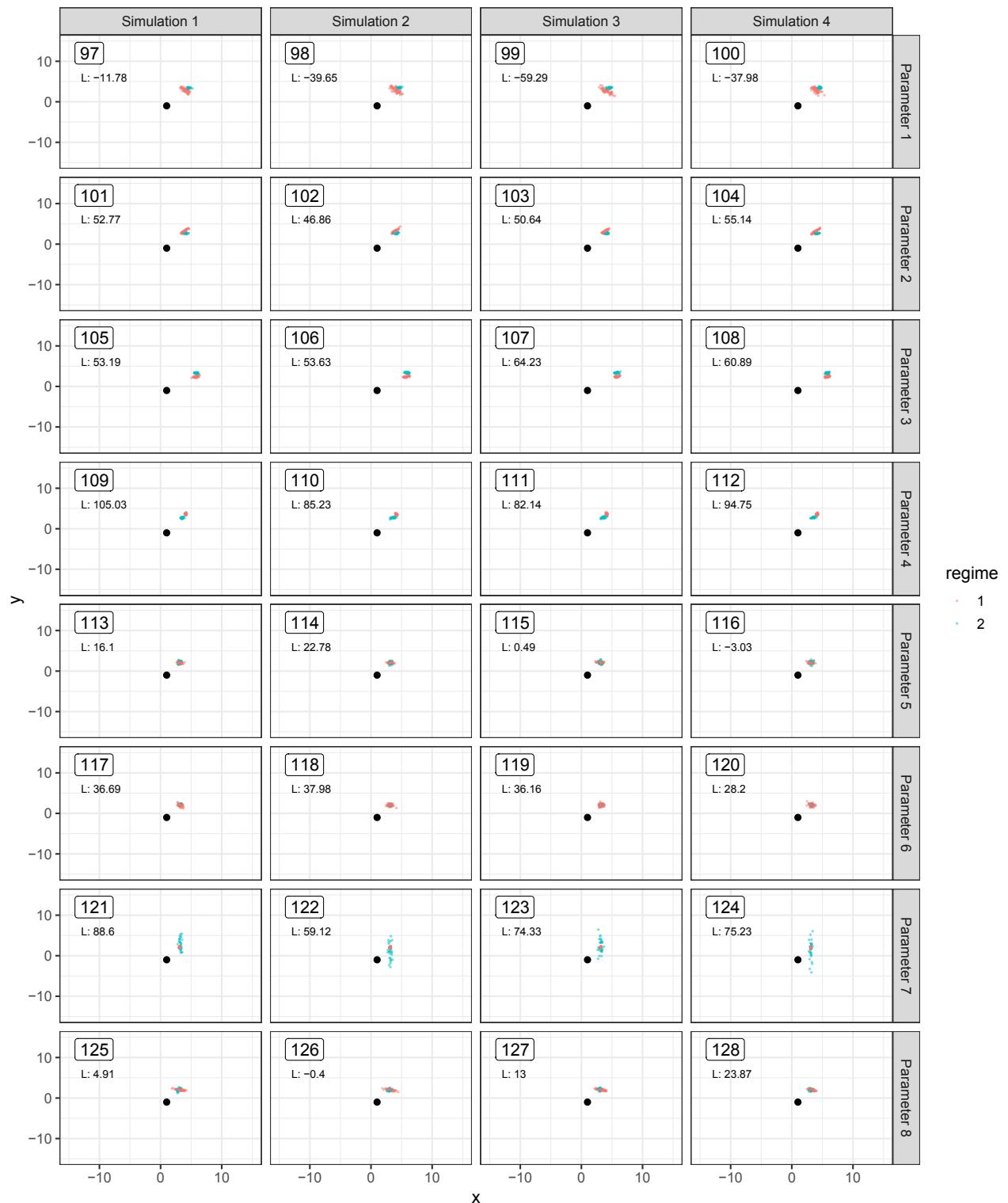
Tree-type ultrametric / N=80 / 2 regimes / Mapping 2. DF



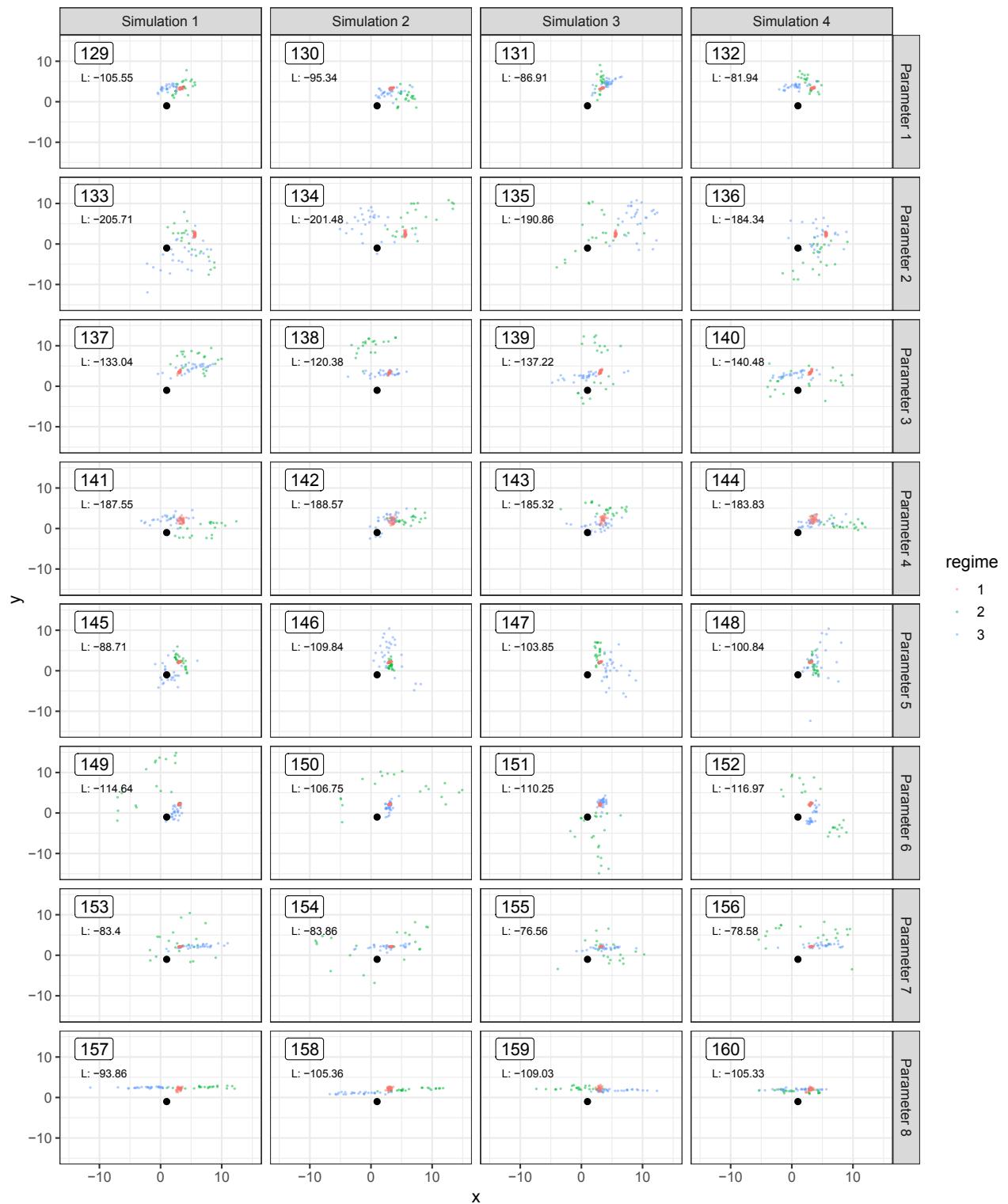
Tree-type ultrametric / N=80 / 2 regimes / Mapping 3. BF



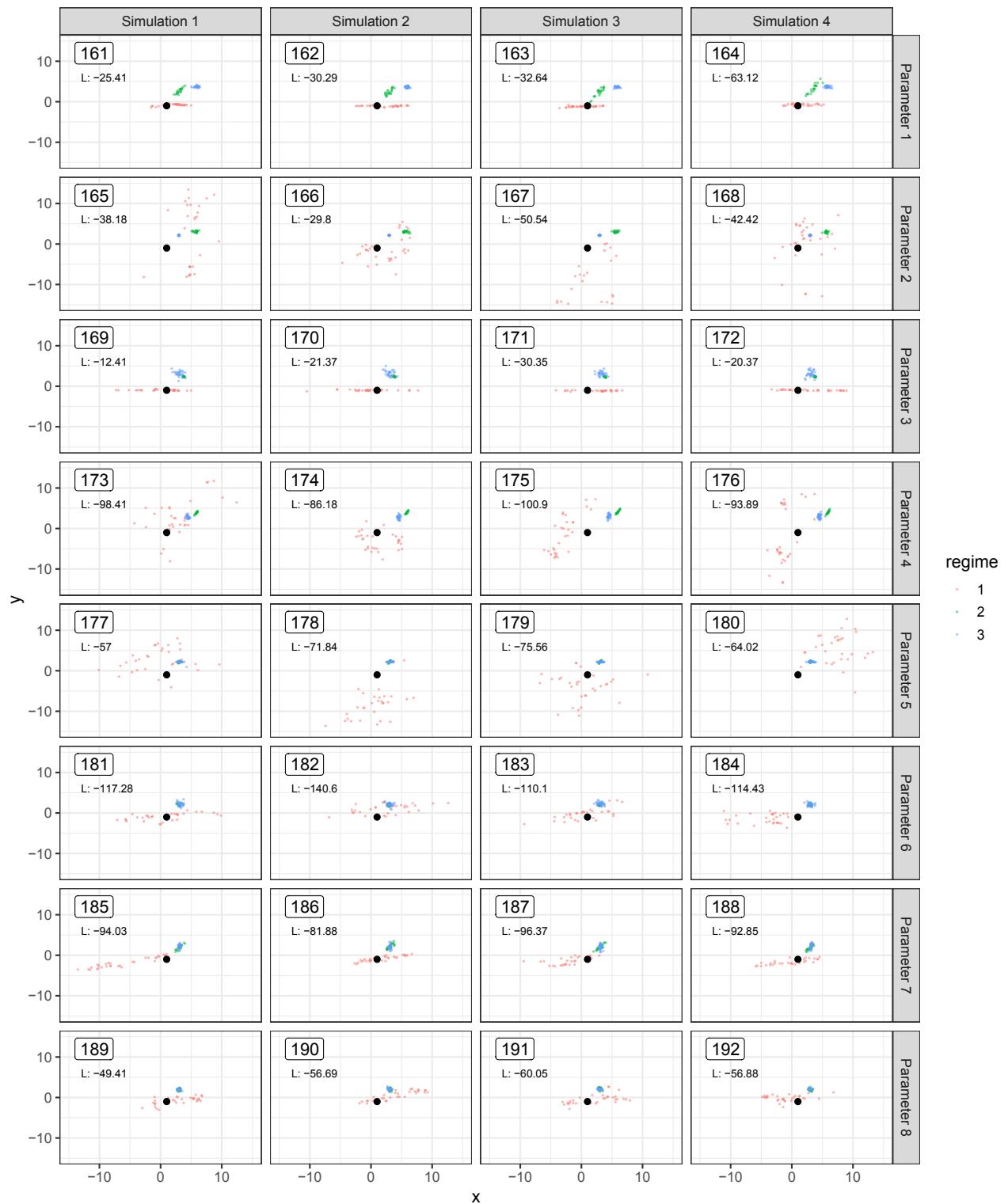
Tree-type ultrametric / N=80 / 2 regimes / Mapping 4. FD



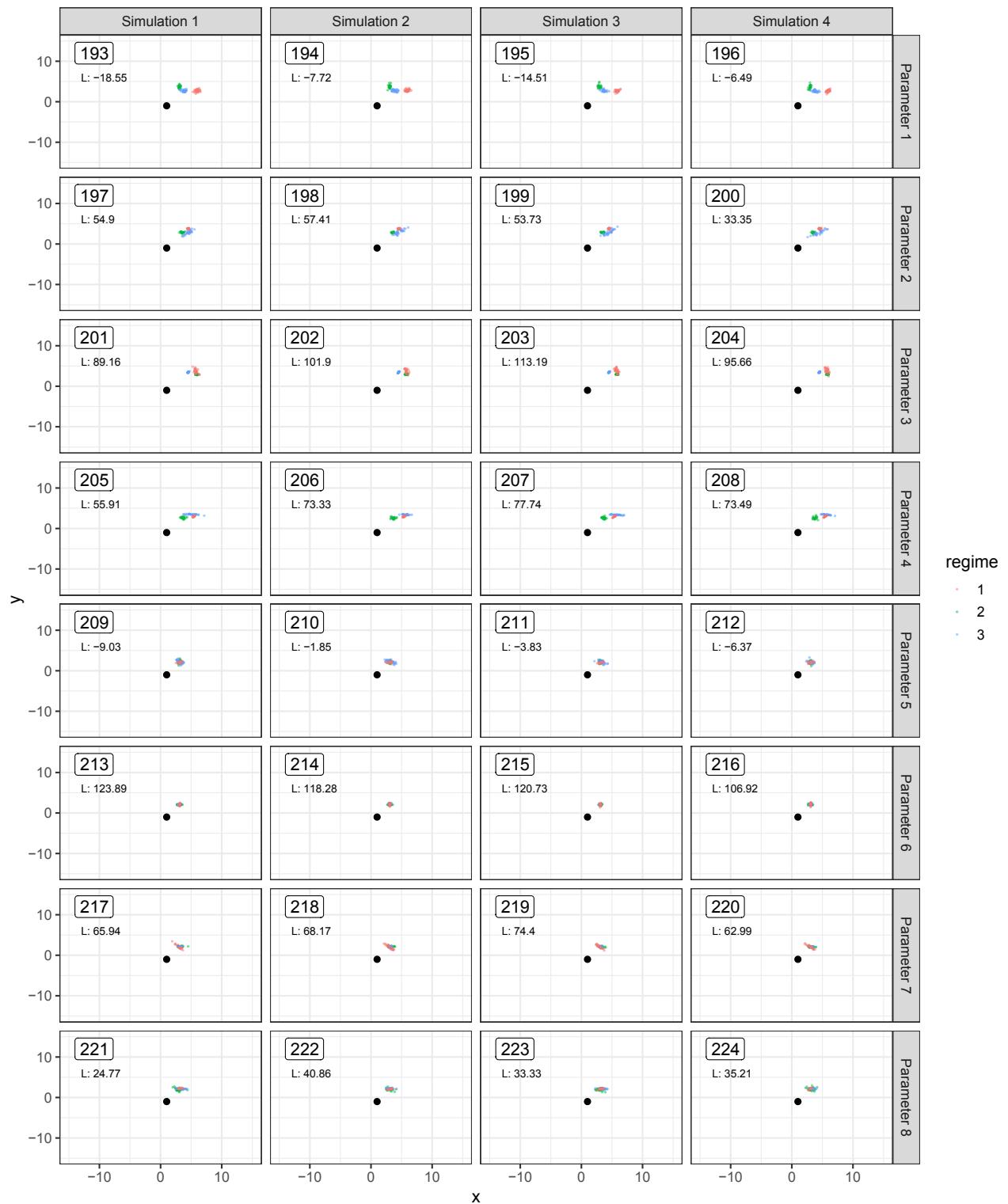
Tree-type ultrametric / N=80 / 3 regimes / Mapping 1. DAB



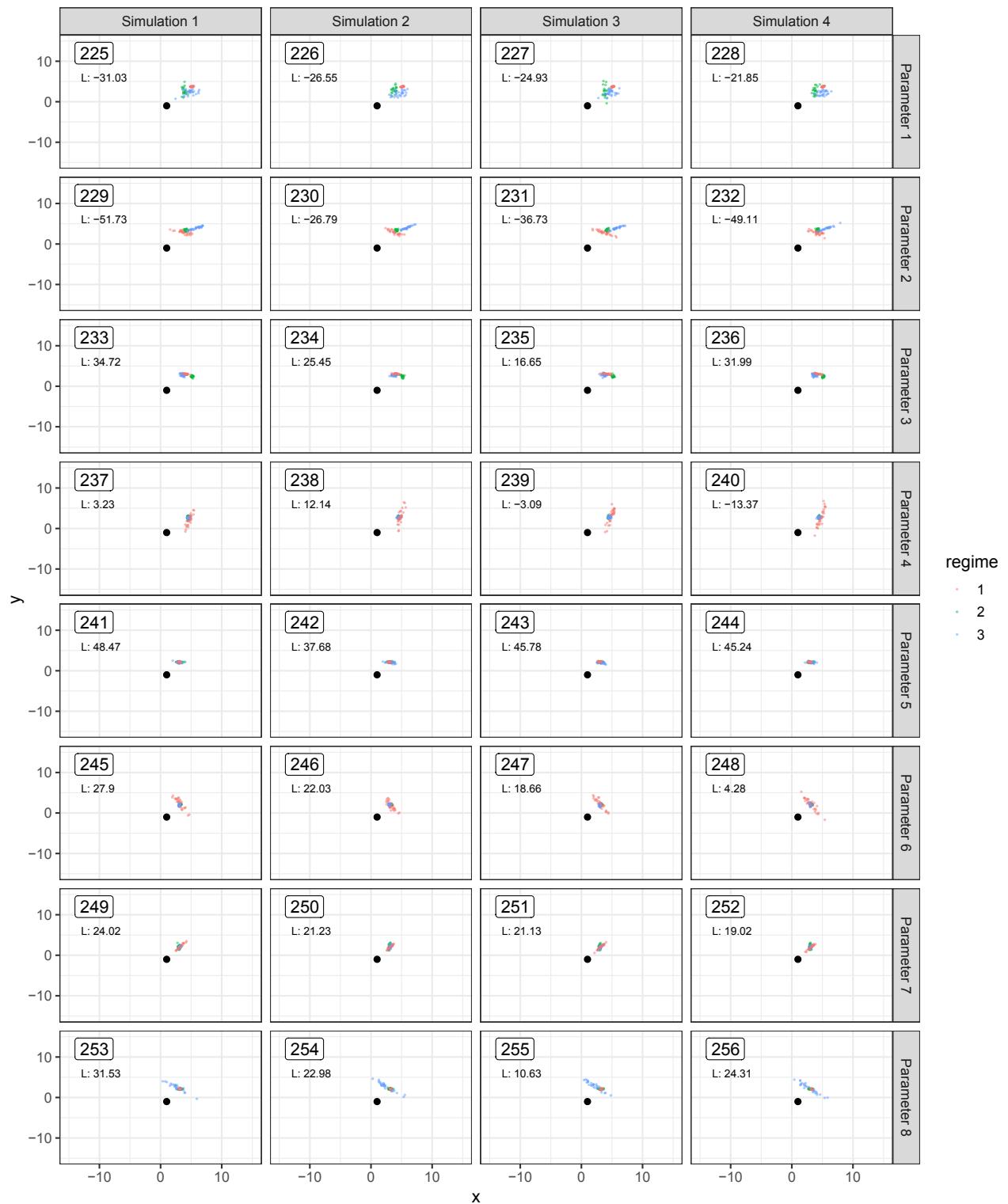
Tree-type ultrametric / N=80 / 3 regimes / Mapping 2. BEC



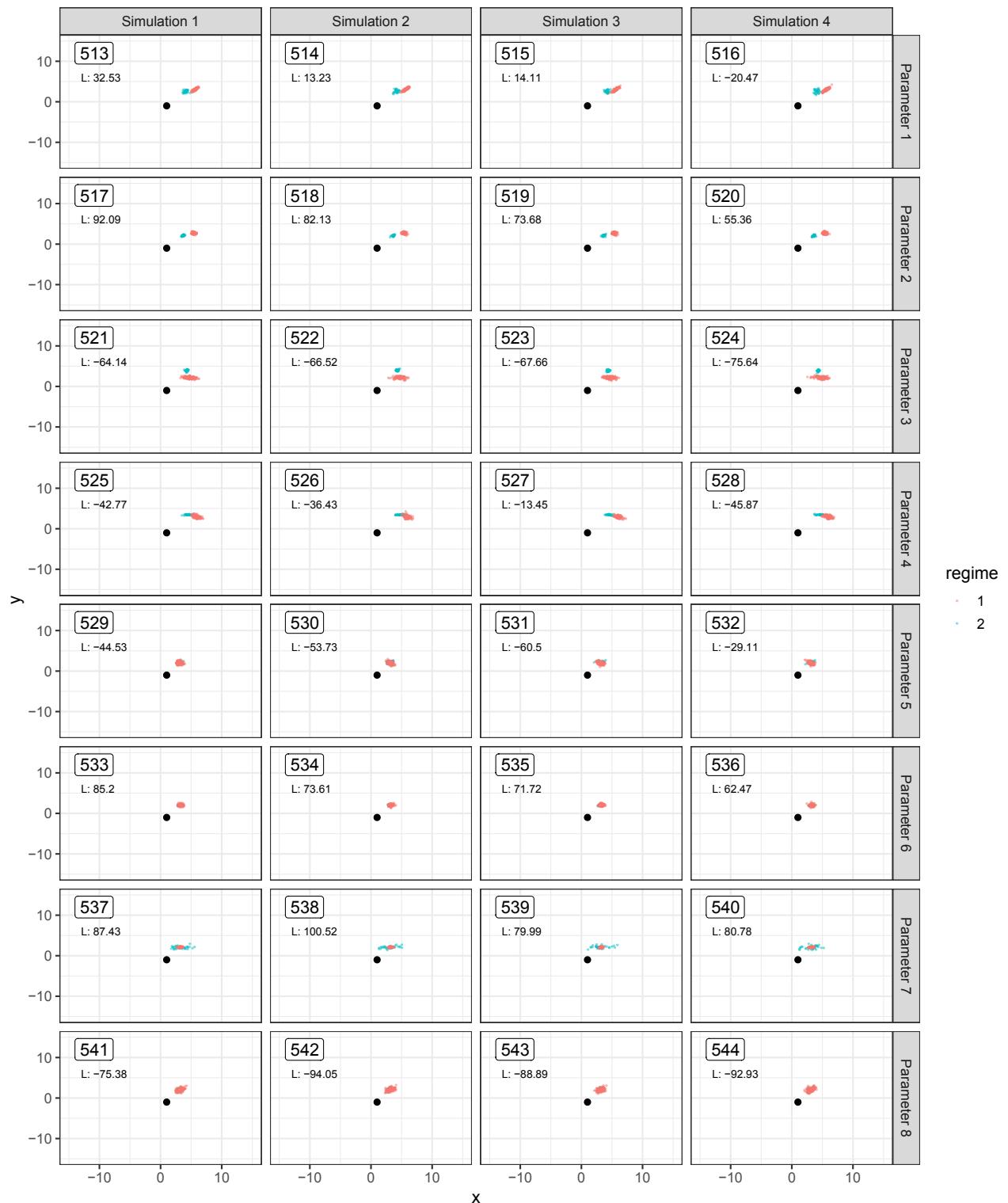
Tree-type ultrametric / N=80 / 3 regimes / Mapping 3. ECE



Tree-type ultrametric / N=80 / 3 regimes / Mapping 4. FCE



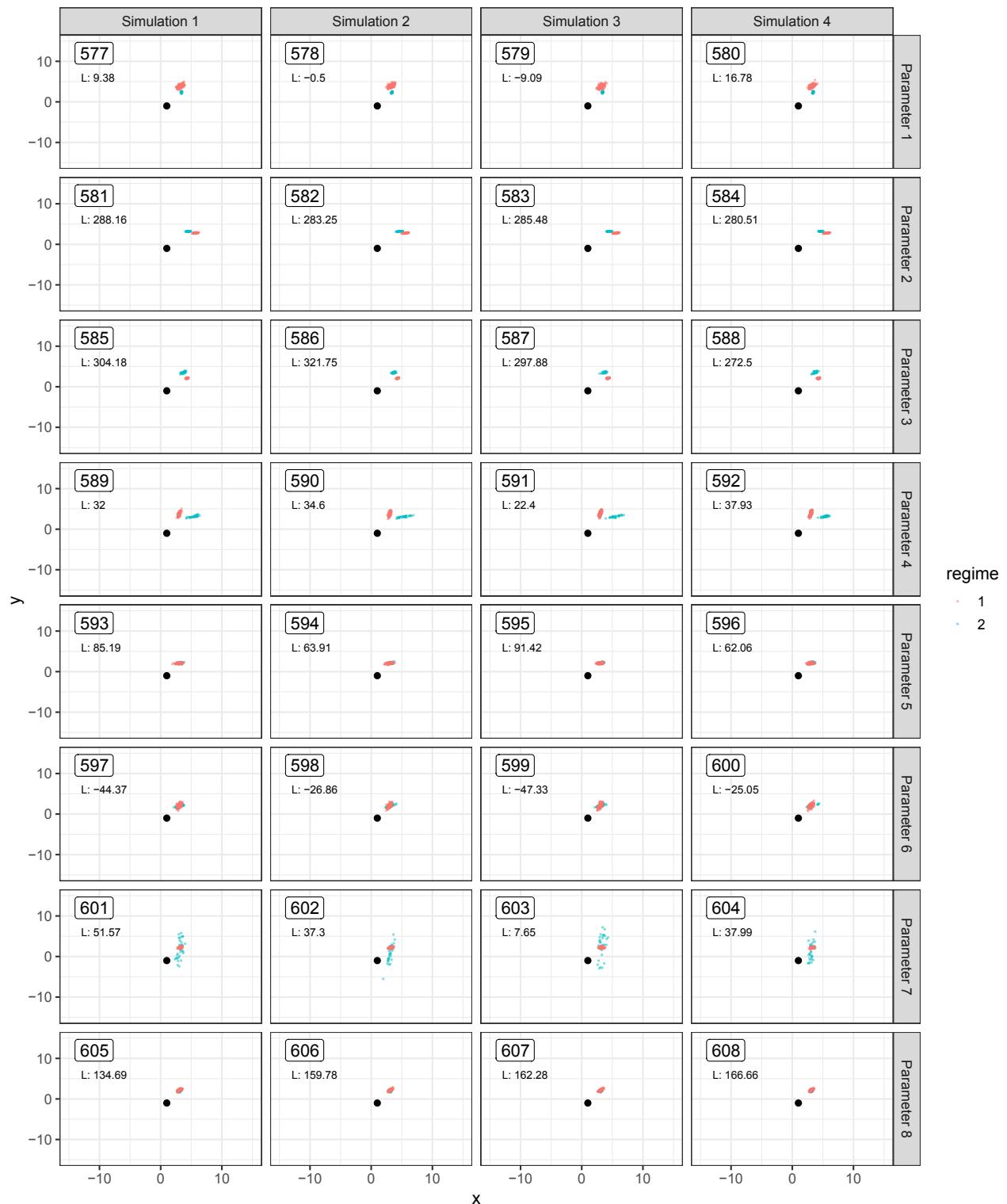
Tree-type ultrametric / N=159 / 2 regimes / Mapping 1. ED



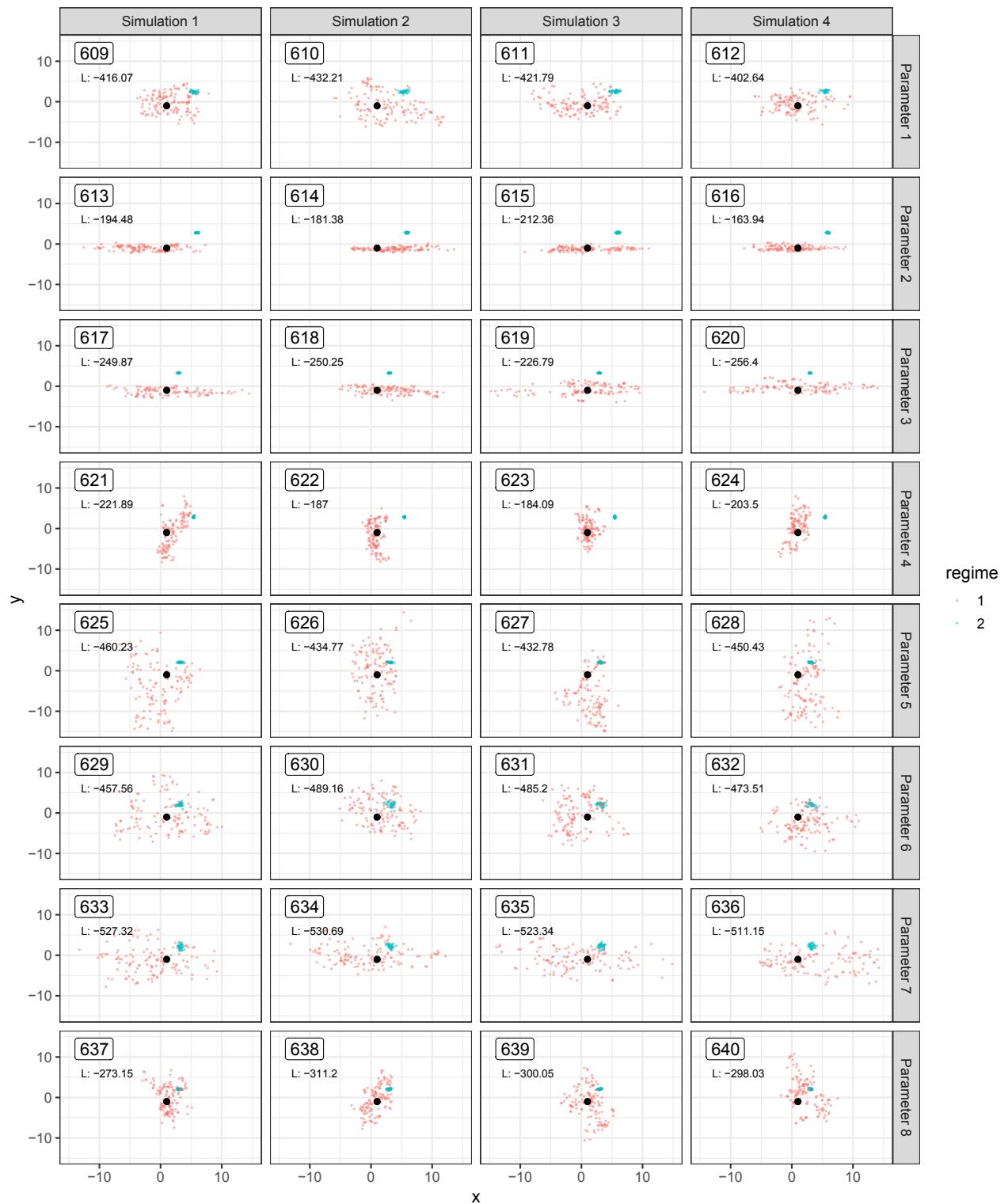
Tree-type ultrametric / N=159 / 2 regimes / Mapping 2. ED



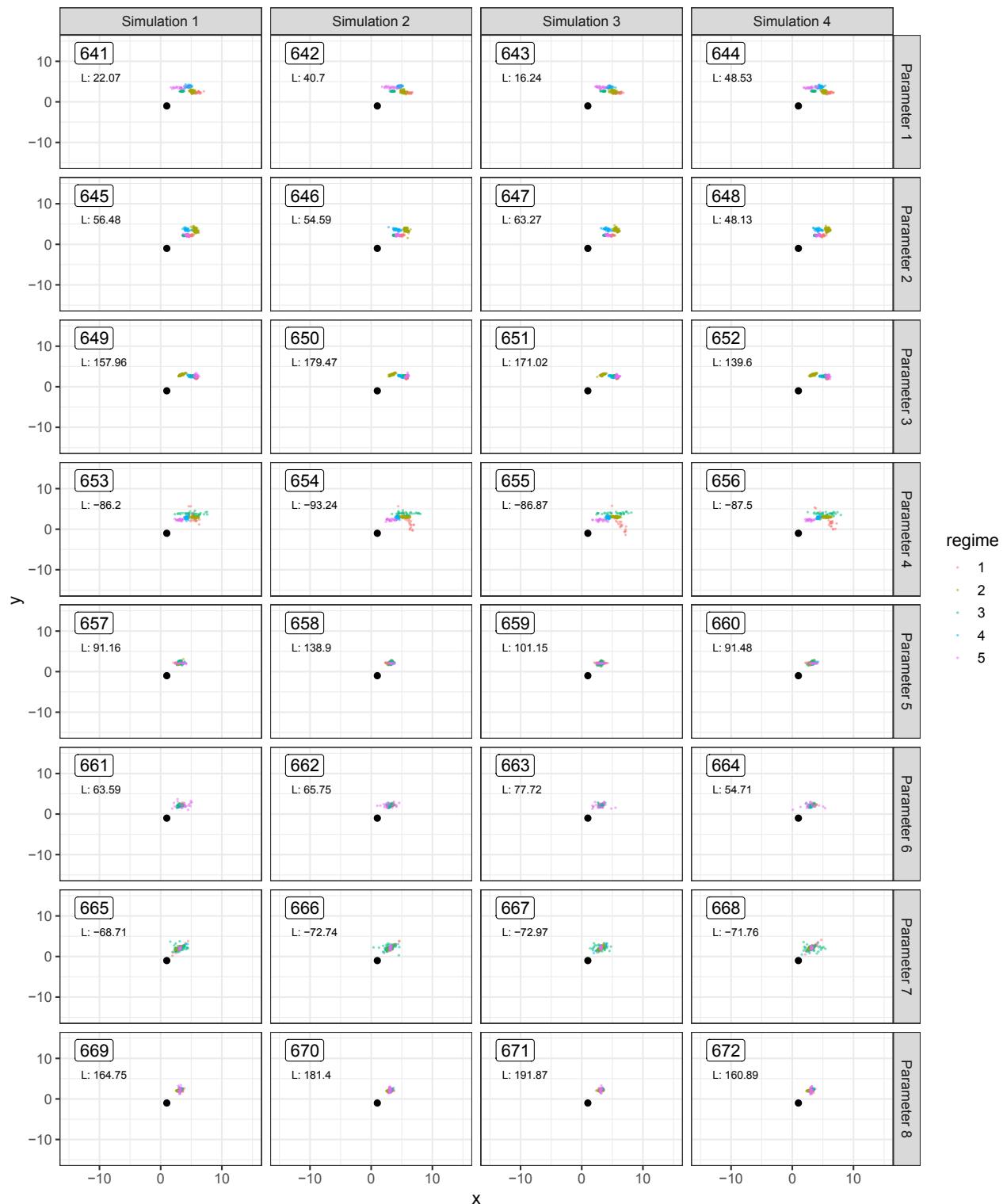
Tree-type ultrametric / N=159 / 2 regimes / Mapping 3. DE



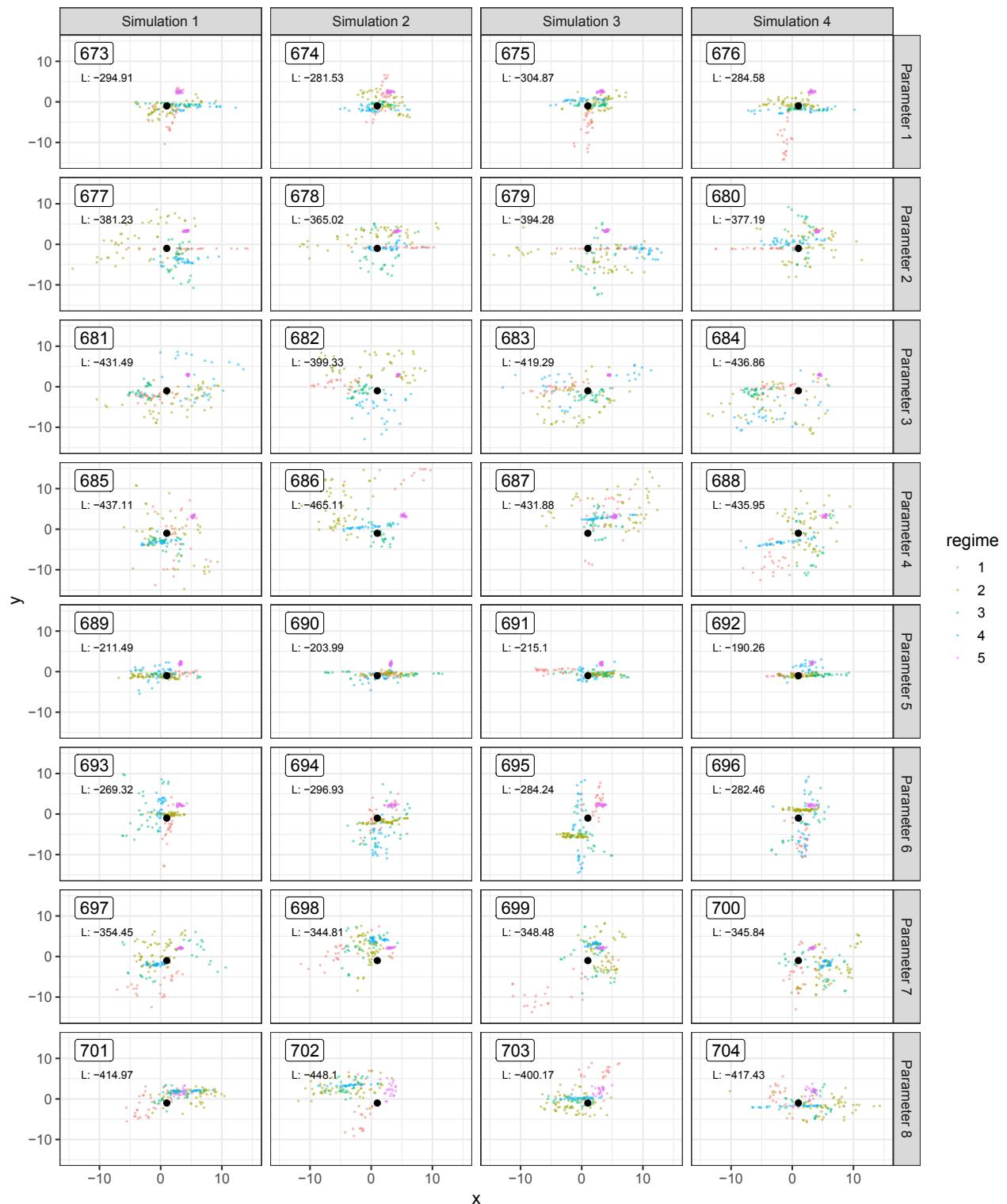
Tree-type ultrametric / N=159 / 2 regimes / Mapping 4. AC



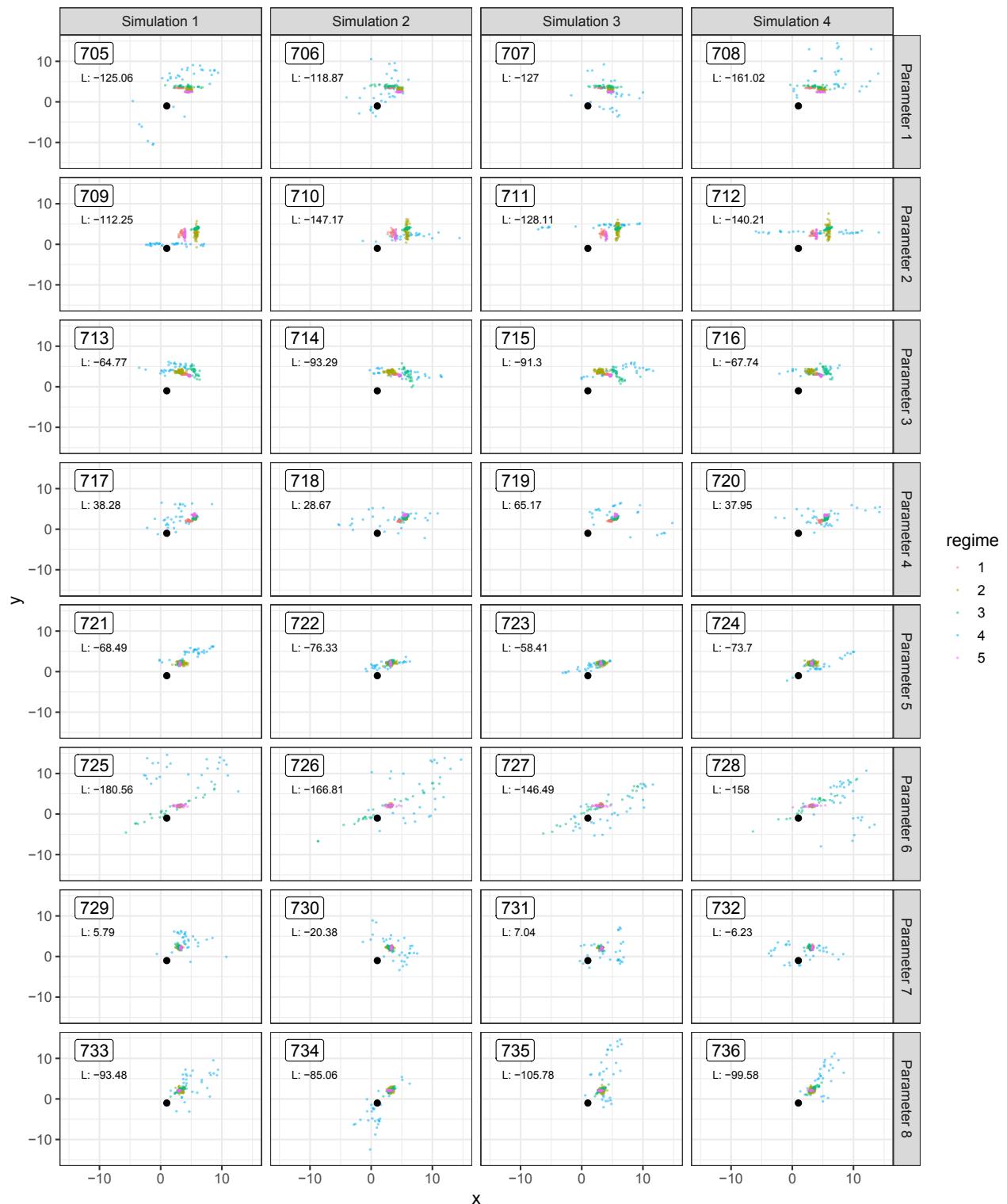
Tree-type ultrametric / N=159 / 5 regimes / Mapping 1. EECFC



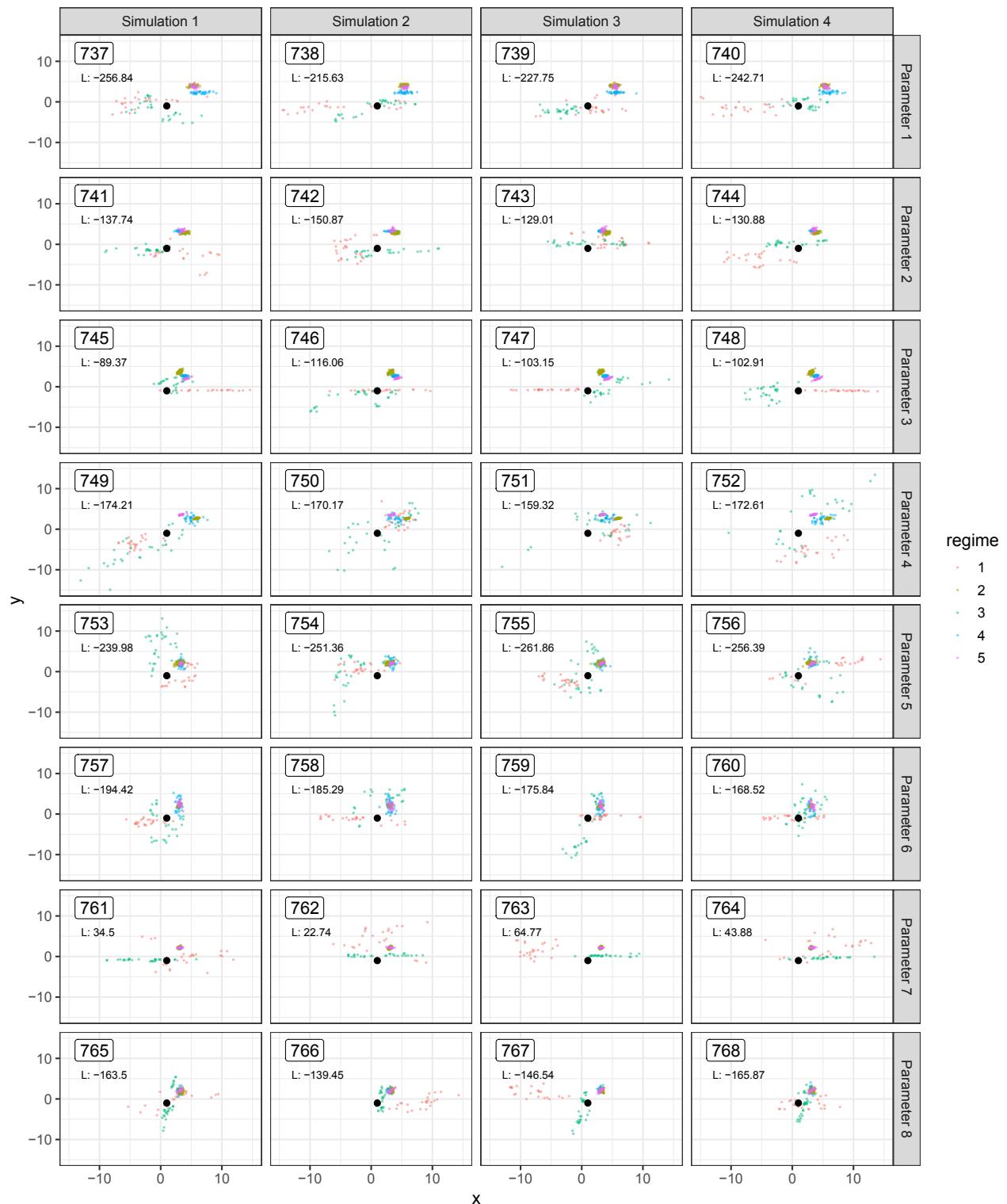
Tree-type ultrametric / N=159 / 5 regimes / Mapping 2. BAABD



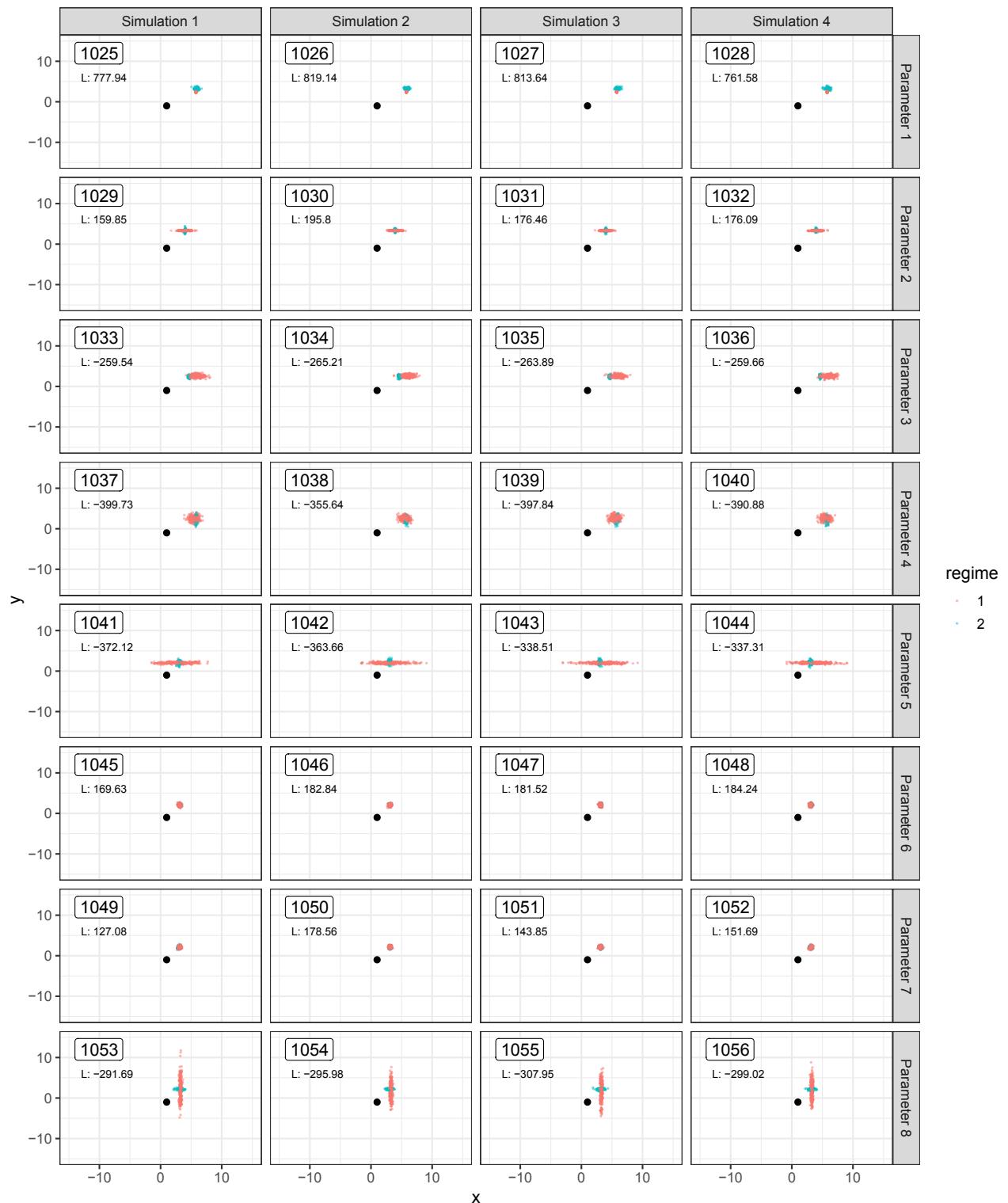
Tree-type ultrametric / N=159 / 5 regimes / Mapping 3. DCFBC



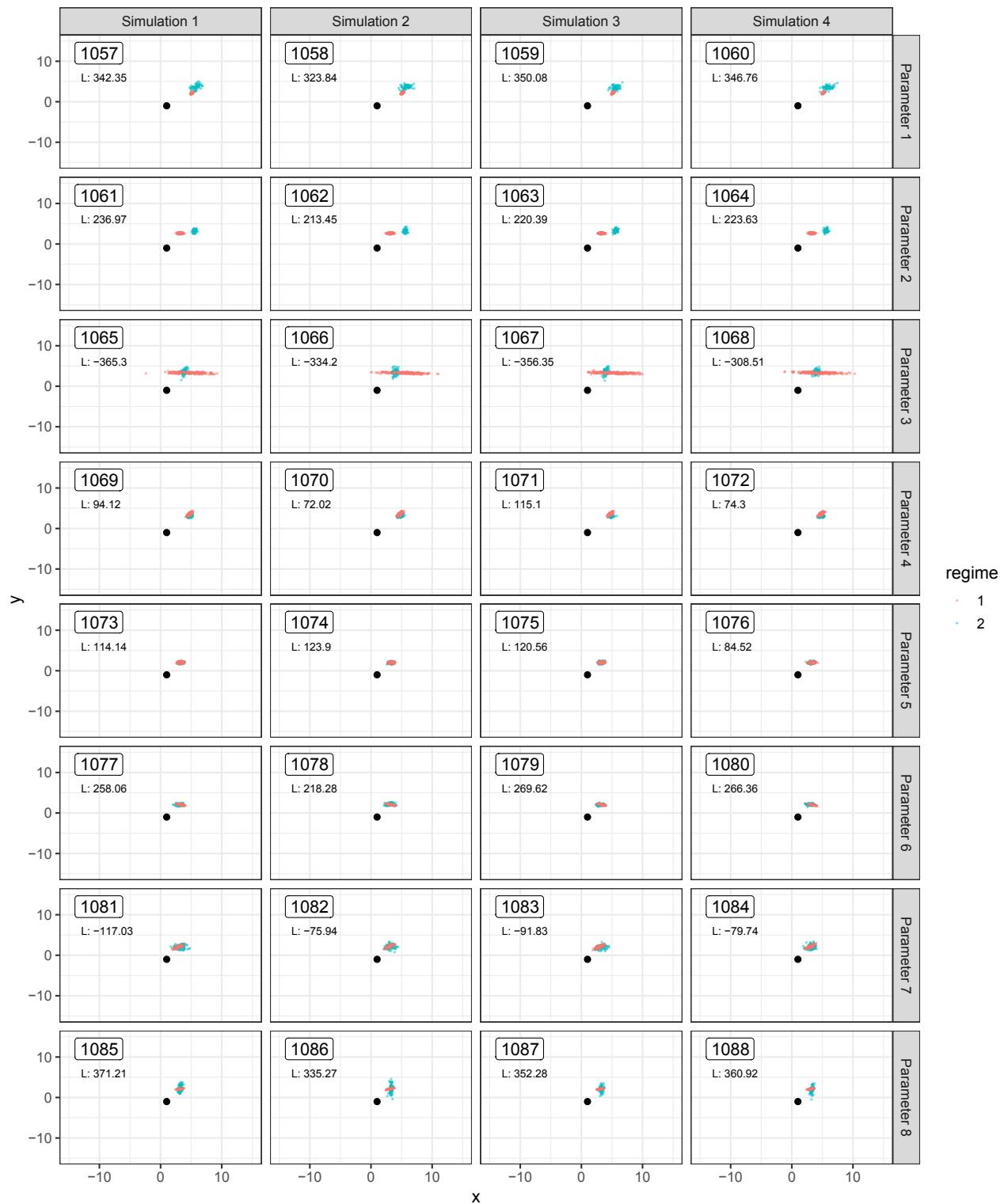
Tree-type ultrametric / N=159 / 5 regimes / Mapping 4. BDBCE



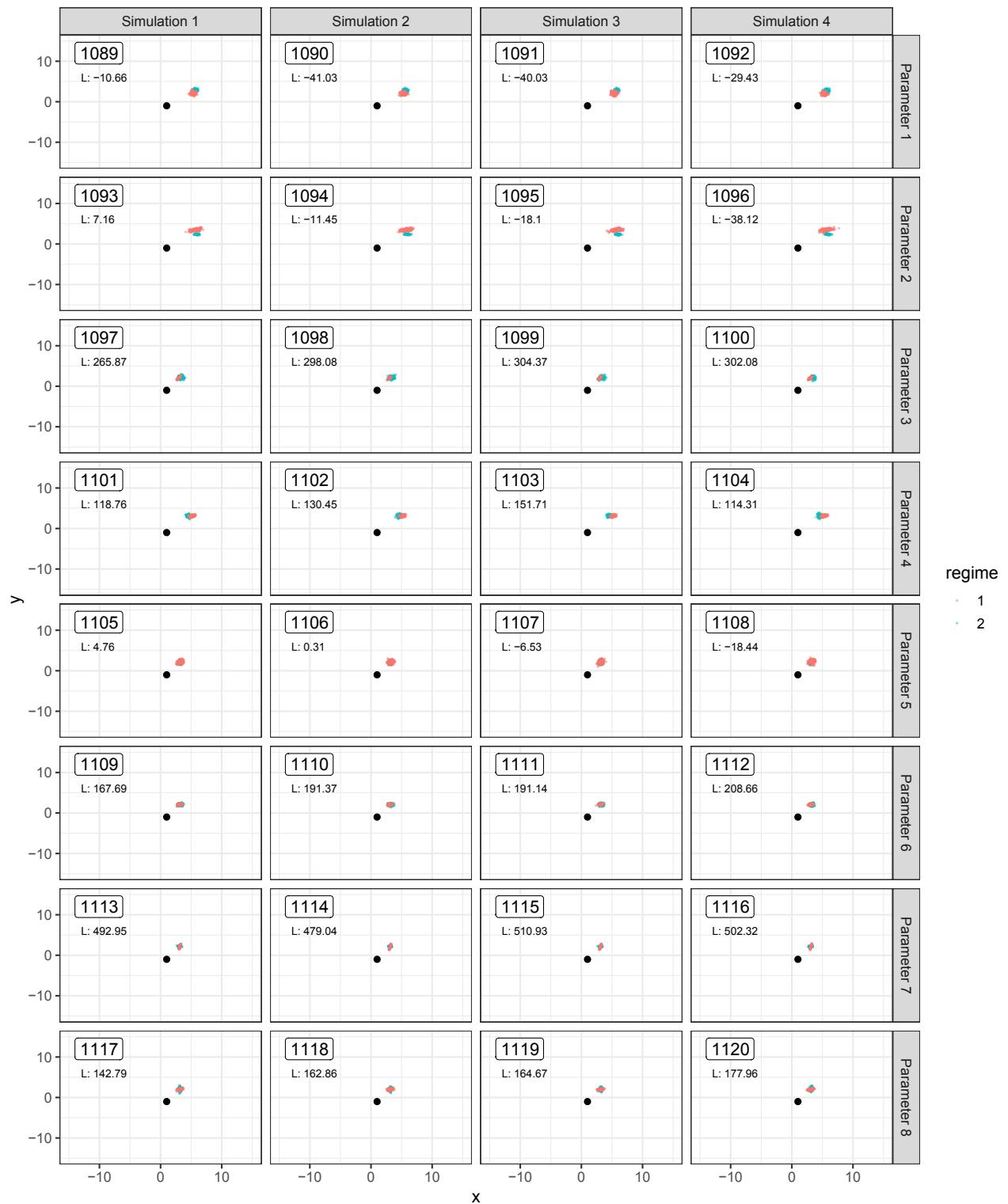
Tree-type ultrametric / N=318 / 2 regimes / Mapping 1. CC



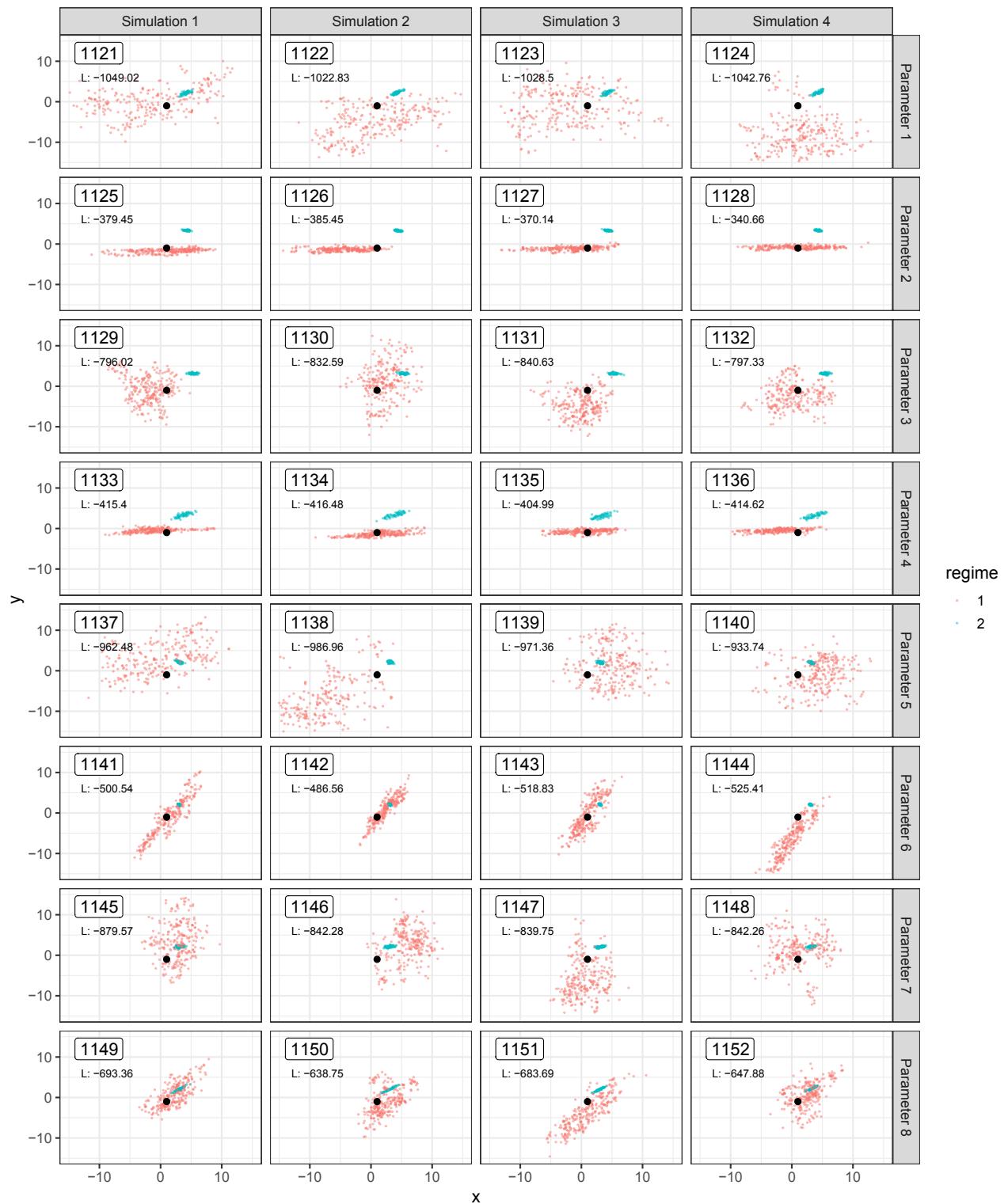
Tree-type ultrametric / N=318 / 2 regimes / Mapping 2. ED



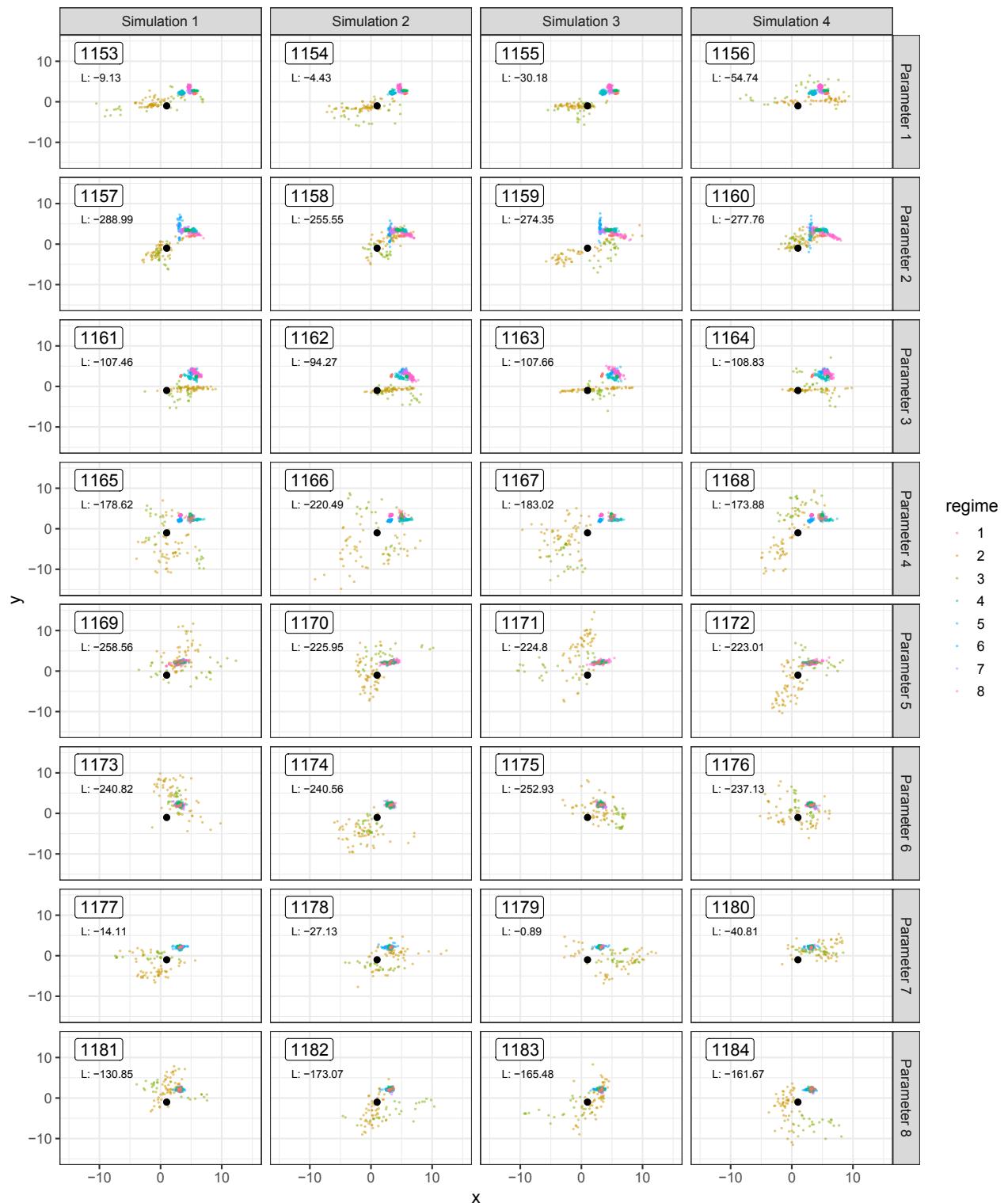
Tree-type ultrametric / N=318 / 2 regimes / Mapping 3. DC



Tree-type ultrametric / N=318 / 2 regimes / Mapping 4. BF



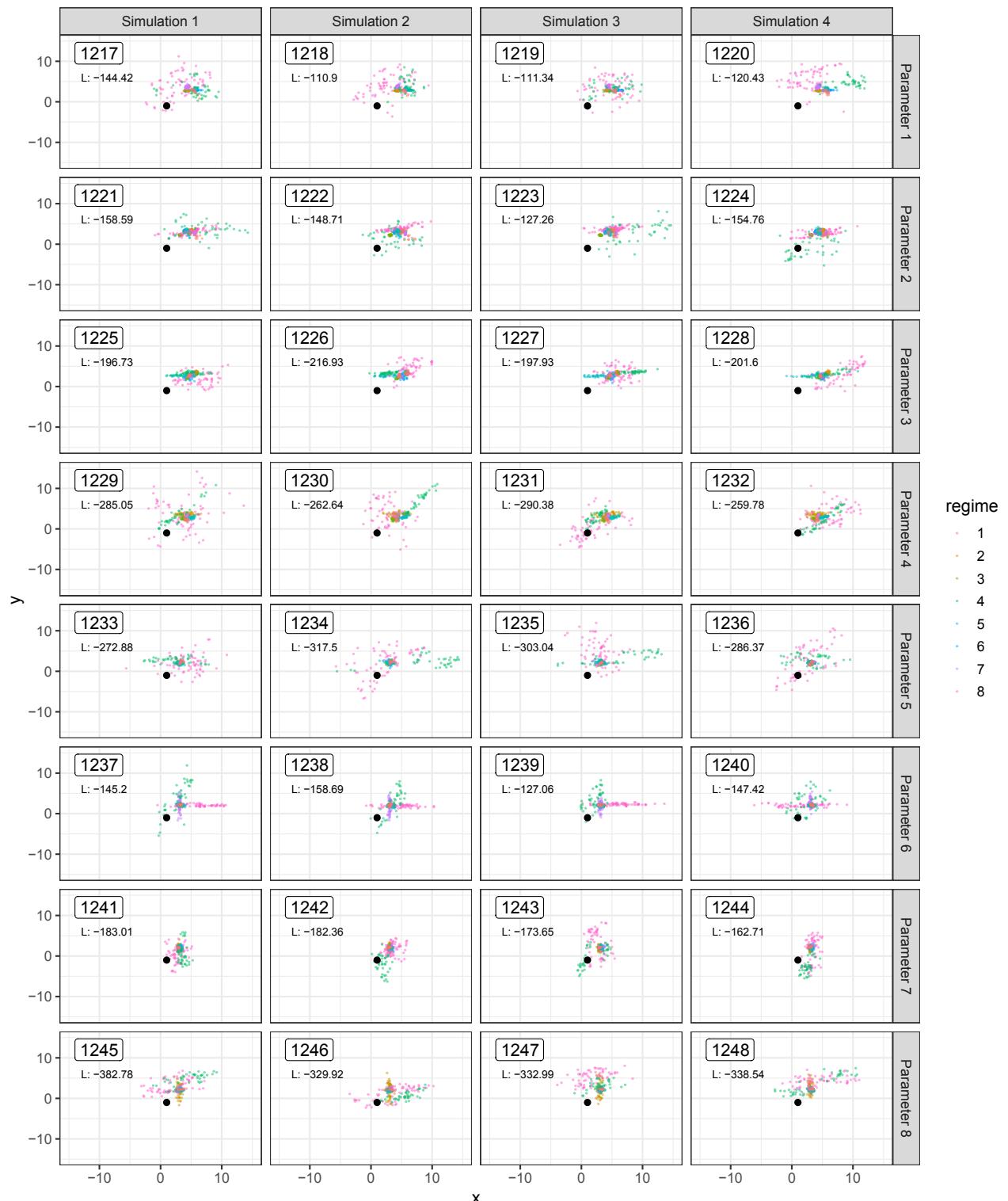
Tree-type ultrametric / N=318 / 8 regimes / Mapping 1. DBACFDFE



Tree-type ultrametric / N=318 / 8 regimes / Mapping 2. CCAAEACD



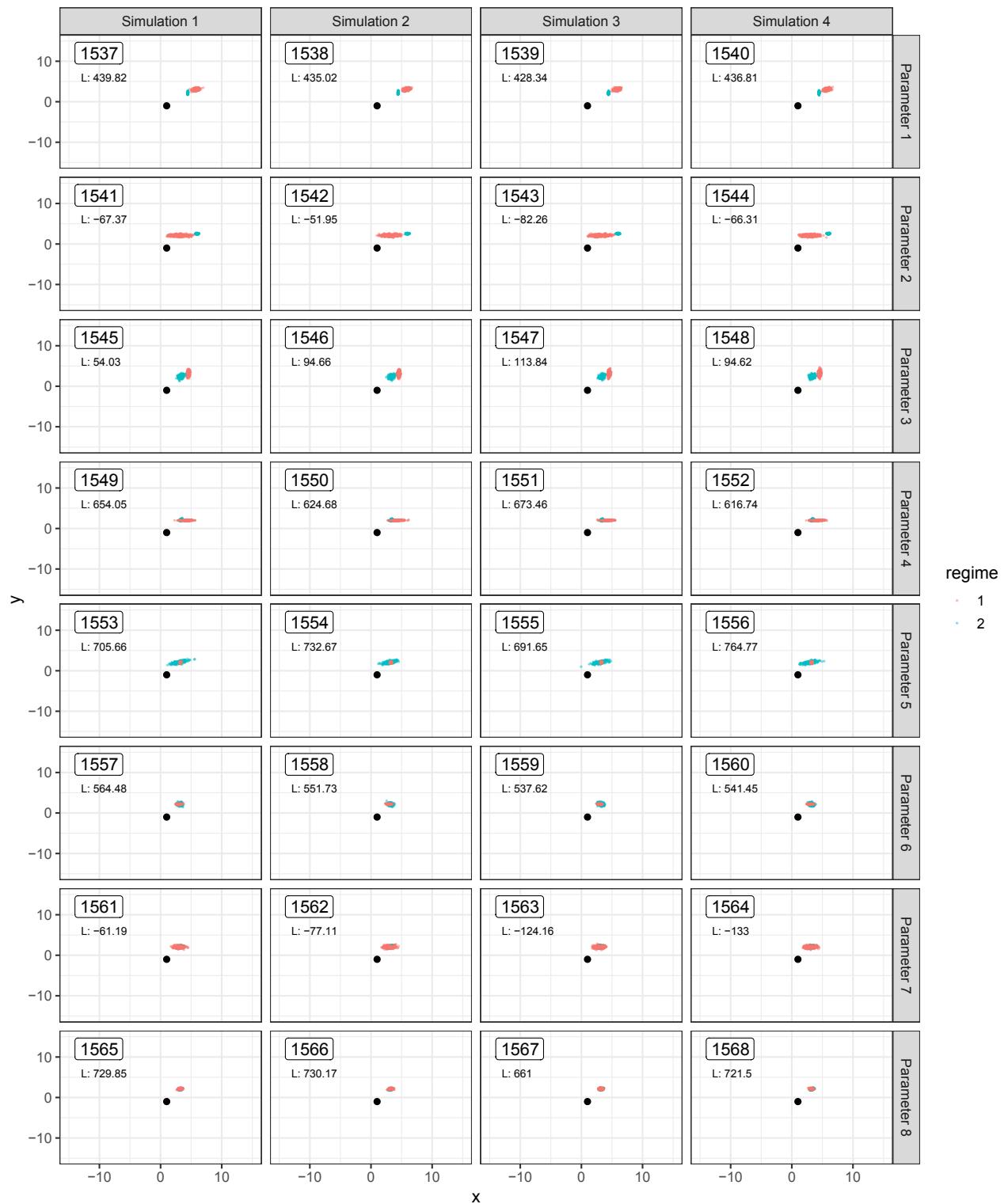
Tree-type ultrametric / N=318 / 8 regimes / Mapping 3. FCCBECDB



Tree-type ultrametric / N=318 / 8 regimes / Mapping 4. BDDAADFD



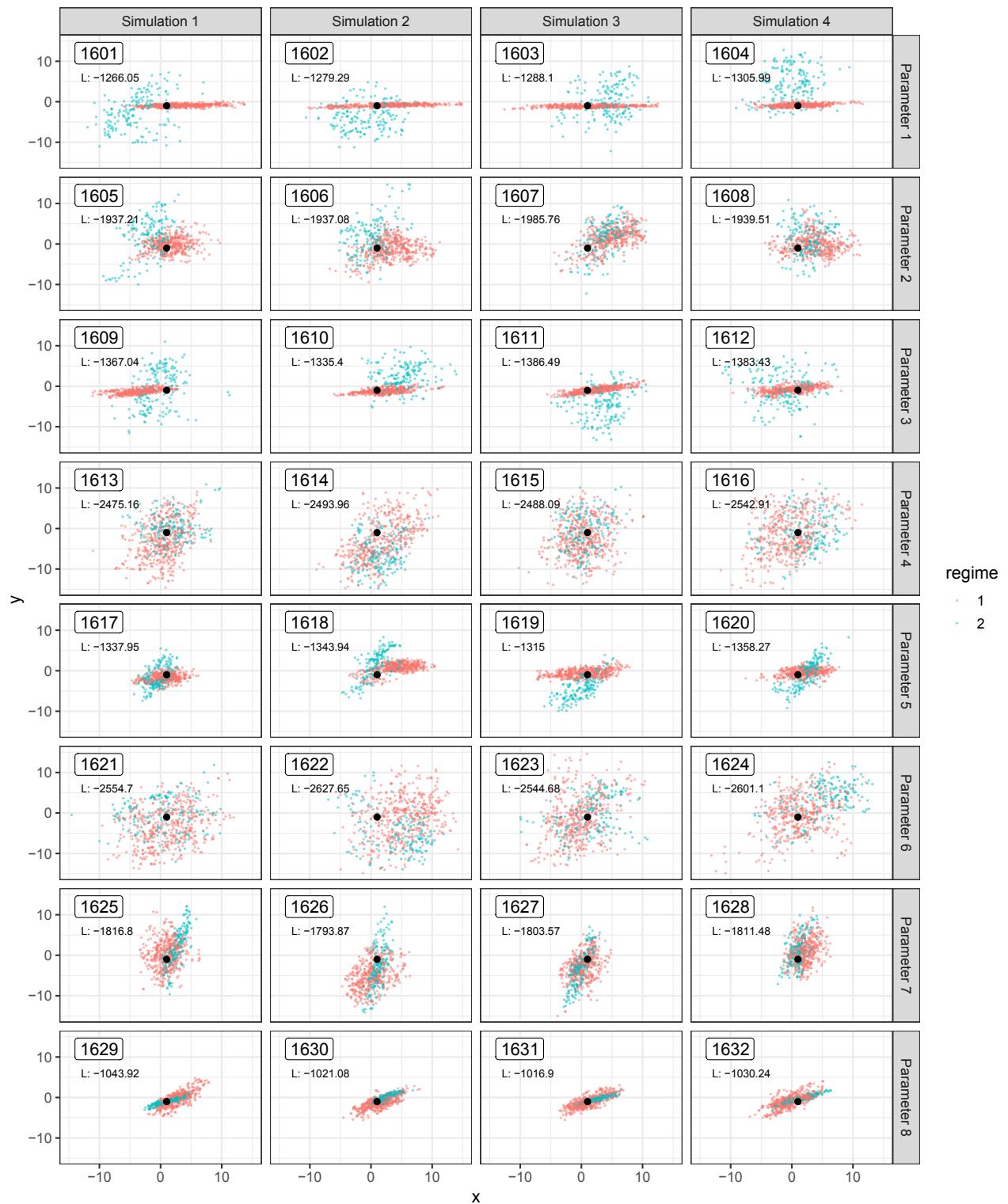
Tree-type ultrametric / N=638 / 2 regimes / Mapping 1. DE



Tree-type ultrametric / N=638 / 2 regimes / Mapping 2. DF



Tree-type ultrametric / N=638 / 2 regimes / Mapping 3. BB



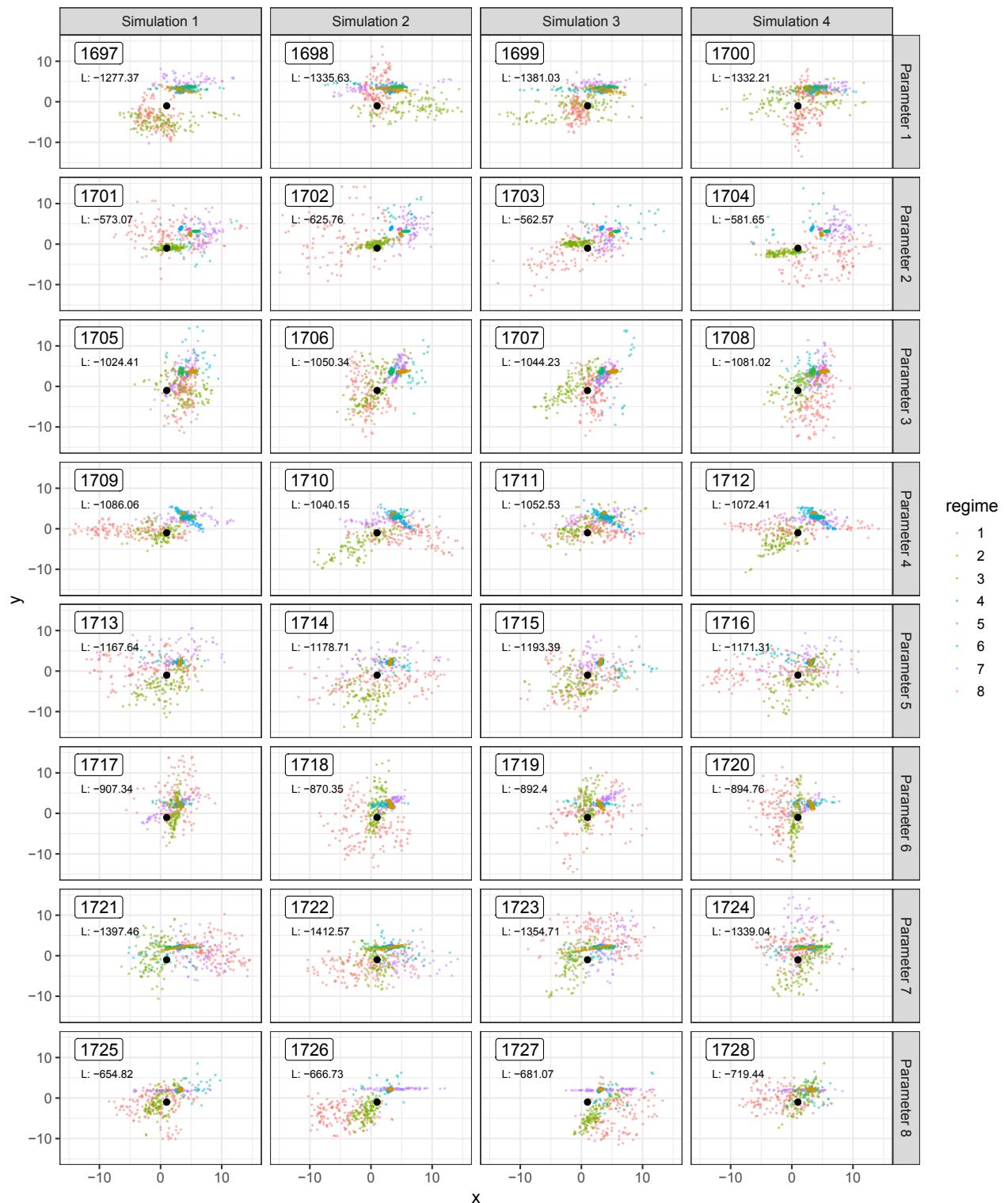
Tree-type ultrametric / N=638 / 2 regimes / Mapping 4. FD



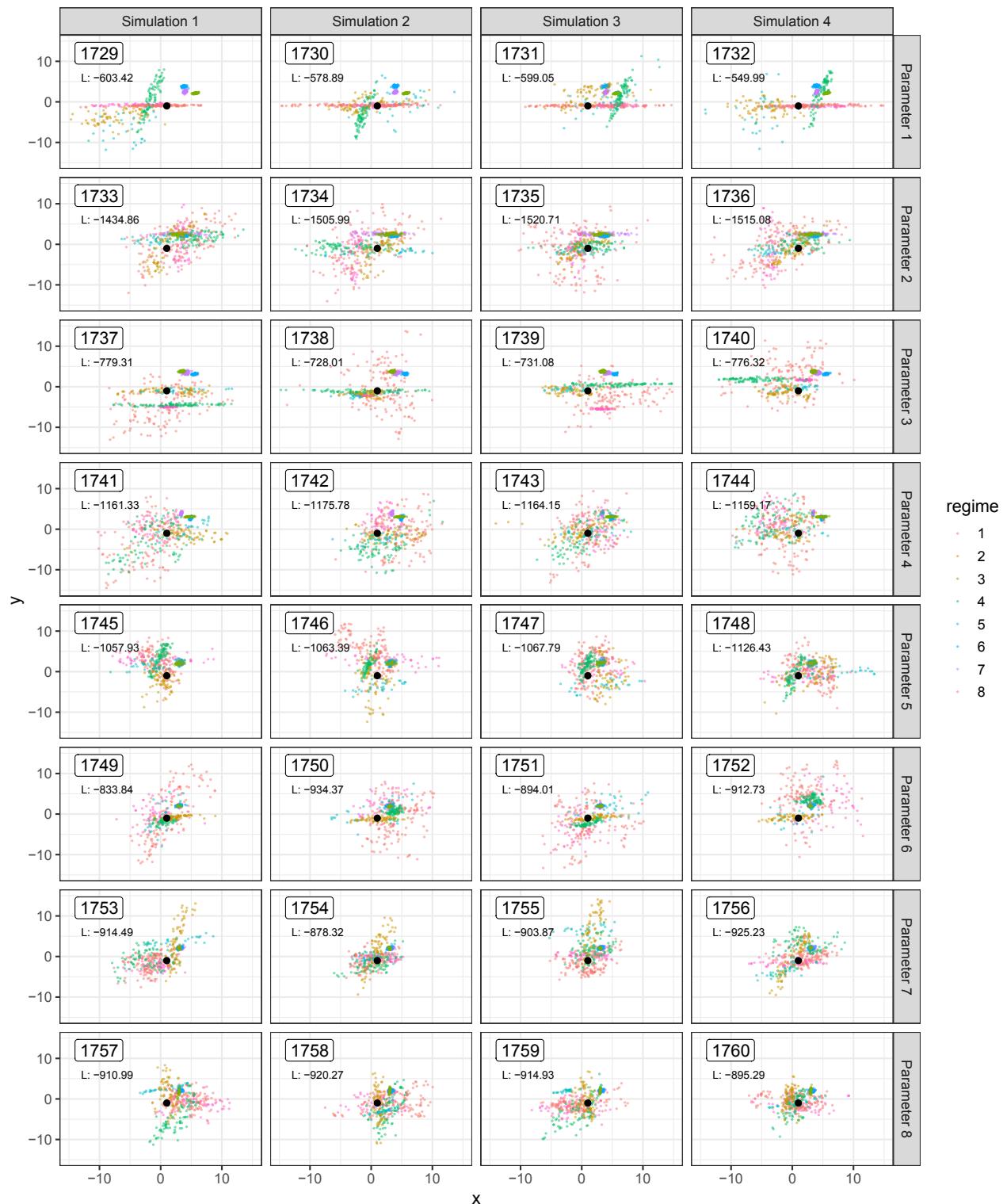
Tree-type ultrametric / N=638 / 8 regimes / Mapping 1. FBEEFDEC



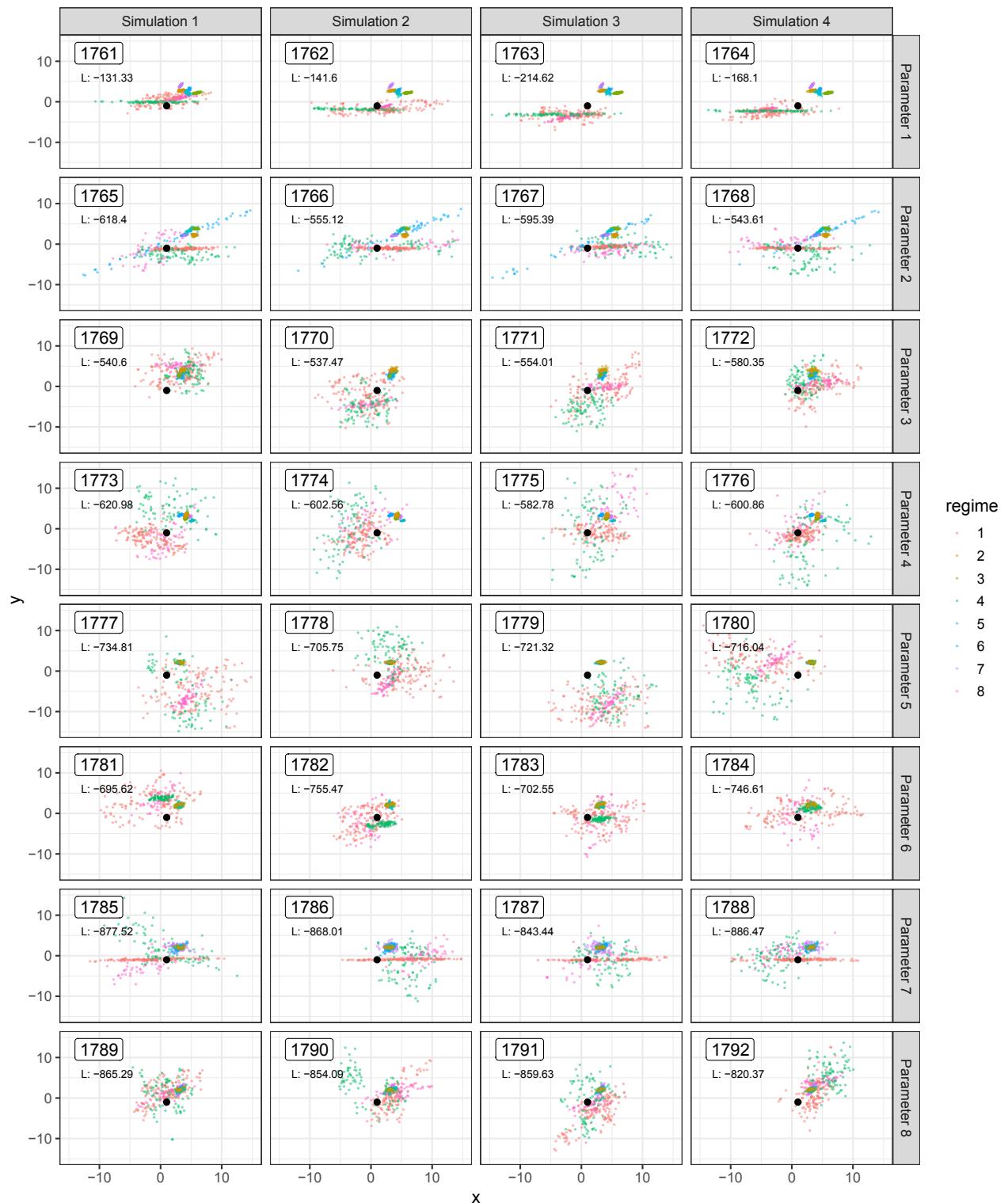
Tree-type ultrametric / N=638 / 8 regimes / Mapping 2. AFBDAFBE



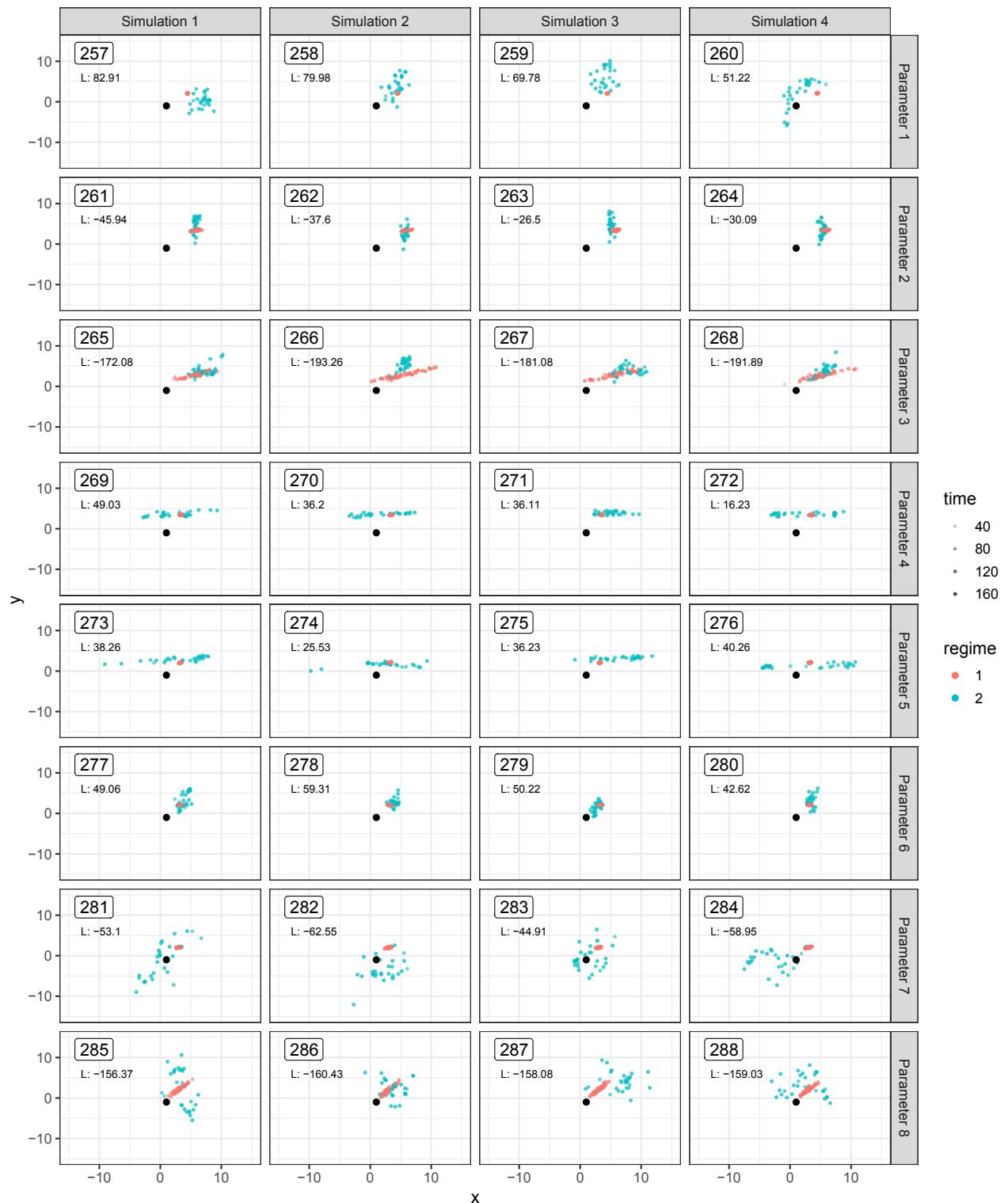
Tree-type ultrametric / N=638 / 8 regimes / Mapping 3. BBDBBDDA



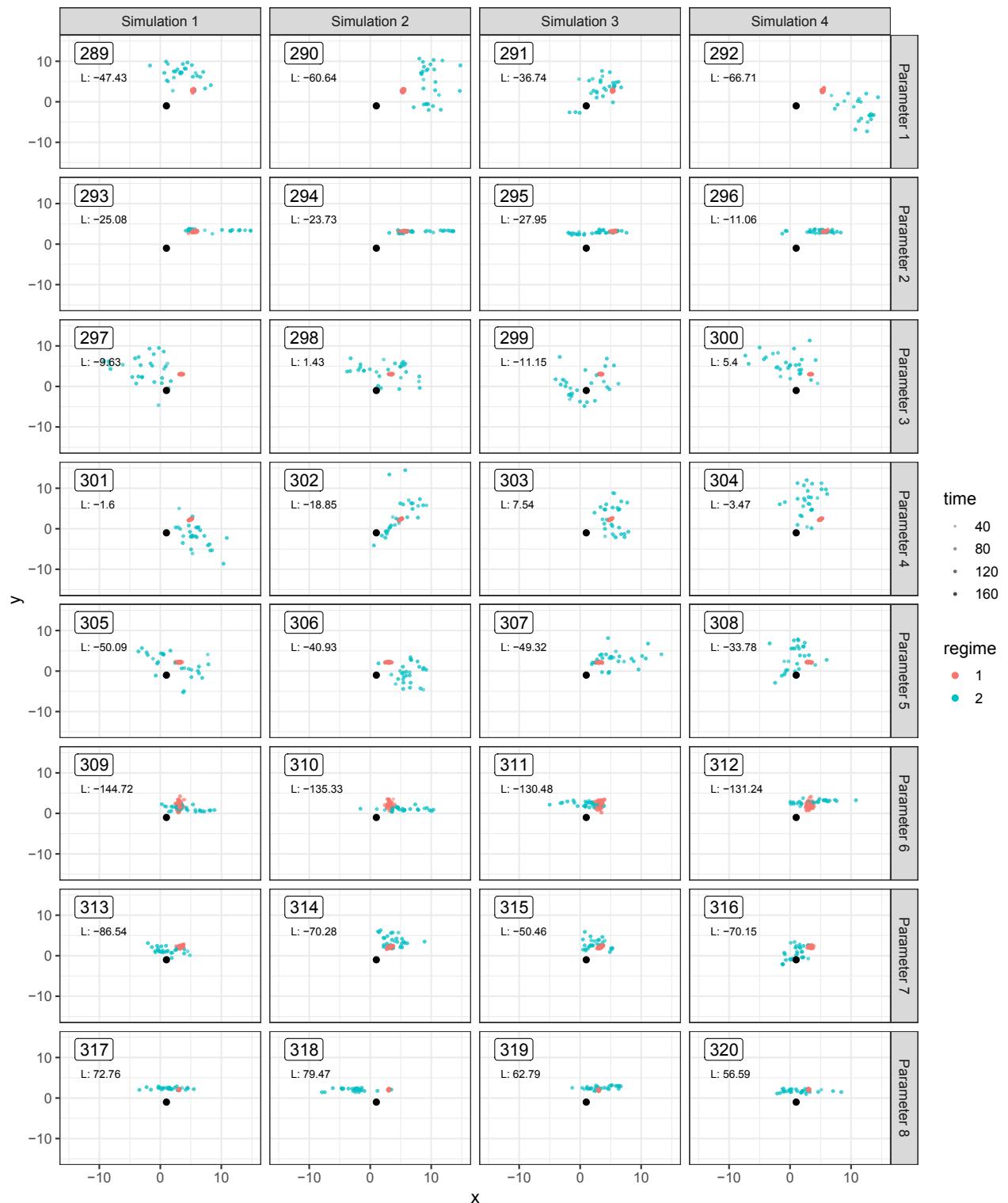
Tree-type ultrametric / N=638 / 8 regimes / Mapping 4. BEFAEFEB



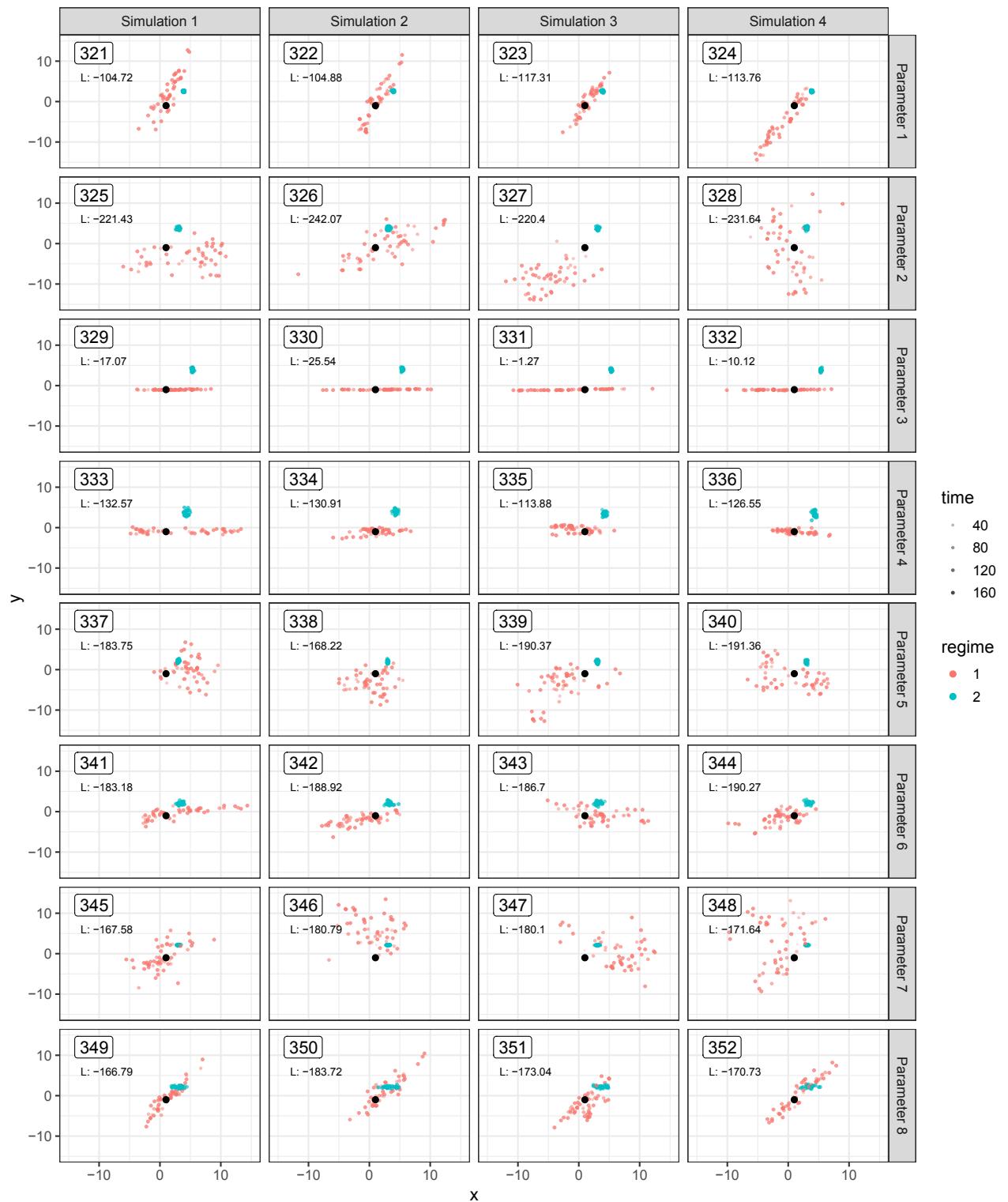
Tree-type non-ultrametric / N=80 / 2 regimes / Mapping 1. FB



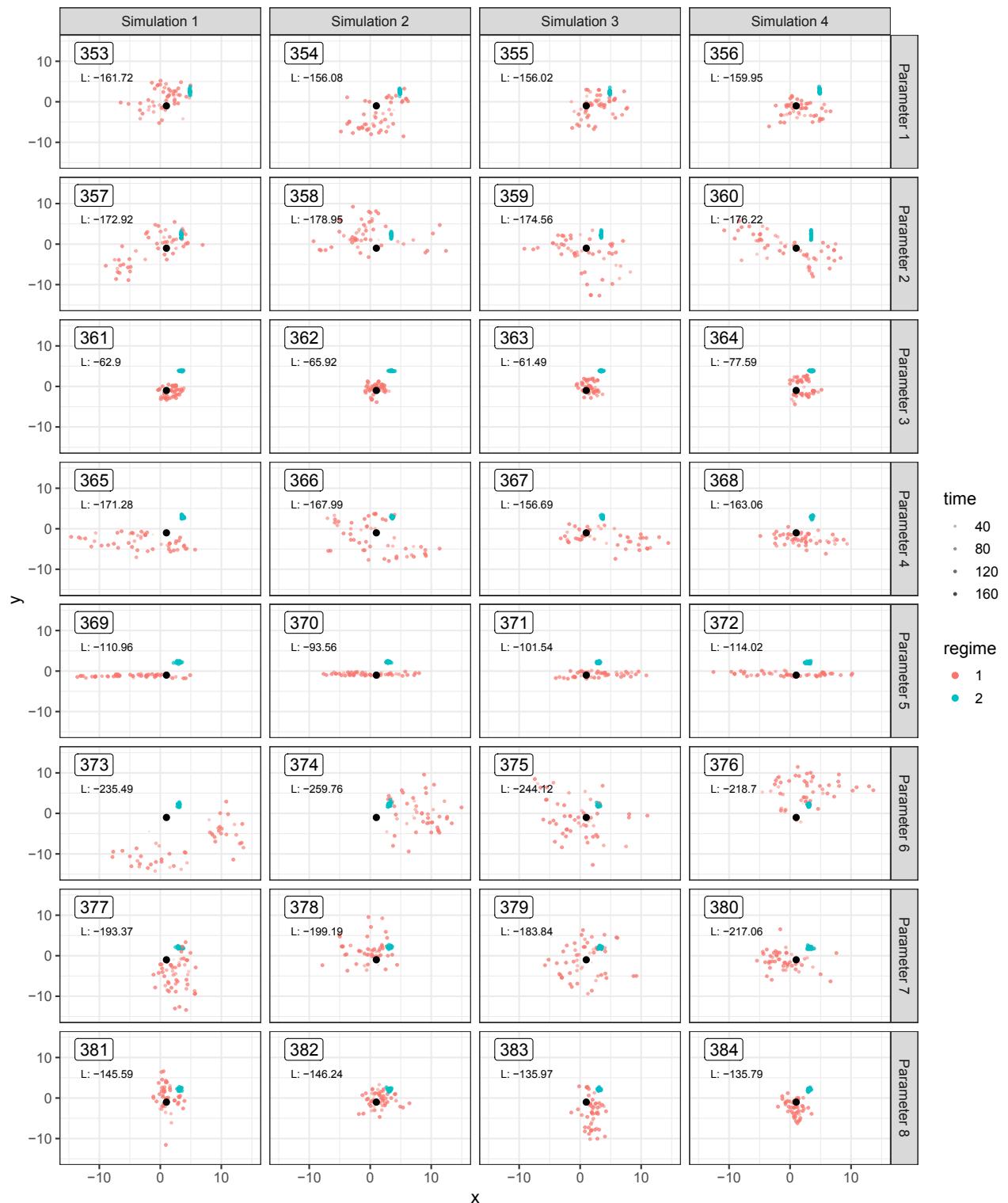
Tree-type non-ultrametric / N=80 / 2 regimes / Mapping 2. DA



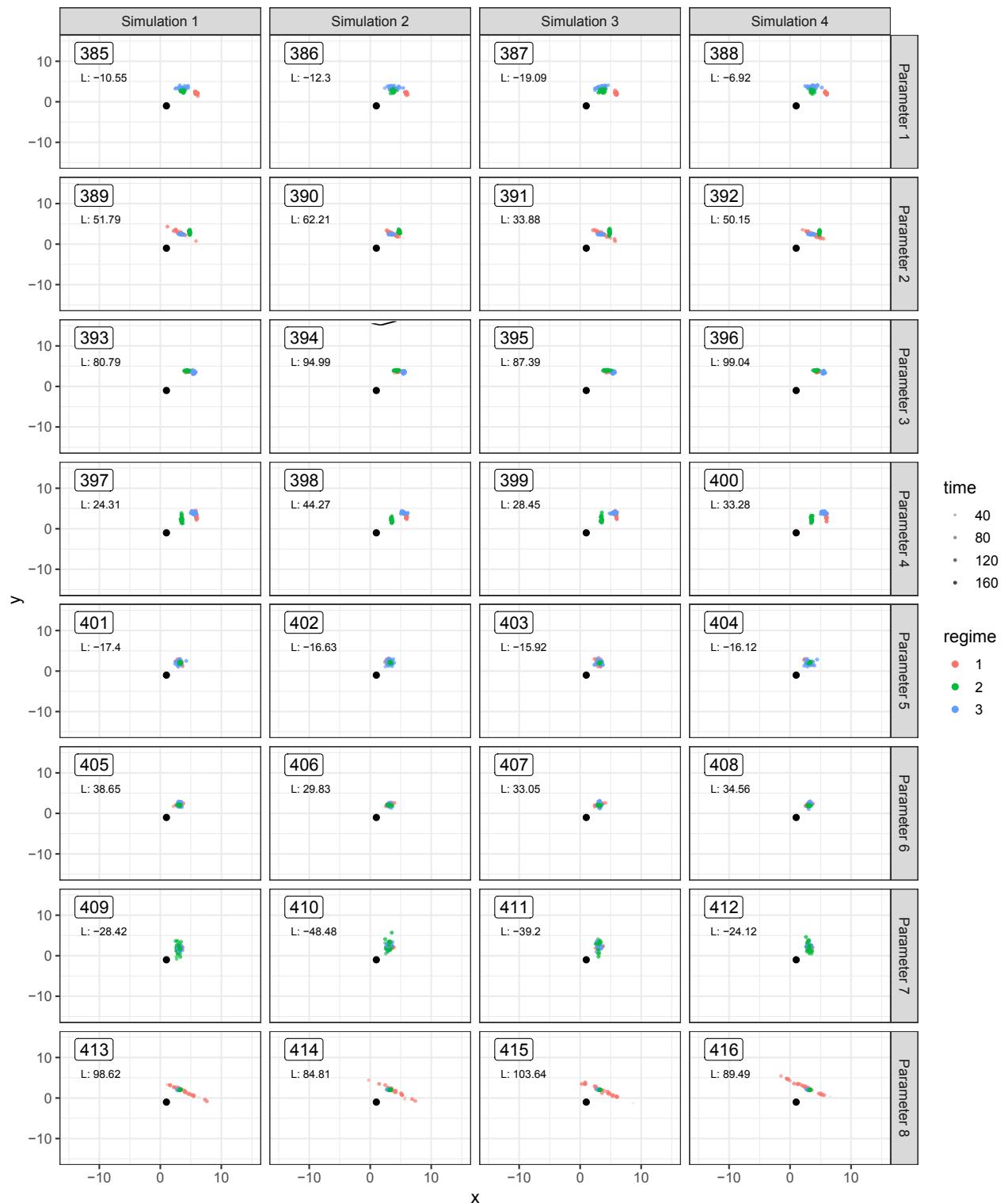
Tree-type non-ultrametric / N=80 / 2 regimes / Mapping 3. BC



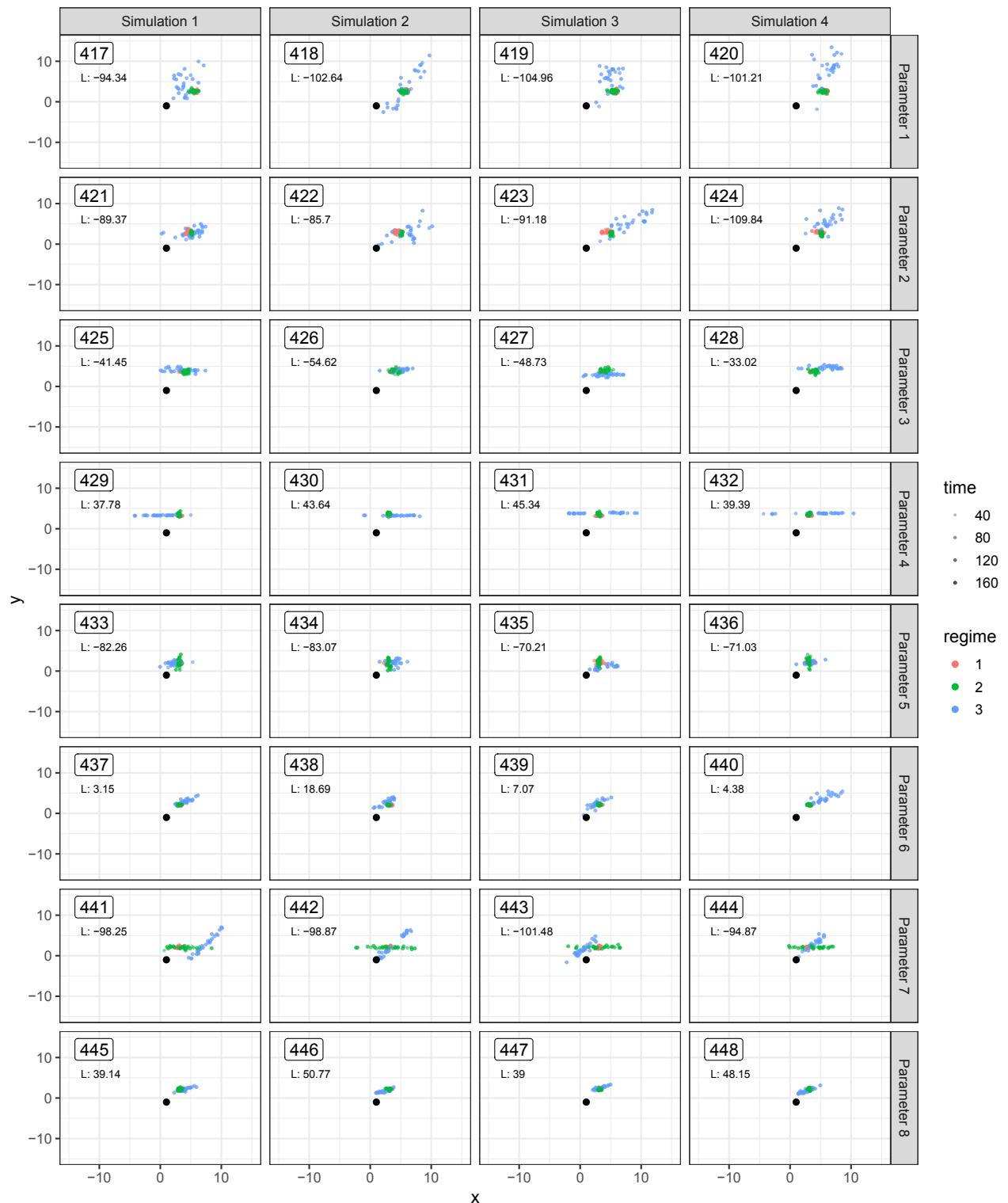
Tree-type non-ultrametric / N=80 / 2 regimes / Mapping 4. AC



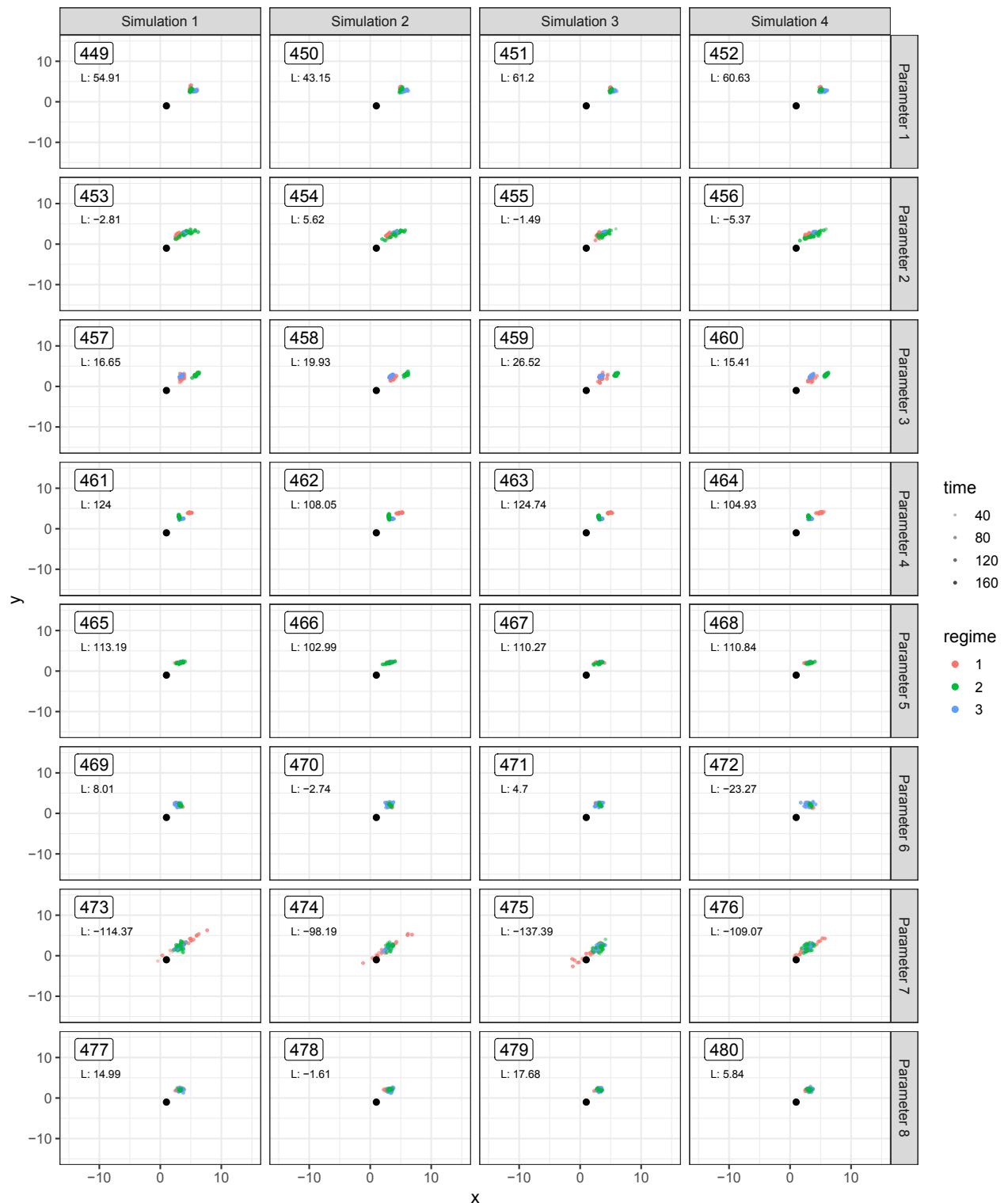
Tree-type non-ultrametric / N=80 / 3 regimes / Mapping 1. FCC



Tree-type non-ultrametric / N=80 / 3 regimes / Mapping 2. DCB



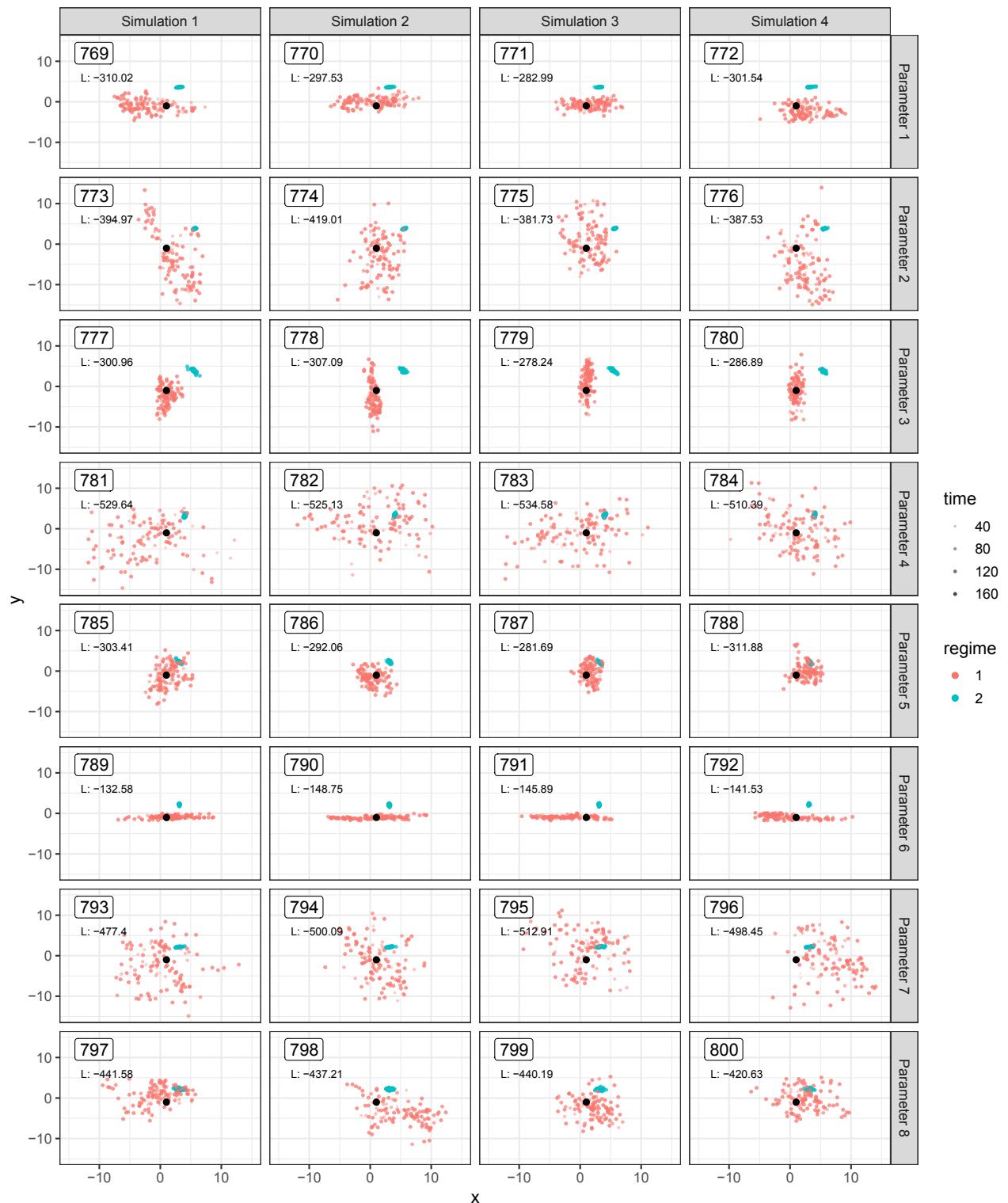
Tree-type non-ultrametric / N=80 / 3 regimes / Mapping 3. EEE



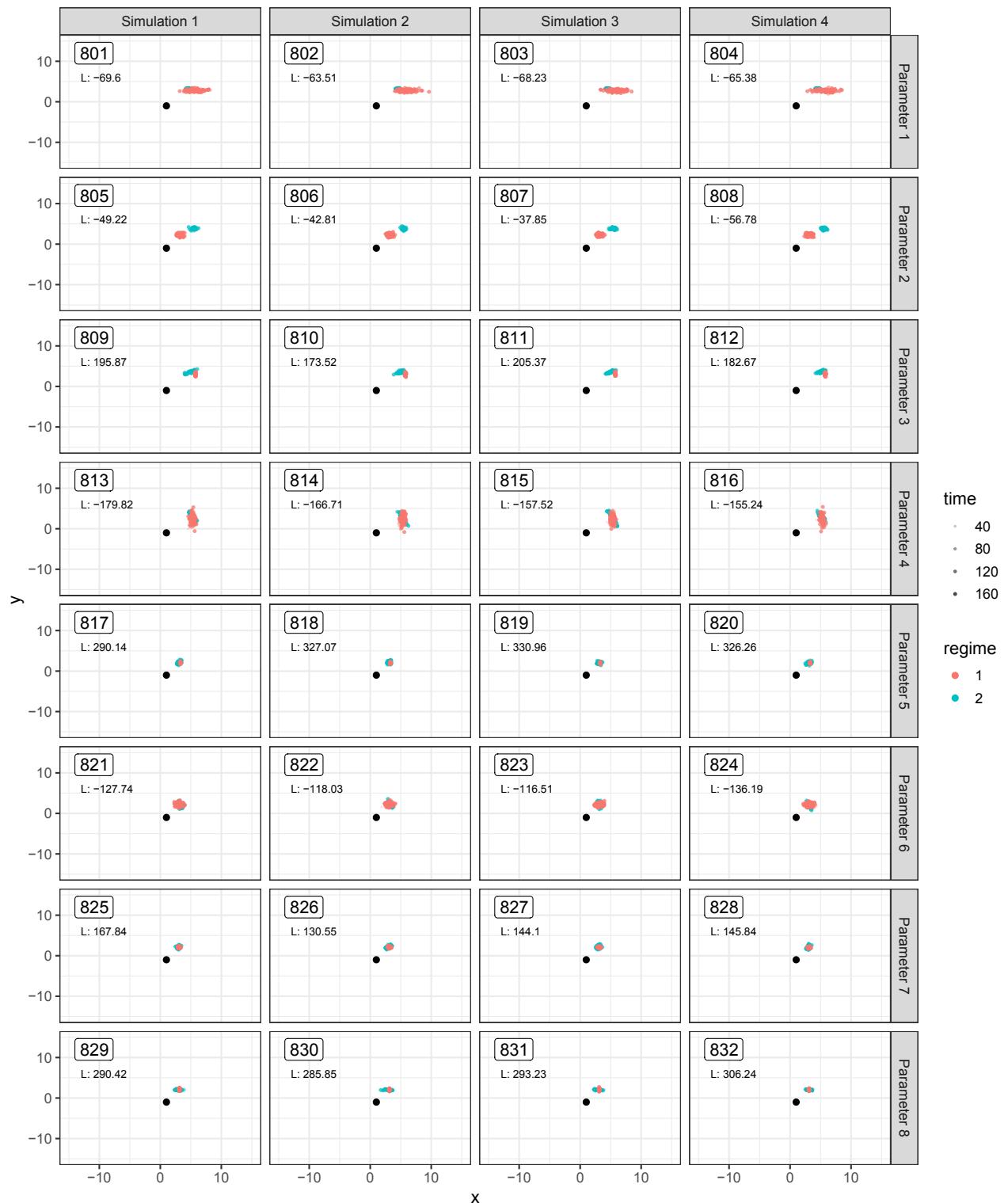
Tree-type non-ultrametric / N=80 / 3 regimes / Mapping 4. AEC



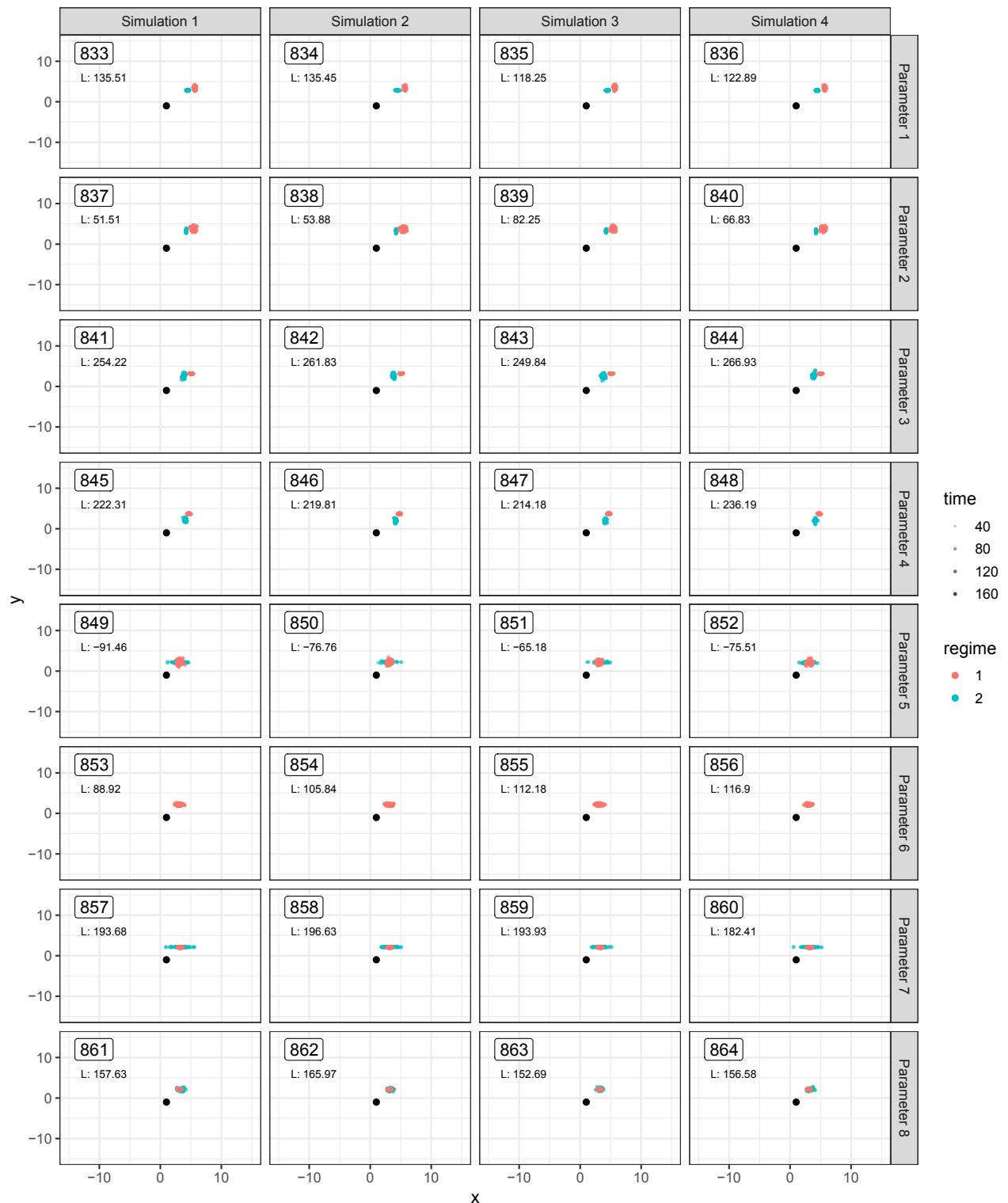
Tree-type non-ultrametric / N=159 / 2 regimes / Mapping 1. AF



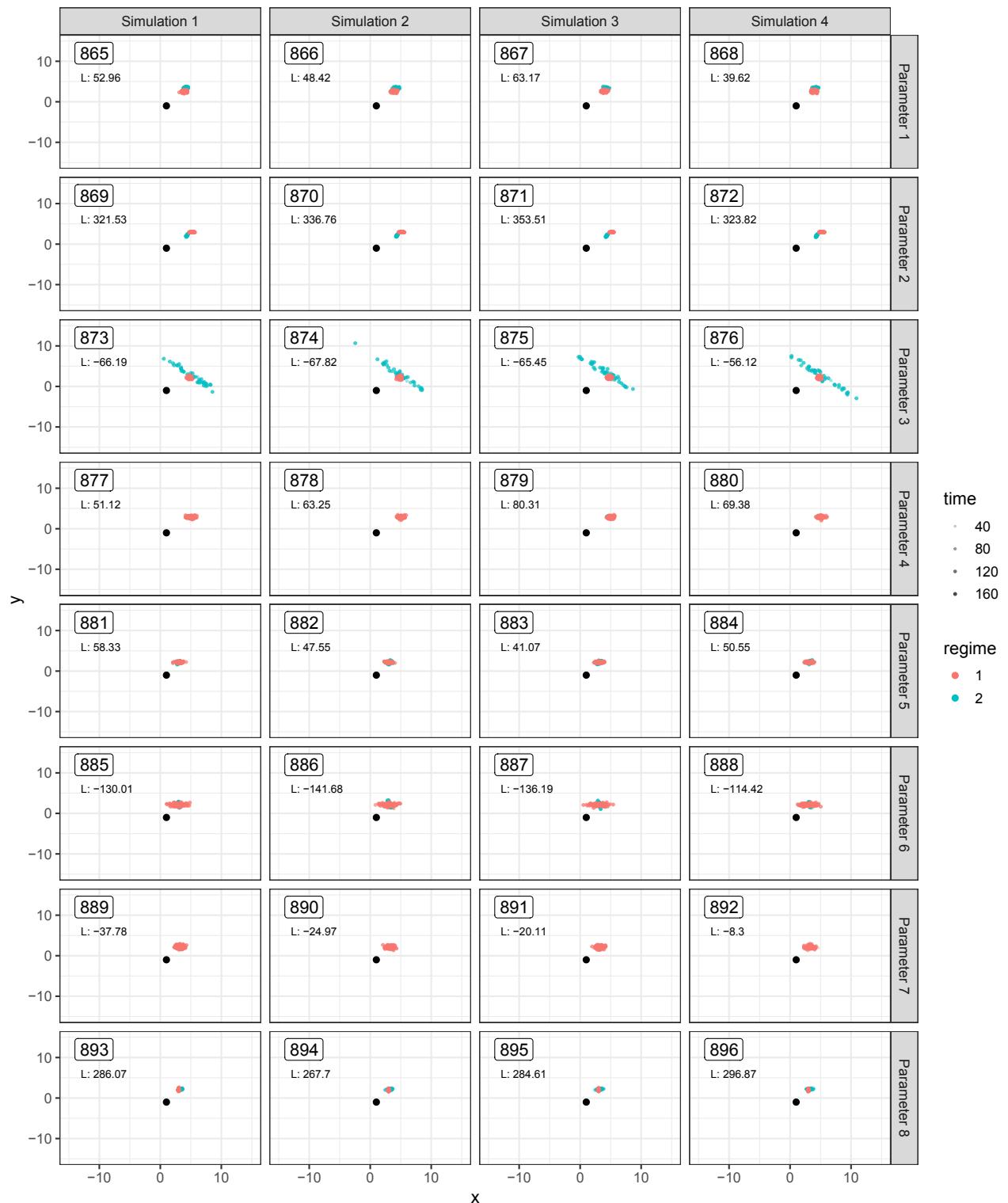
Tree-type non-ultrametric / N=159 / 2 regimes / Mapping 2. CF



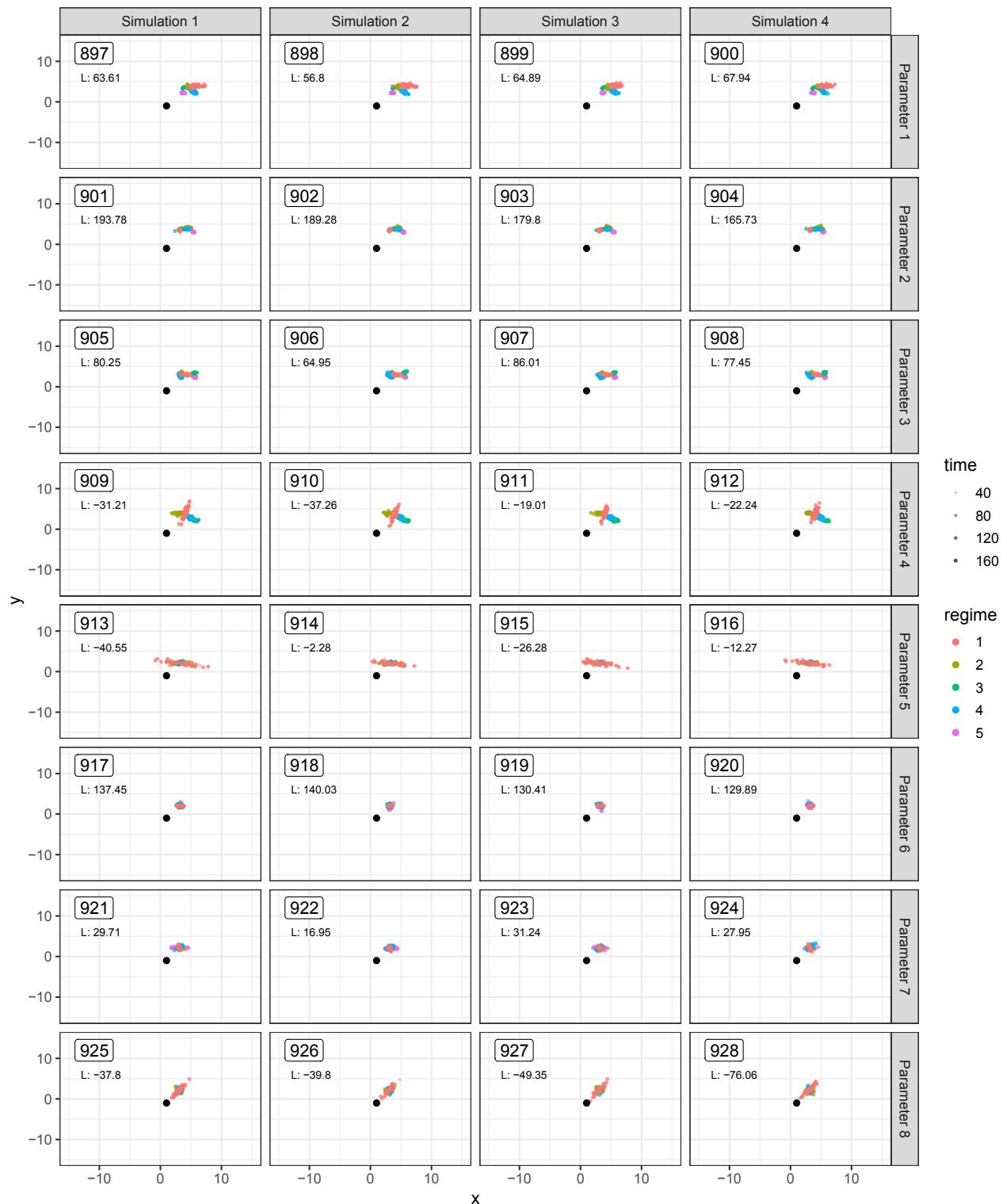
Tree-type non-ultrametric / N=159 / 2 regimes / Mapping 3. CC



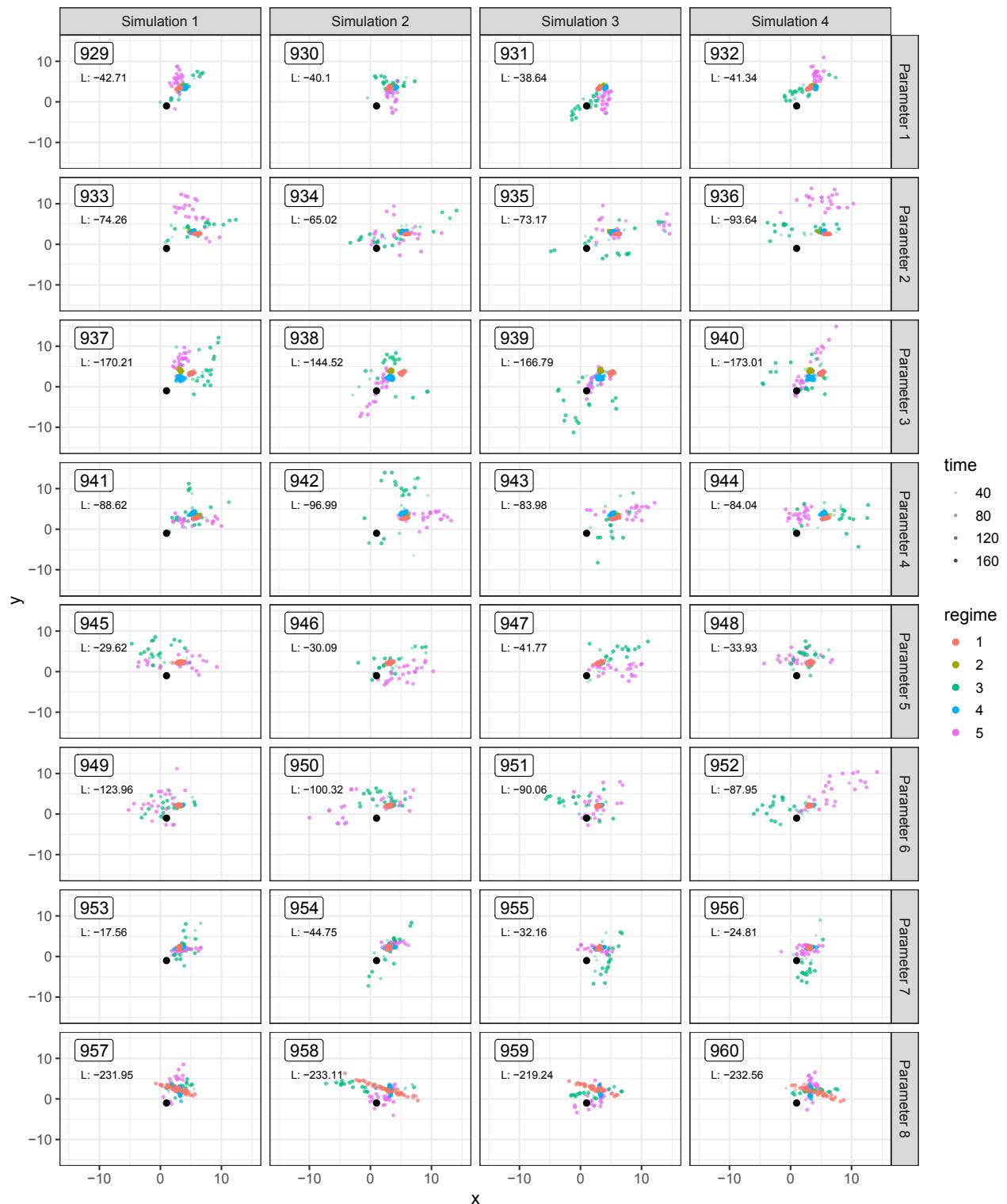
Tree-type non-ultrametric / N=159 / 2 regimes / Mapping 4. CF



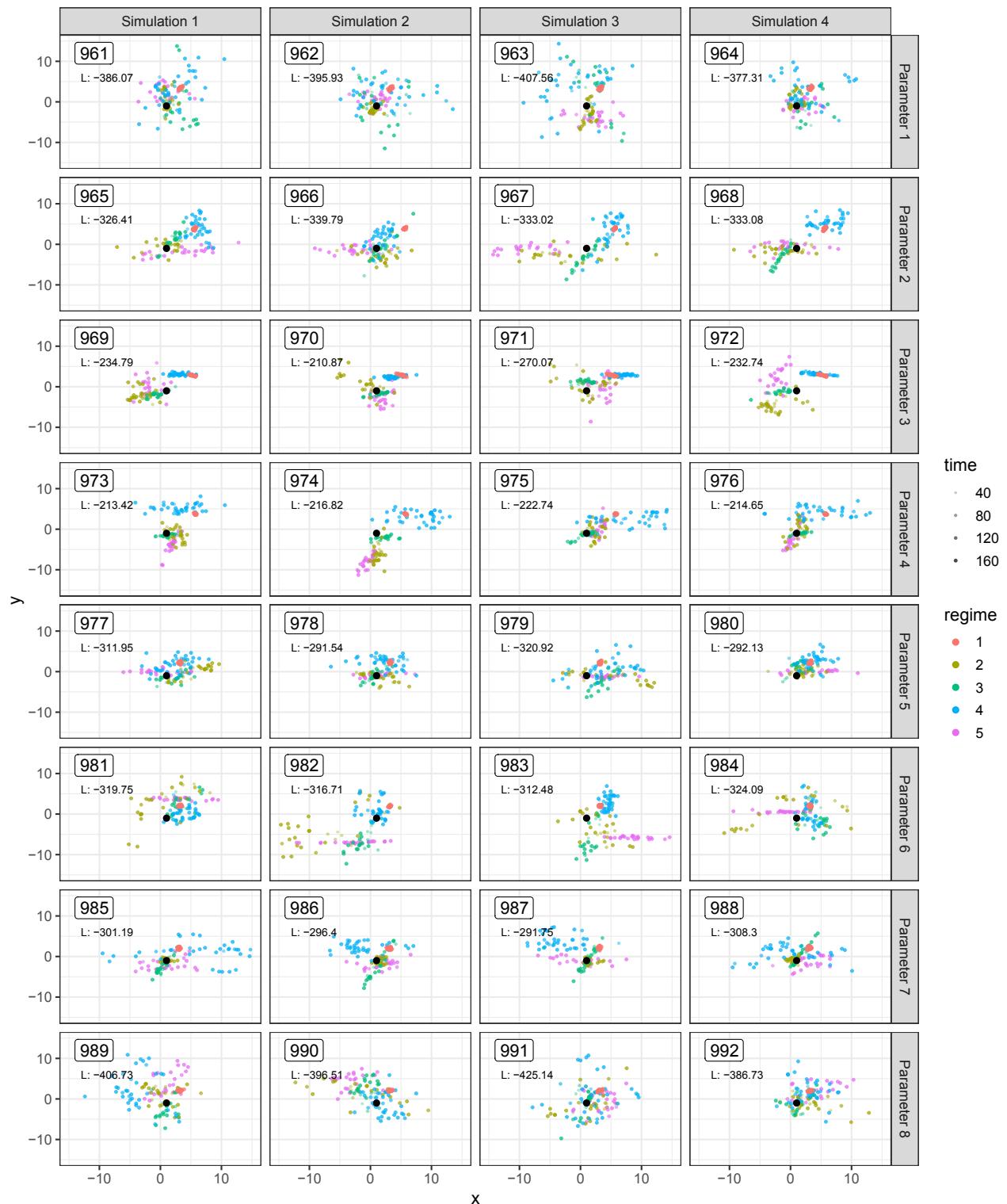
Tree-type non-ultrametric / N=159 / 5 regimes / Mapping 1. FCEFC



Tree-type non-ultrametric / N=159 / 5 regimes / Mapping 2. ECBEB



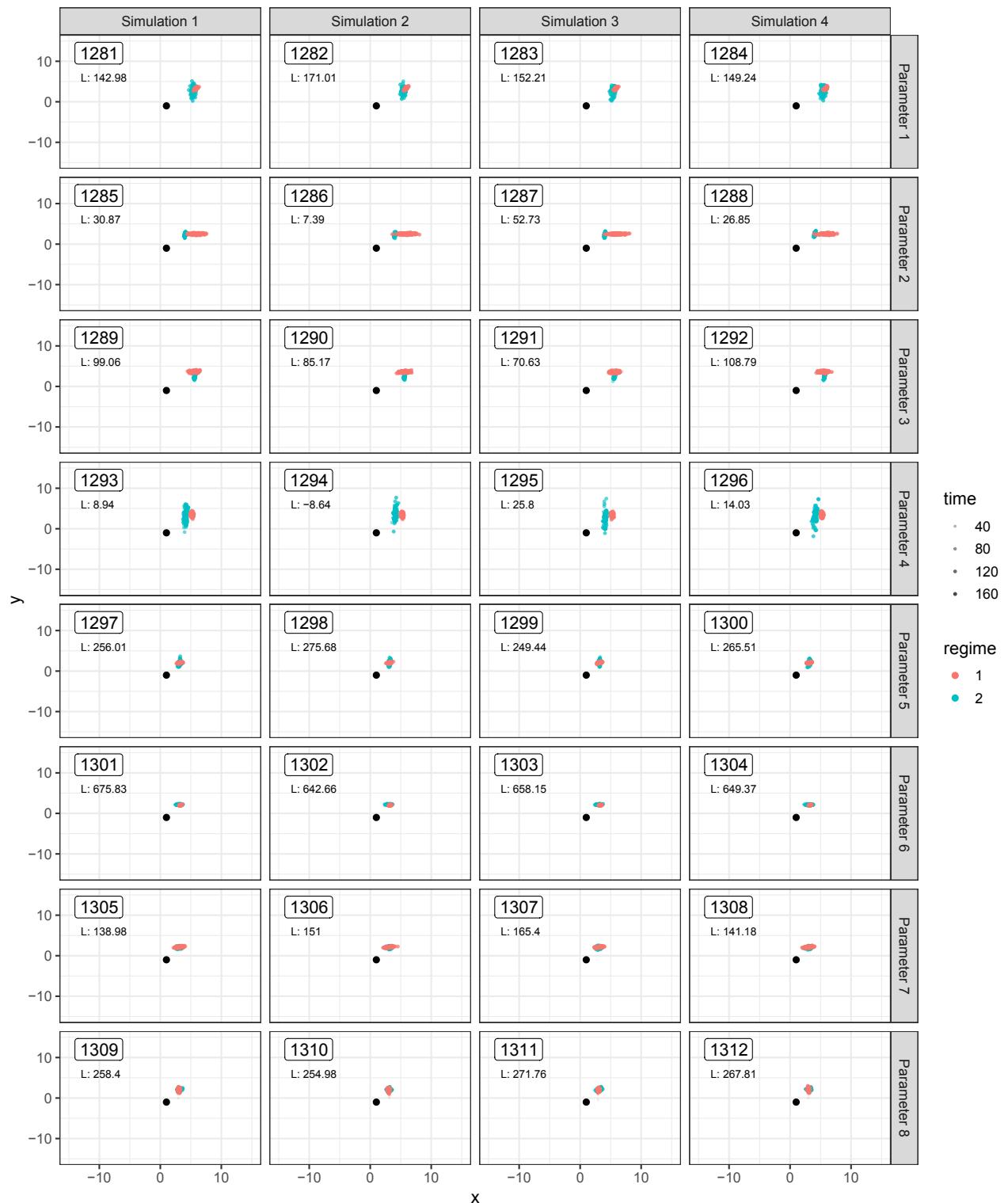
Tree-type non-ultrametric / N=159 / 5 regimes / Mapping 3. EABAB



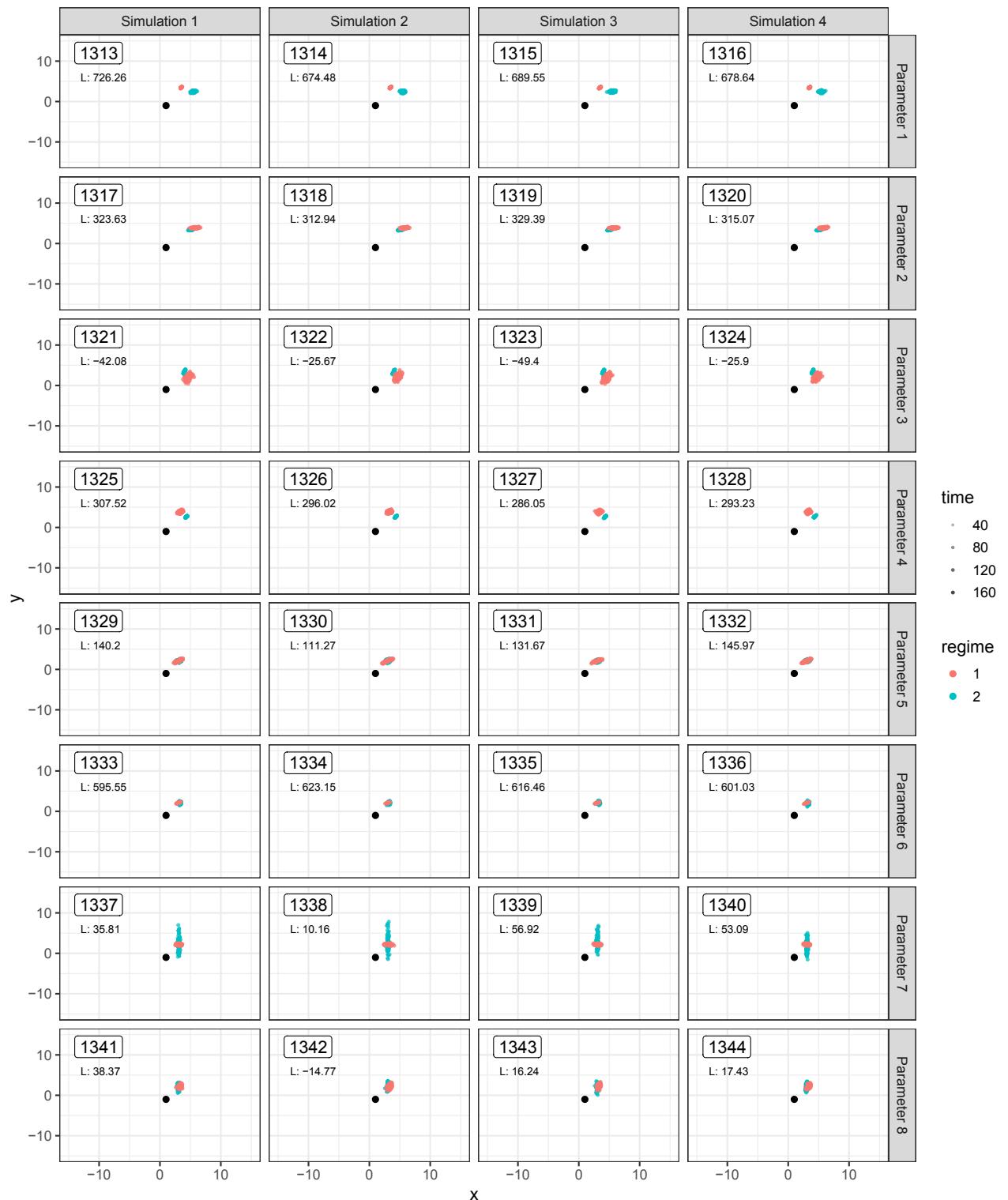
Tree-type non-ultrametric / N=159 / 5 regimes / Mapping 4. ADFEE



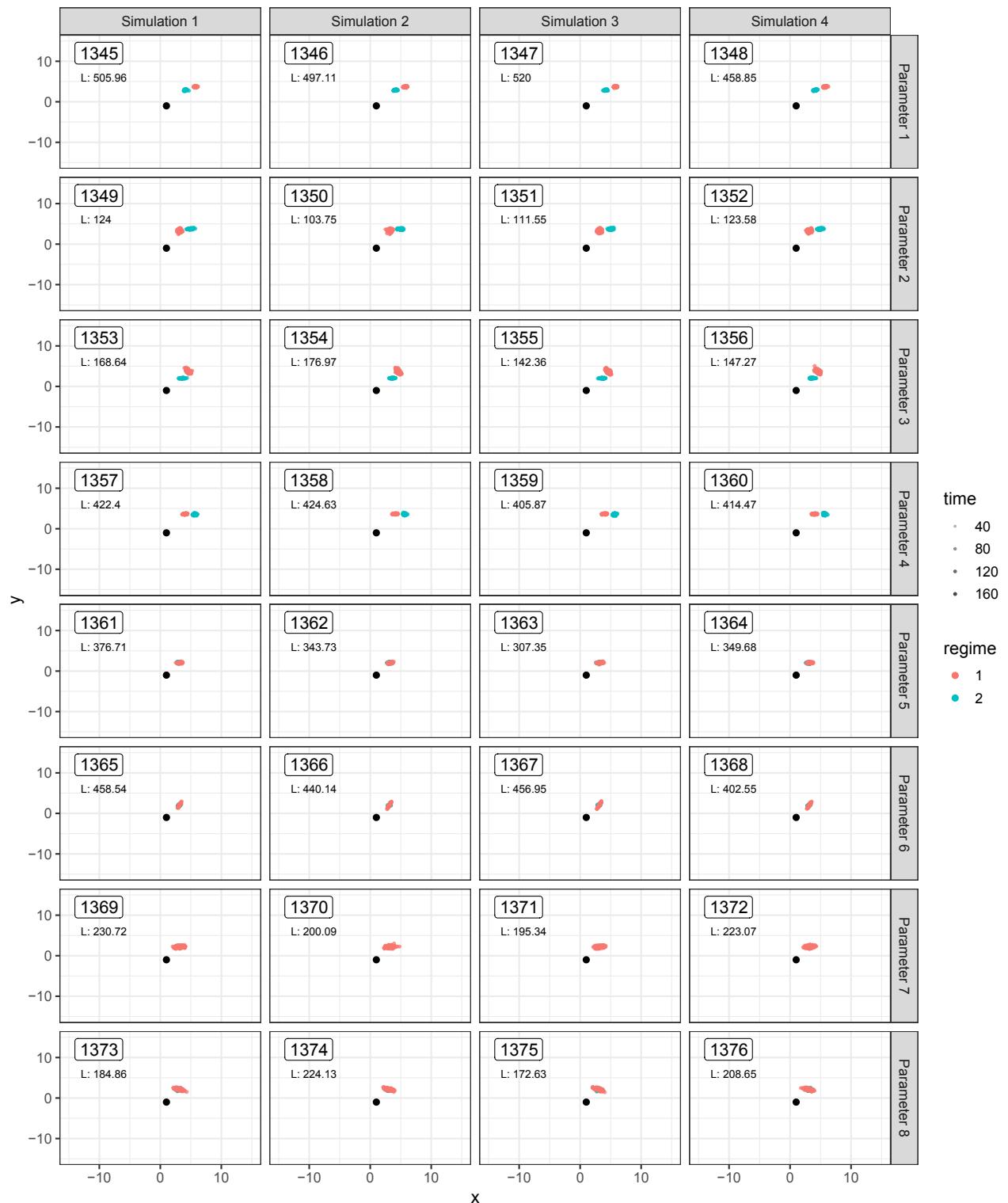
Tree-type non-ultrametric / N=318 / 2 regimes / Mapping 1. DD



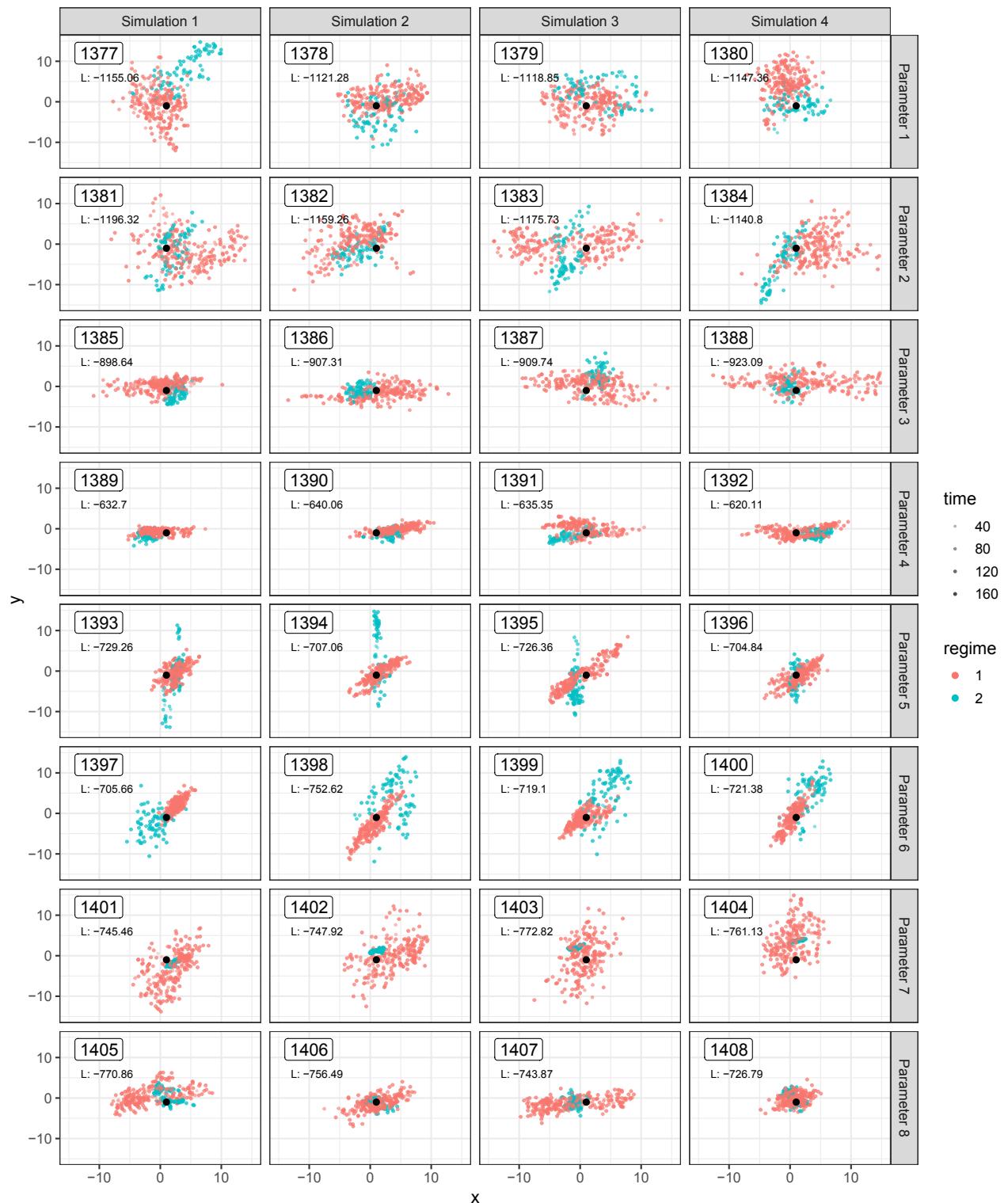
Tree-type non-ultrametric / N=318 / 2 regimes / Mapping 2. FD



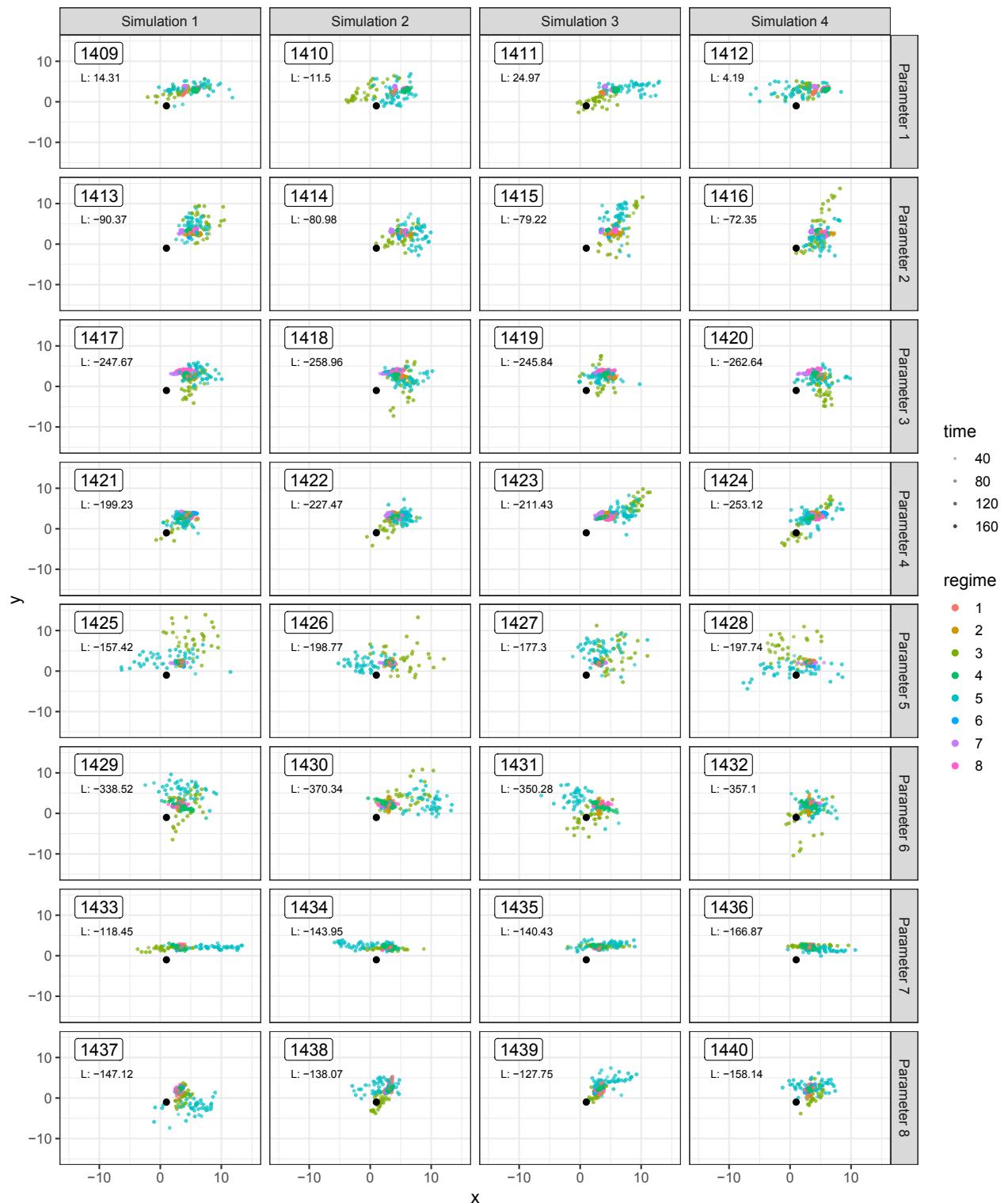
Tree-type non-ultrametric / N=318 / 2 regimes / Mapping 3. ED



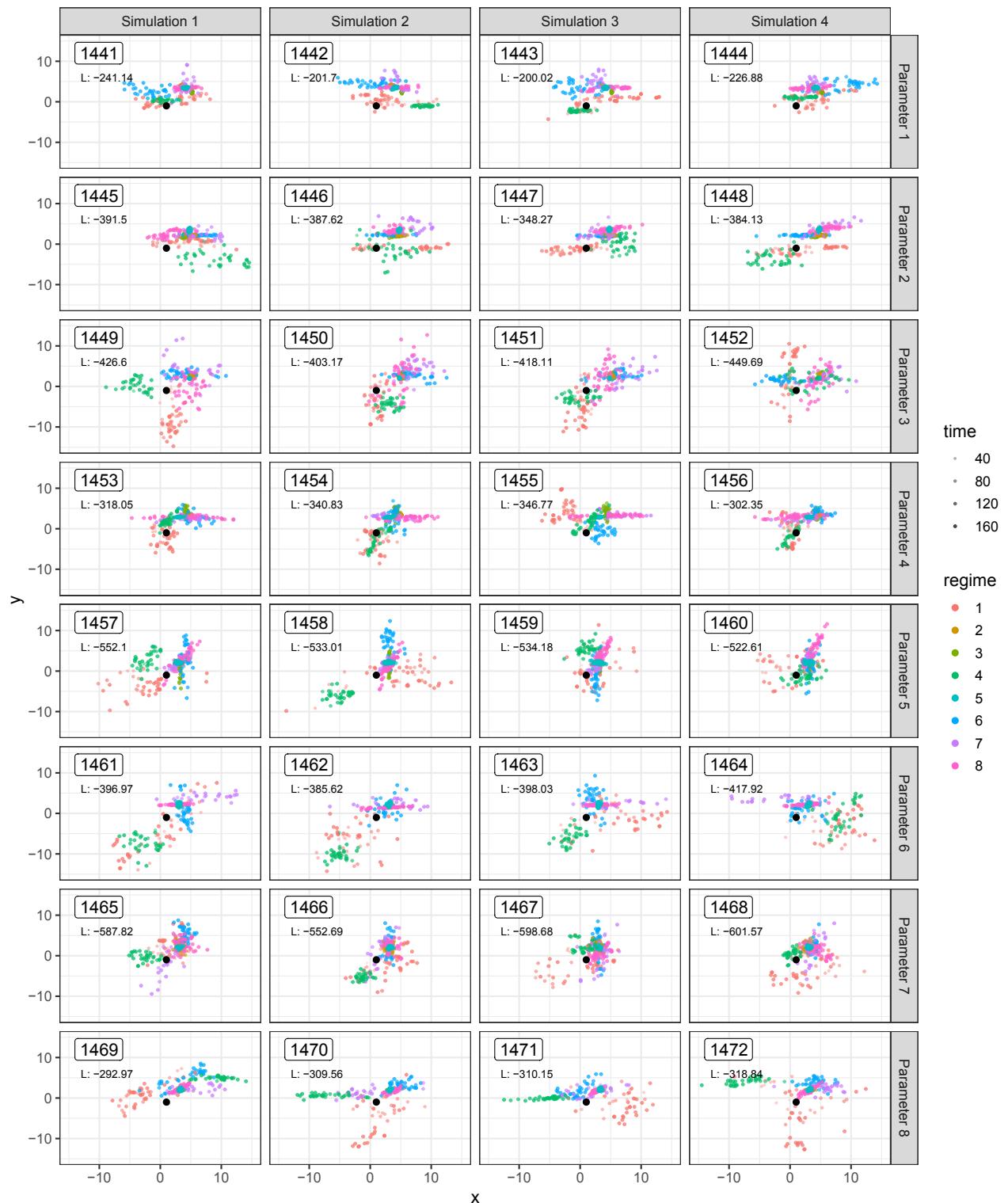
Tree-type non-ultrametric / N=318 / 2 regimes / Mapping 4. BB



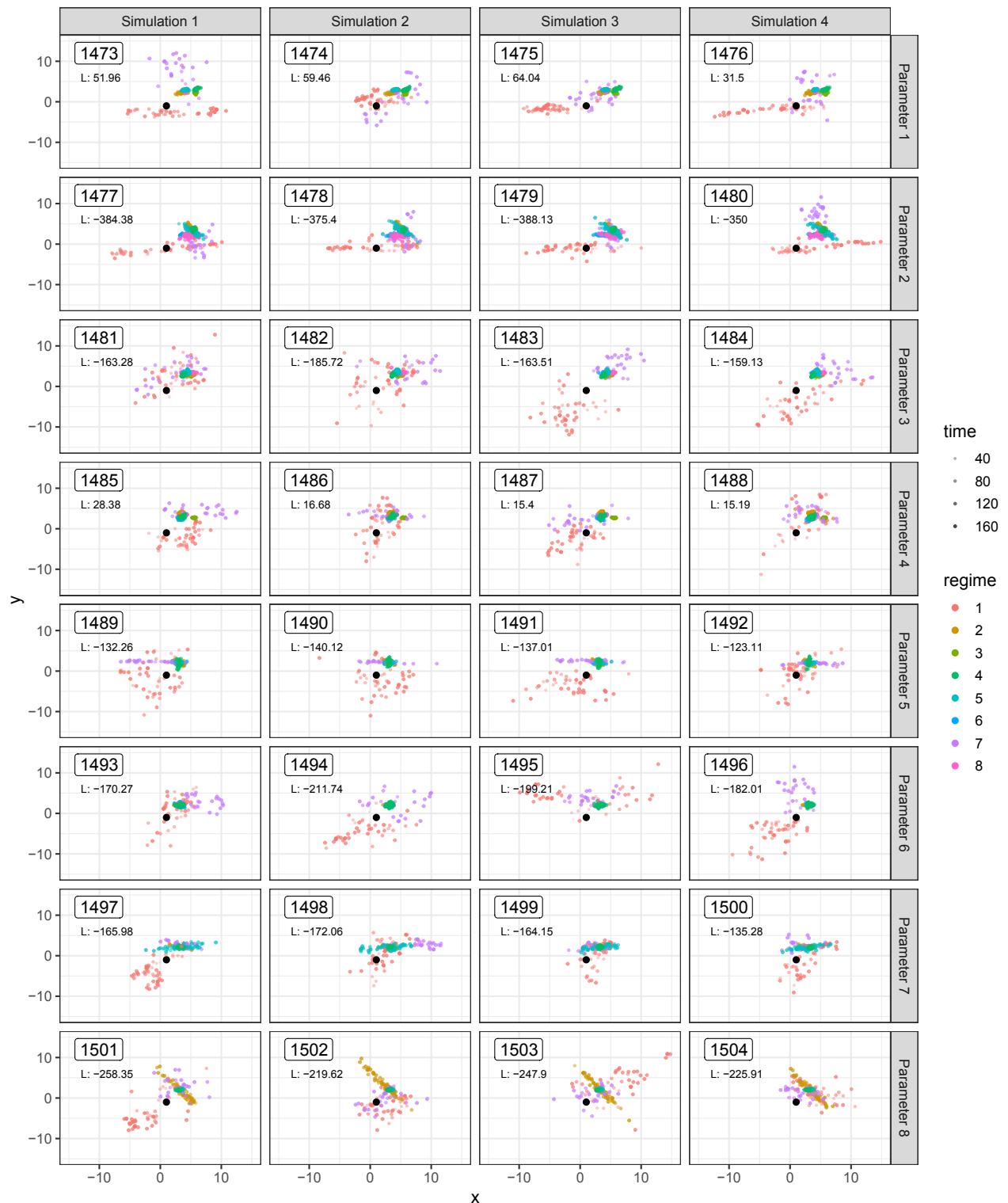
Tree-type non-ultrametric / N=318 / 8 regimes / Mapping 1. ECBEAFDD



Tree-type non-ultrametric / N=318 / 8 regimes / Mapping 2. BCDBDABB



Tree-type non-ultrametric / N=318 / 8 regimes / Mapping 3. BFCEFCAC



Tree-type non-ultrametric / N=318 / 8 regimes / Mapping 4. ECDFFCCF



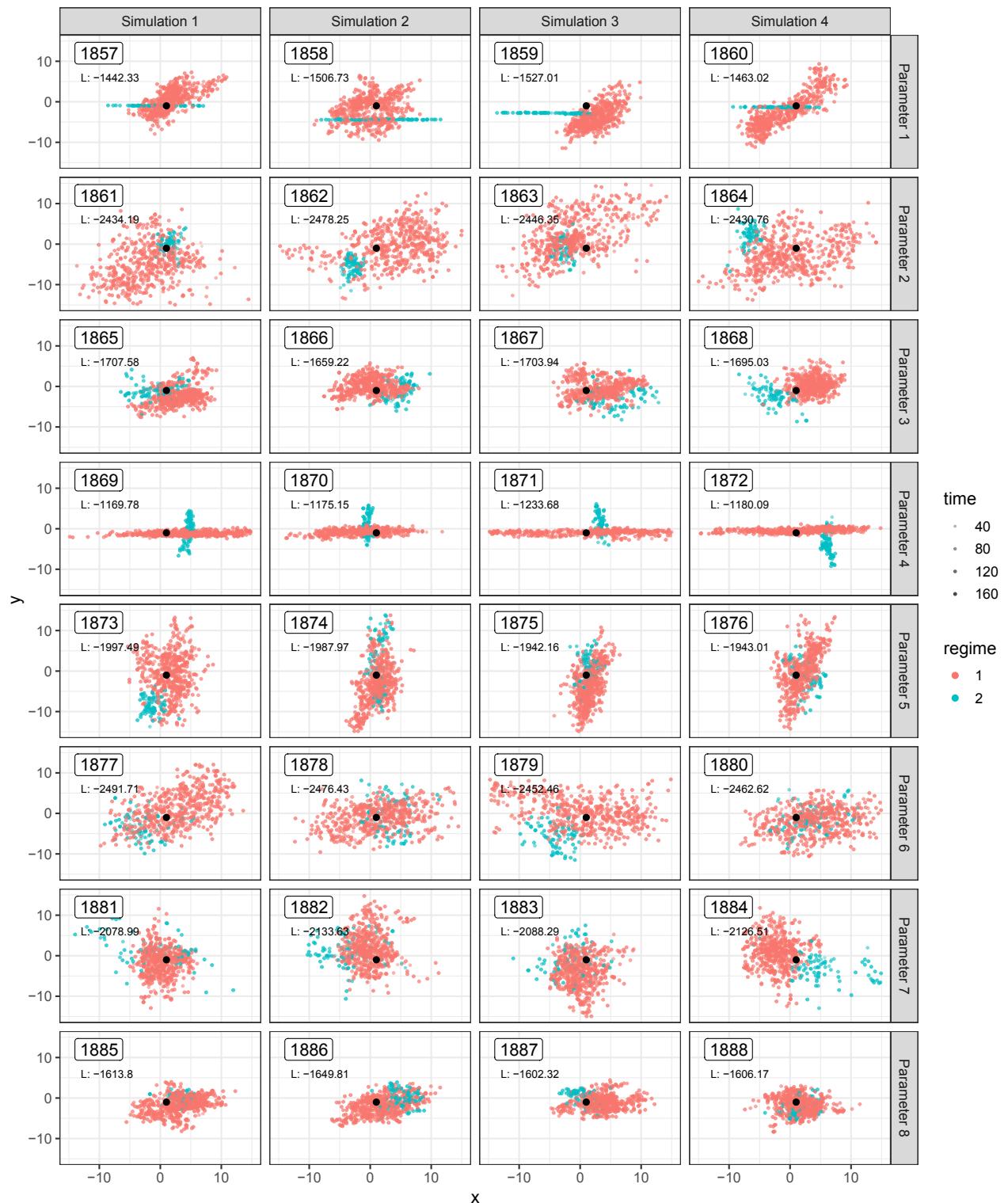
Tree-type non-ultrametric / N=638 / 2 regimes / Mapping 1. DF



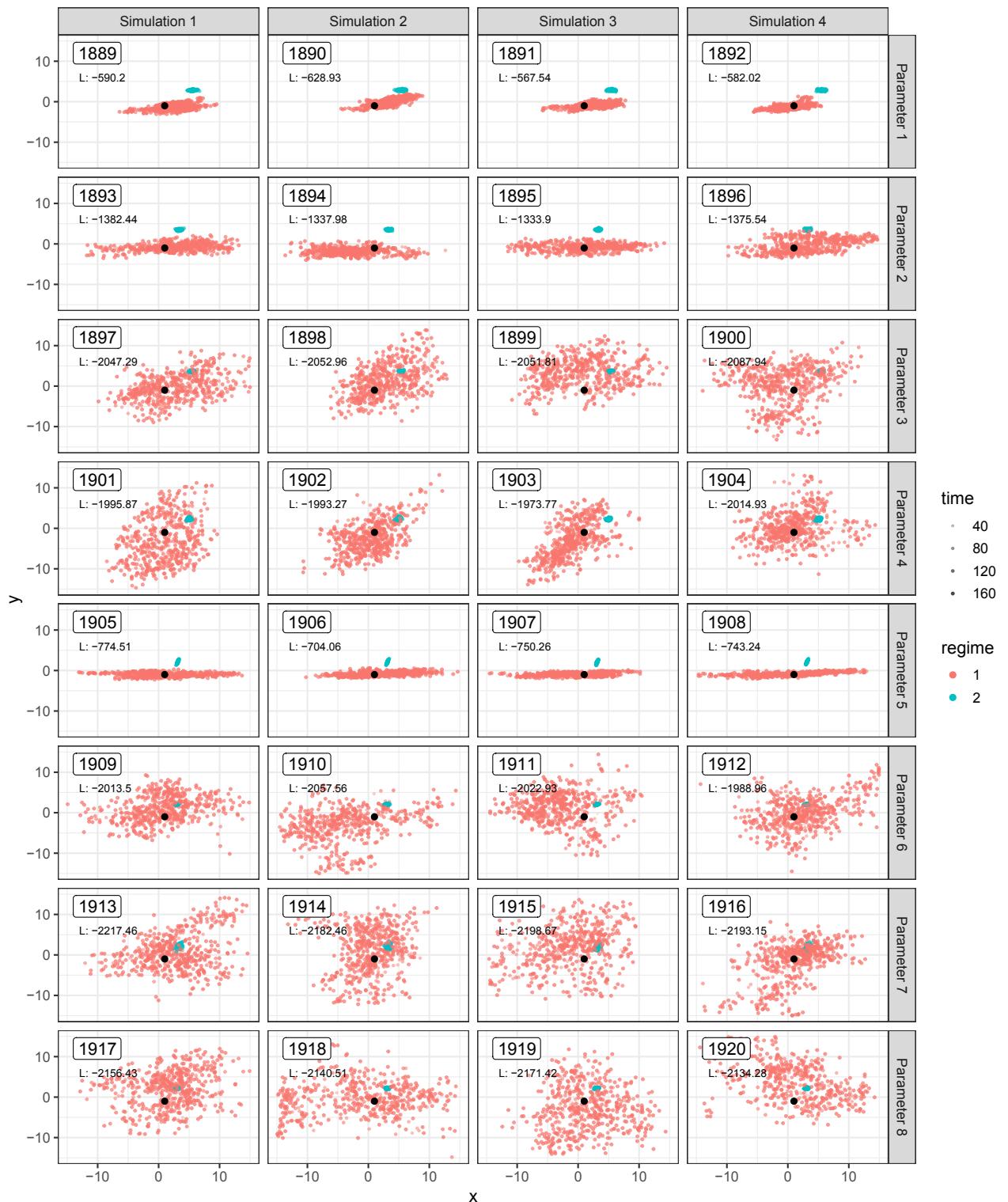
Tree-type non-ultrametric / N=638 / 2 regimes / Mapping 2. FC



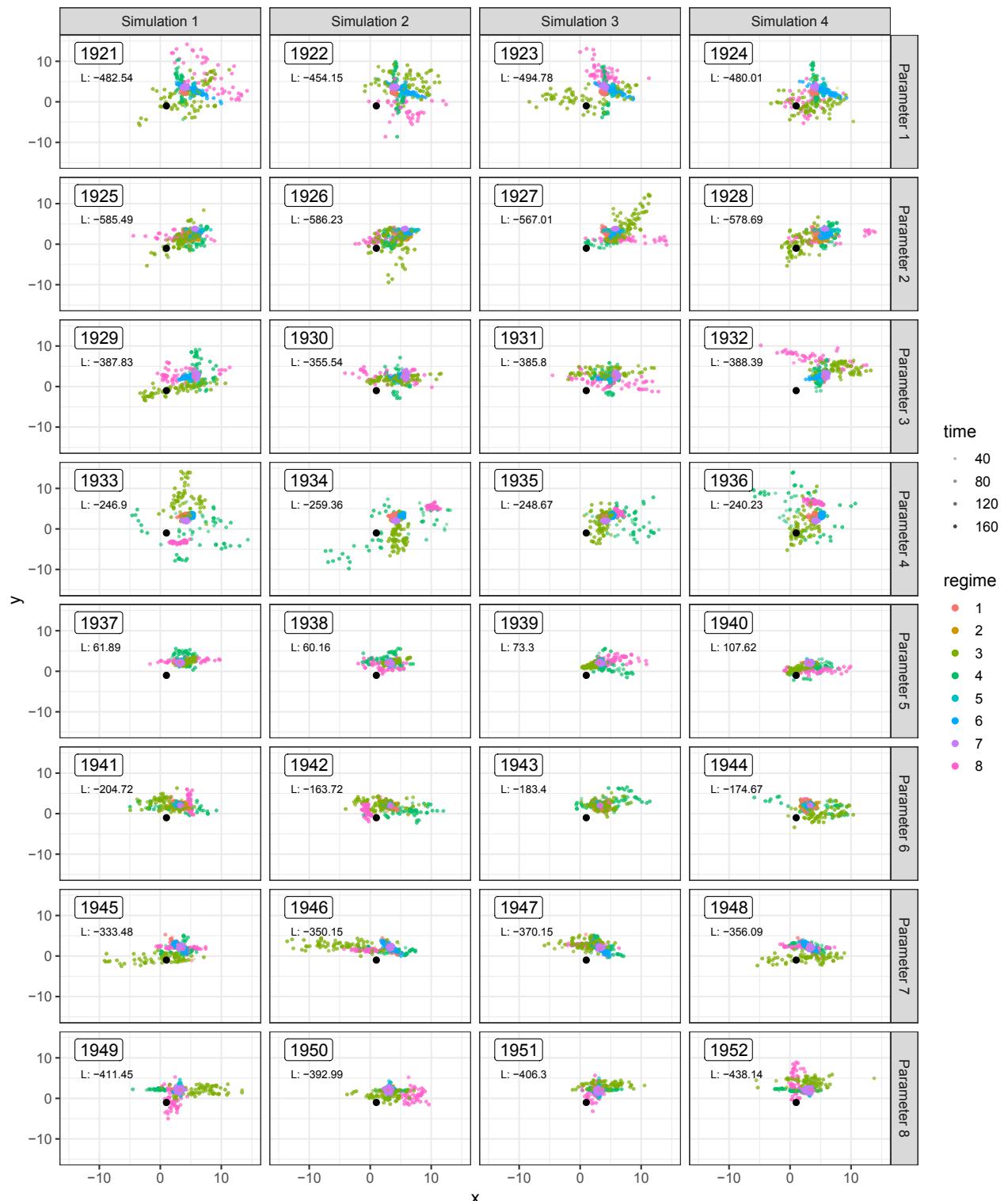
Tree-type non-ultrametric / N=638 / 2 regimes / Mapping 3. BA



Tree-type non-ultrametric / N=638 / 2 regimes / Mapping 4. BD



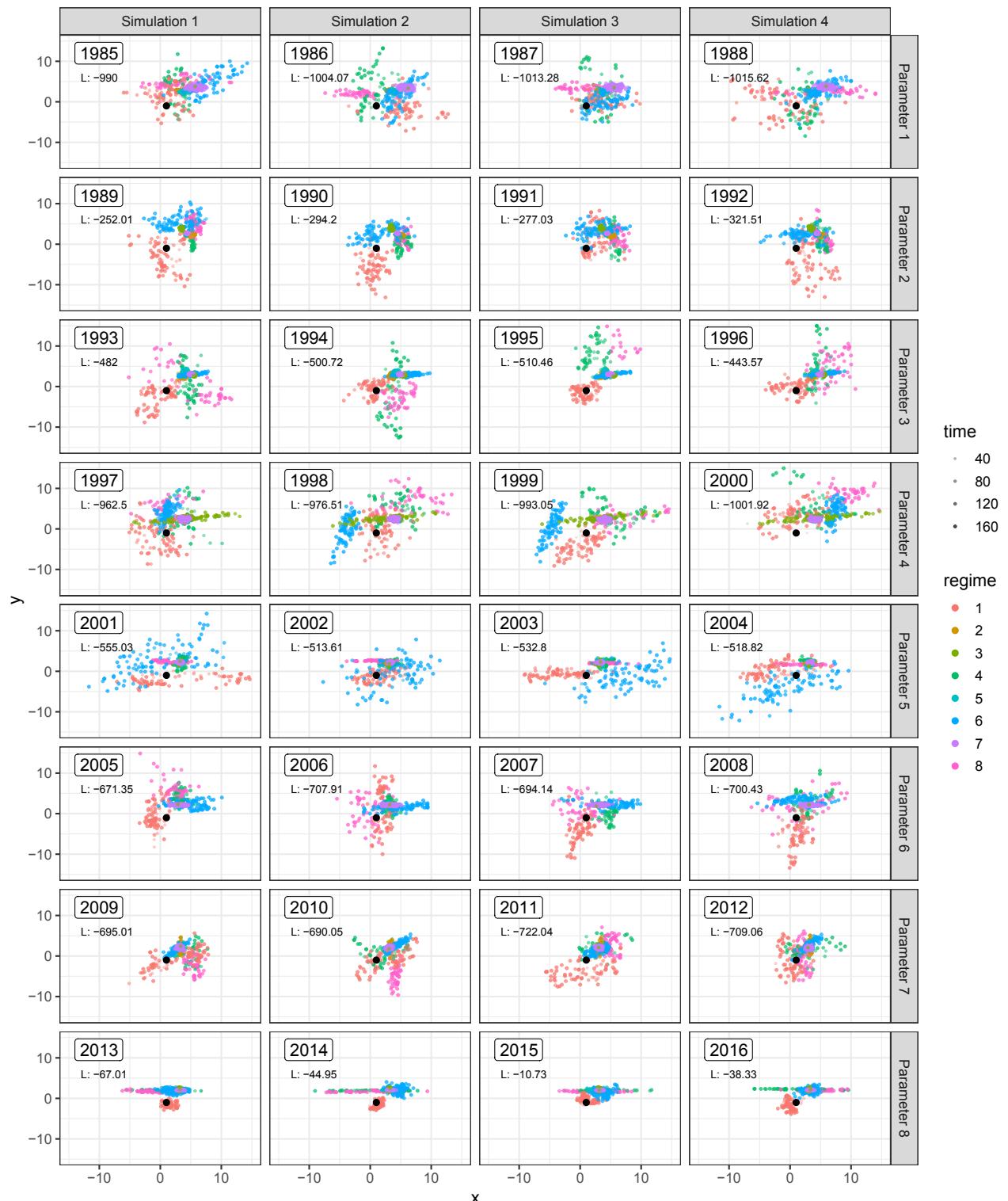
Tree-type non-ultrametric / N=638 / 8 regimes / Mapping 1. FDBACFCA



Tree-type non-ultrametric / N=638 / 8 regimes / Mapping 2. CDCEEDCC



Tree-type non-ultrametric / N=638 / 8 regimes / Mapping 3. BDFACBCA



Tree-type non-ultrametric / N=638 / 8 regimes / Mapping 4. CFEFCDA



```
load("../data-raw/ResultsTestData_t5/TestResultData_t5.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("N=80", "N=159", "N=318", "N=638"), labels=c(80, 159,
`#regimes`=as.integer(numClusters),
`Tree-type`=factor(treeType, levels=c("ultrametric", "non-ultrametric")))] [Crit. %in% c("Clustered", "Unclustered")]
print.xtable(xtable(summaryTable), comment=FALSE, include.rownames = FALSE)

```