How to use the findSegments function to fit a piecewise constant curve

Wolfgang Huber

October 28, 2005

Contents

1 Introduction

The problem of segmenting a series of numbers into piecewise constant segments occurs in multiple application areas. Two examples are

- arrayCGH data, where the segments correspond to regions of copy number gain, loss, or no change.
- tiling microarray data for transcriptomics, where the segments correspond to transcripts. Here we assume that the probe effects (which lead to different fluorescence intensities even for the same mRNA abundance) have been normalized away, so that all probes for one transcript have the same fluorescence (in expectation).

To demonstrate and verify the correctness of the algorithm, let's generate simulated data:

```
> genData = function(lenx, nrcp, stddev = 0.1) {
+     x = numeric(lenx)
+     cp = c(1, sort(sample(1:floor(lenx/15), nrcp - 1) * 15),
+         lenx + 1)
```

```
+     s = 0
+     for (j in 2:length(cp)) {
+         sel = cp[j - 1]:(cp[j] - 1)
+         s = (0.5 + runif(1)) * sign(rnorm(1)) + s
+         x[sel] <- rnorm(length(sel), mean = s, sd = stddev)
+     }
+     return(list(x = x, cp = cp[-1]))
+ }

> lenx = 1000
> nrcp = 10
> gd = genData(lenx, nrcp)
> plot(gd$x, pch = ".")
> abline(v = gd$cp, col = "red")
```

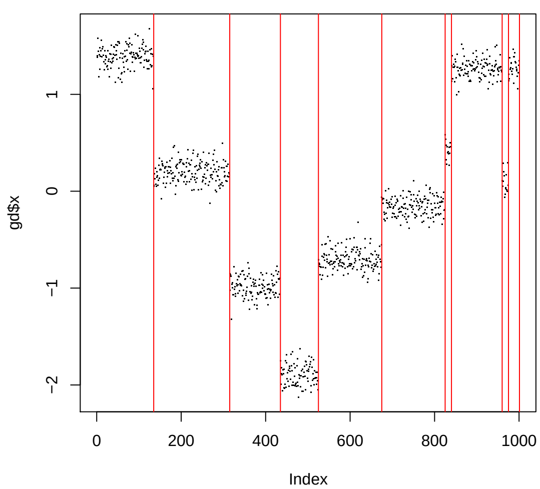

Figure 1: A simulated data example with 10 change points, shown with red vertical lines

The result is shown in Figure 1. We can use the function *findSegments* to reconstruct the changepoints from the `x` data alone.

```
> library(tilingArray)
```

```
> library(tilingArray)
> maxk = 500
> maxcp = 12
> seg = findSegments(gd$x, maxk = maxk, maxcp = maxcp)

Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...

> seg

$J
 [1]        -Inf -0.03080845  1.10556635  1.55645830  2.28610954  2.82401243
 [7]  3.38504510  3.56416206  4.19402379  4.55827036  4.56555013  4.57462247

$th
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
 [1,] 1001    0    0    0    0    0    0    0    0     0     0     0
 [2,]  501 1001    0    0    0    0    0    0    0     0     0     0
 [3,]  315  815 1001    0    0    0    0    0    0     0     0     0
 [4,]  315  525  825 1001    0    0    0    0    0     0     0     0
 [5,]  135  315  525  825 1001    0    0    0    0     0     0     0
 [6,]  135  315  435  525  825 1001    0    0    0     0     0     0
 [7,]  135  315  435  525  675  840 1001    0    0     0     0     0
 [8,]  135  315  435  525  675  840  960 1001    0     0     0     0
 [9,]  135  315  435  525  675  840  960  975 1001     0     0     0
[10,]  135  315  435  525  675  825  840  960  975  1001     0     0
[11,]  135  147  315  435  525  675  825  840  960   975  1001     0
[12,]  135  315  435  525  619  620  675  825  840   960   975  1001

$dat
   [1]  1.3984295294  1.3848577430  1.5772046950  1.4452445036  1.1820121087
   [6]  1.4107103442  1.4787158924  1.2614271595  1.3948922183  1.4515951193
  [11]  1.5572095662  1.3861241726  1.4702013996  1.4938037975  1.4918983259
  [16]  1.4041860665  1.3347949717  1.2557018532  1.4521529824  1.3578407056
  [21]  1.3177590386  1.3981172939  1.3852547964  1.2875382385  1.2933215833
  [26]  1.2586768986  1.4487968718  1.3441888103  1.3649213846  1.1799349128
  [31]  1.4201997394  1.3391962303  1.4150935338  1.2677546761  1.5068794435
  [36]  1.3277542283  1.3701890173  1.3411968009  1.4136363933  1.3790564252
  [41]  1.4902845011  1.4842379842  1.4056873682  1.1239907510  1.5121442011
```

```
 [46]  1.4064965414  1.4523546020  1.3650193858  1.5088147237  1.5406433309
 [51]  1.1693177291  1.2720141389  1.5406490721  1.5286559666  1.3775039867
 [56]  1.1577156381  1.4229082744  1.2138164603  1.2221561301  1.1226730091
 [61]  1.3721114790  1.4217075470  1.3339107239  1.2818945037  1.3901596832
 [66]  1.2317538018  1.4513030818  1.3949746243  1.5424857337  1.3397989609
 [71]  1.4540865269  1.5248869510  1.3554201520  1.2606202070  1.5098289052
 [76]  1.4904879148  1.5785468560  1.5256465345  1.4225422691  1.2396888440
 [81]  1.4174899812  1.4341167125  1.5118522736  1.3857493205  1.3327474820
 [86]  1.3825280254  1.4294581824  1.4093856515  1.2397421423  1.3818228598
 [91]  1.6179350011  1.3664824808  1.4250629202  1.3744651975  1.4239263976
 [96]  1.4072394853  1.6050114564  1.3952723286  1.3979127231  1.3746196944
[101]  1.4748565293  1.4449470548  1.3251481457  1.3780086769  1.3288591495
[106]  1.3907461132  1.4745724372  1.5158585192  1.5620806065  1.4607517637
[111]  1.5379663400  1.2804646742  1.3543582794  1.4201994906  1.4401294877
[116]  1.4950676836  1.4459807870  1.3578448404  1.3413585779  1.2638629917
[121]  1.4473448737  1.3442568448  1.3508432710  1.2641942632  1.6784301766
[126]  1.4593088292  1.3074641798  1.4214903761  1.2863333412  1.4705100885
[131]  1.2936965037  1.4118329418  1.0584847185  1.3951170647  0.0524059778
[136]  0.0639020966  0.1828997651  0.1298910712  0.2391528878  0.0708136253
[141]  0.0503245217  0.1438241865  0.2009299750  0.1604586514  0.0624945405
[146]  0.1111313810  0.2698447499  0.1314144468  0.3405614649  0.2210719012
[151]  0.2243593168  0.2191356595 -0.0792309249  0.3071641733  0.2706595226
[156]  0.2817101150  0.2644835376  0.0826730979  0.1438632828  0.1981464006
[161]  0.2456571628  0.2363544463  0.0408624618  0.2539463125  0.0811553280
[166]  0.2986585938  0.1474576604  0.2098534304  0.1864016645  0.2516172795
[171]  0.2528102750  0.1901243672  0.3143549721  0.0836052311  0.2486122856
[176]  0.2089878524  0.1785137108  0.3395122476  0.1314873568  0.1925804998
[181]  0.1100487704  0.4530908935  0.1335557911  0.4694558052  0.1539878988
[186] -0.0332395192  0.2518422737  0.0549247764  0.2680365623  0.2397813519
[191]  0.2778777401  0.1170580565  0.1021411353  0.4011284651  0.1882213367
[196]  0.2962418052  0.3251699005  0.3028780827  0.1652935439  0.3267960221
[201]  0.1595475968  0.2011699106  0.1760620697  0.2909247394  0.2801880075
[206]  0.0979201109  0.2770932986  0.1958864890  0.1418465832  0.1562066710
[211]  0.0710431665  0.2509159379  0.2949923808  0.0970677235  0.3358685560
[216]  0.4269445723  0.2068224734  0.2861857419  0.2119993574  0.0984845864
[221]  0.2720192037  0.2707388368  0.1248717166  0.3184430388  0.1796300766
[226]  0.0077658348  0.1506558939  0.3946283281  0.4217733741  0.0147862036
[231]  0.1418973230  0.1205699583  0.2514688624  0.1793265995  0.0904906711
[236]  0.2652543625  0.1335685987  0.3608100607  0.2185615012  0.3810961799
[241]  0.0163932349  0.2290352196  0.2662448856  0.2903587895  0.2521926049
```

```
[246]  0.3269799667  0.2356997209  0.0191028325  0.3017449852  0.0991062294
[251]  0.1127226109  0.1504215674  0.4001442615  0.1427445973  0.1743801003
[256]  0.2411356343  0.2714303836  0.0779536157  0.1338622087  0.2927992997
[261]  0.2646001915  0.2168096664  0.2791253188  0.2790689972  0.0595531168
[266]  0.3945158979  0.0956101893 -0.1243133283  0.1636585971  0.1436669863
[271]  0.2384223817  0.1694078126  0.1688766277  0.1270661276  0.3193312480
[276]  0.2017487188  0.3824499160  0.2709325176  0.4228702679  0.2261971322
[281]  0.0095107813  0.1059985602  0.2017320248 -0.0069199842  0.3017168039
[286]  0.1584230080  0.1545423866  0.2205573491  0.0570952712  0.2518187059
[291]  0.1191825523  0.0300869204  0.0697026794  0.0684345540  0.2425271999
[296]  0.0883108715  0.2428055448  0.4935412799  0.0570206762  0.3242567799
[301]  0.2422276714  0.2017154530  0.1502039140  0.1606554934  0.3073737819
[306]  0.1067767823  0.2578438794  0.2587737636  0.1011974109  0.1564076298
[311]  0.2778718427  0.1487778032  0.1922056419  0.1810970255 -1.0241331356
[316] -0.8508655249 -0.8604968401 -0.8758711292 -1.3236909542 -0.9583925156
[321] -0.9437183495 -0.9968205030 -1.0668809760 -1.0699200560 -0.7785808373
[326] -1.0099713872 -0.9853077869 -1.0578078554 -1.0856183938 -0.8488634835
[331] -0.9565938328 -0.9765925828 -0.9117365232 -1.0513169156 -0.9635849288
[336] -0.8774788080 -0.8805762126 -1.0281812813 -0.9532349139 -0.9682556142
[341] -0.8577948121 -0.9634534034 -0.9778997769 -0.8260478364 -1.0777688004
[346] -0.8204980852 -1.0247976115 -1.1373351240 -0.8626556487 -1.1083483138
[351] -0.9104389115 -0.9056230677 -1.0519362461 -1.0741131139 -1.0329637987
[356] -0.9920210936 -0.8779560126 -0.7402320666 -1.1243003890 -0.7965135476
[361] -0.8500255606 -1.2190816702 -1.0363532934 -0.9003084030 -1.0551794049
[366] -1.0407303660 -0.9748320341 -1.1185086583 -1.0844849039 -0.9410427206
[371] -0.9108060147 -0.9608532004 -0.9805900332 -1.1738884608 -0.9521396539
[376] -0.9183511300 -0.9825054918 -1.1111395080 -1.2140674091 -1.1762430600
[381] -1.0470081886 -1.0625636914 -1.0645367448 -0.9727057754 -0.9089722415
[386] -0.9687345968 -1.0504625313 -1.0385455117 -0.8437103971 -1.0441897529
[391] -0.9626804283 -0.8598948906 -1.0012259938 -0.9709371386 -0.9210728229
[396] -1.0255105952 -0.9870240370 -1.0575859830 -1.0612816614 -1.0616331578
[401] -1.1029086946 -0.8467677113 -1.1010808225 -1.0127091059 -0.9640098922
[406] -1.1706097172 -1.0043705825 -0.9600144902 -1.0469960451 -1.0109215183
[411] -1.1033210707 -1.0250251053 -1.0168736008 -0.8562283154 -0.9195913583
[416] -0.9537290697 -0.9779943945 -1.0454676502 -0.8735998682 -1.0669351976
[421] -0.9650020822 -1.0950594831 -1.0500458467 -1.0550768798 -0.8194837555
[426] -0.9412341953 -0.7767252683 -0.8562654168 -1.0969081568 -0.8319646815
[431] -0.9015099458 -0.9098584617 -1.0661164208 -0.8976011090 -1.7521169209
[436] -1.9923447303 -1.8833521397 -1.8218469607 -1.8925220693 -2.0610684141
[441] -1.8421809343 -1.8849116666 -1.8399693732 -2.0289323027 -2.0330491222
```

```
[446] -1.9909133309 -1.7514085726 -2.0231178291 -1.6863695638 -1.7717089276
[451] -1.9367706108 -1.8749561660 -1.9631465561 -1.8485687208 -2.0153851711
[456] -1.8270608010 -1.8059079260 -1.8603750218 -1.9429976574 -1.6832122518
[461] -1.8024091877 -1.8512272613 -2.0034563636 -1.6580713979 -1.9721593960
[466] -2.0031068789 -1.7797539293 -1.9107481764 -1.8234107108 -2.0099857567
[471] -1.9928459253 -1.9833506419 -2.0115793707 -1.8034308059 -1.9730714469
[476] -1.7299235690 -2.0092811661 -2.1249392616 -1.8962425879 -1.9290530305
[481] -1.6244598705 -1.9071552335 -2.0168930435 -1.9060954365 -1.8554879964
[486] -2.0636902443 -1.8386598477 -2.0626025871 -1.9165138940 -1.9128137748
[491] -1.8691448891 -1.8062670886 -1.9121633550 -2.0503225877 -1.7569628241
[496] -1.7627798835 -1.7891963821 -1.8532061277 -1.9309718363 -1.9633432595
[501] -1.8366150395 -1.8602245082 -1.7012620490 -1.8127080605 -1.7859272353
[506] -1.7626087767 -2.0745714771 -2.0086439409 -1.7159015976 -1.8159324951
[511] -1.9498595649 -1.8831270160 -1.7377859715 -1.9378186518 -1.9989905398
[516] -1.8537748022 -1.9282587169 -1.9277694578 -1.8709409566 -1.9936958079
[521] -1.9276545856 -1.8674180317 -1.9577089085 -2.0520478652 -0.7051855926
[526] -0.7430520457 -0.7769650219 -0.7532450127 -0.8631139558 -0.7796674160
[531] -0.7849906390 -0.9111414989 -0.6694052050 -0.5537295695 -0.8665695512
[536] -0.7830459855 -0.6810094312 -0.6442512140 -0.7132218098 -0.6582670329
[541] -0.5458739237 -0.7922827740 -0.5404960512 -0.6632900585 -0.7266272535
[546] -0.7026618670 -0.8754273389 -0.4692440892 -0.7430651782 -0.6544966740
[551] -0.7386957547 -0.7599513404 -0.5117163497 -0.7236504298 -0.5938908159
[556] -0.7218880787 -0.7551587579 -0.6013340599 -0.8637366896 -0.8002363304
[561] -0.7428618534 -0.7275141291 -0.7806849127 -0.7307069858 -0.5606969489
[566] -0.6254800662 -0.6617240615 -0.7017235136 -0.6709856242 -0.8148846028
[571] -0.5358114575 -0.6913603794 -0.7503793321 -0.7690333462 -0.7195123508
[576] -0.7338344471 -0.7098457878 -0.6621247781 -0.7695048426 -0.5318733513
[581] -0.7386779378 -0.7789186272 -0.6608272460 -0.7437341189 -0.8682564768
[586] -0.7562058456 -0.6781637764 -0.7943585065 -0.6233163489 -0.6649805388
[591] -0.5763297737 -0.4936918980 -0.6535743197 -0.7783030950 -0.8531827501
[596] -0.7019933855 -0.6760194612 -0.7705252786 -0.7504242832 -0.4893925017
[601] -0.7696444888 -0.7499308660 -0.7548893731 -0.6243477147 -0.6002730844
[606] -0.7999381174 -0.6055158556 -0.7815653533 -0.4811386889 -0.5766550853
[611] -0.7405784575 -0.6993434883 -0.7171919092 -0.6277032259 -0.5785665625
[616] -0.8098930427 -0.5800943625 -0.6042540194 -0.3220722112 -0.6606203081
[621] -0.6676309653 -0.7075905157 -0.6259131521 -0.6463038968 -0.5918057568
[626] -0.8055328385 -0.7811425937 -0.7476571078 -0.8905310886 -0.6953664385
[631] -0.7698665907 -0.8706927487 -0.7036929493 -0.8372220823 -0.5861042884
[636] -0.4882157519 -0.7650700035 -0.7134104000 -0.6985061328 -0.7156308087
[641] -0.9385220661 -0.7638274403 -0.9062188609 -0.7780304046 -0.6913759473
```

```
[646] -0.7253407031 -0.5394069085 -0.4914440794 -0.7264924647 -0.6431622098
[651] -0.7879393045 -0.7633582931 -0.6575858602 -0.8255492626 -0.8238712445
[656] -0.8054804392 -0.7482203581 -0.7724945082 -0.6193463071 -0.7719927223
[661] -0.7008097866 -0.7537815733 -0.6551426968 -0.7350710020 -0.7801908157
[666] -0.6831358298 -0.9197079165 -0.6591627250 -0.8519975856 -0.5703800622
[671] -0.7025982397 -0.7271396099 -0.7873143939 -0.5527848034 -0.0606987319
[676] -0.0863402605 -0.0712769642 -0.1216544645 -0.0782819689 -0.2892863621
[681] -0.2184378038 -0.2351861419 -0.3066676336  0.0030977557 -0.2265132595
[686] -0.0932732553 -0.1205610935 -0.2789901285 -0.1692343616 -0.1250645675
[691]  0.0272035055 -0.1415284483 -0.1154930863 -0.1597271614 -0.1657410581
[696] -0.1662811528 -0.0808358069 -0.1365861846 -0.2709634066 -0.0573513812
[701] -0.2252305162 -0.1764542519 -0.1608373518 -0.2346637690 -0.2634304542
[706] -0.0494657659 -0.1425112106 -0.2640811563 -0.1214422221 -0.2770831125
[711] -0.0814643093 -0.1044233093 -0.0984324206 -0.2569536277 -0.1868993825
[716] -0.2183726165 -0.1752827593 -0.2143865688 -0.3489829116 -0.0463096807
[721] -0.2134478582 -0.1578131769 -0.1673021627 -0.2455952621 -0.0721184829
[726]  0.0069604228 -0.1751744973 -0.1349864526 -0.2637742320 -0.1310772280
[731] -0.1738196250 -0.2318628901 -0.2910731002 -0.0611745176 -0.3195303345
[736] -0.1873190622 -0.2626286672 -0.2581539389 -0.0023791880 -0.3836324938
[741] -0.1416420678 -0.0074010994 -0.1469656449 -0.0905454347 -0.0346630857
[746] -0.1668348374 -0.2195811040 -0.1506031371 -0.1835987507  0.1070334031
[751] -0.0381457508 -0.1047733496 -0.2365054376 -0.2737632884 -0.2416246942
[756] -0.0760619383 -0.2572390590 -0.1514160650 -0.1060496401 -0.0433048762
[761] -0.1166074198 -0.1909149834 -0.0018417492 -0.1515408000 -0.1748902108
[766] -0.0812081440 -0.1013548075 -0.2525895237 -0.1610624849 -0.1202054115
[771] -0.3210042302 -0.1102939841 -0.2347150099 -0.3122301076 -0.0834423919
[776] -0.0506876927 -0.2235533255 -0.1924337518 -0.0777165831  0.0642513973
[781] -0.2335537258 -0.1639145560 -0.1265766667 -0.0911837947 -0.2724505218
[786] -0.0907928321 -0.3708851732  0.0201337532  0.0374922290 -0.0143682338
[791] -0.2765320496 -0.1275599085 -0.1888525226 -0.2229120052 -0.1908404263
[796] -0.0238639209 -0.2699383166 -0.3150248602 -0.0474233819 -0.1264026553
[801] -0.1682175644 -0.1714573631 -0.2472525187 -0.2364519557 -0.0134052403
[806] -0.0482268116 -0.1267668170 -0.2344406892 -0.0555460204 -0.1700273328
[811] -0.2408891013 -0.1130216631 -0.1557793076 -0.2338942100 -0.0615887391
[816] -0.1138582839 -0.1036162250 -0.3395241523 -0.1093769949 -0.0785355183
[821] -0.0130748823 -0.2858953439 -0.1948639310 -0.1916074171  0.5837762621
[826]  0.5375434845  0.4746209541  0.3263776726  0.2807492254  0.4596789434
[831]  0.3974222458  0.4064026861  0.3843908104  0.2683316052  0.4557607028
[836]  0.3898200991  0.4993900488  0.4577662271  0.4571614088  1.2928982953
[841]  1.1362979086  1.3983924950  1.1614693727  1.2805202321  1.3117345994
```

```
[846]  1.2785415874  1.1322094176  1.2927050743  1.3645464340  1.3389966337
[851]  1.1702551184  0.9976907655  1.2794122508  1.4148877524  1.2773132610
[856]  1.2433150066  1.0264684445  1.2099638233  1.2691208370  1.1988525332
[861]  1.1917390625  1.2748550425  1.2485821538  1.5174618466  1.2536841839
[866]  1.3201347149  1.3102458628  1.4729913608  1.1963108518  1.3953621374
[871]  1.2588850327  1.2183921569  1.2178613234  1.3006496229  1.2126333918
[876]  1.3447625728  1.2925498629  1.2142655481  1.2337028497  1.1863469589
[881]  1.3479661357  1.1331007434  1.3084482280  1.1456585962  1.3008112835
[886]  1.1350792802  1.2943659102  1.2451631493  1.2953495089  1.3376935769
[891]  1.4265548221  1.2595655005  1.3284244084  1.2628951973  1.2426620732
[896]  1.2573188622  1.3574921128  1.4506052444  1.2758599295  1.3873231299
[901]  1.2849358392  1.3344417995  1.3159154427  1.2257059281  1.2265857558
[906]  1.1716616519  1.1322276172  1.1668926074  1.3116722250  1.1509438942
[911]  1.2180326088  1.3535627752  1.2670037934  1.4218614369  1.2884824988
[916]  1.2005022917  1.3600216581  1.2863911340  1.1891240599  1.3253358053
[921]  1.1878211755  1.3232457980  1.3641405215  1.3019045523  1.4592915768
[926]  1.3130645471  1.0598980859  1.2497367615  1.1230752214  1.3930105898
[931]  1.3169627476  1.2115663179  1.2030962340  1.2492883815  1.2490198704
[936]  1.2952483291  1.2565808756  1.1031449863  1.2394829616  1.1918840413
[941]  1.2428143846  1.2829121128  1.3025702216  1.4900078838  1.3462935301
[946]  1.3090099892  1.5058950446  1.1318975479  1.2537711128  1.2679601921
[951]  1.2038619530  1.2377405704  1.2787239186  1.2762865202  1.4099542052
[956]  1.2849063137  1.1339055084  1.2472068454  1.2342018855  0.1064263596
[961]  0.0495401168  0.2905256812  0.1050111491  0.2014016011  0.1583079110
[966] -0.0617431948 -0.0300413230  0.0232676259  0.0355718333  0.1709404229
[971]  0.0141027134  0.0005742359  0.2935830800  0.0631113249  1.2057671952
[976]  1.1426238010  1.1592634227  1.3794100263  1.2592734360  1.2734945741
[981]  1.3207588392  1.3280818815  1.2398828060  1.2090403645  1.2620732374
[986]  1.4678454360  1.3860161012  1.1153535214  1.3860966518  1.4282148084
[991]  1.3374214686  1.3262547682  1.2806063803  1.2449742867  1.2591888964
[996]  1.1880977430  1.0581393554  1.2255303597  1.3025731186  1.3453091401

$call
findSegments(x = gd$x, maxcp = maxcp, maxk = maxk)

attr(,"class")
[1] "segmentation" "list"
```

> gd$cp

```
[1]  135  315  435  525  675  825  840  960  975 1001
```

We see that the 10-th row of the matrix `seg$th` exactly reconstructs the change points `gd$cp` that were used in the simulation.

The parameters `maxcp` and `maxk` are the maximum number of segments and the maximum length per segment. The algorithm finds for each value of k from `1` to `maxcp` the best segmentation under the restriction that no individual segment be longer than `maxk`. In the original paper of Picard et al. [1] and in their software, `maxk` is implicitely set to the number of data points `length(x)`. I have introduced this parameter to reduce the algorithm's complexity. The complexity of Picard's software is `length(x)*length(x)` in memory and `length(x)*length(x)*maxcp` in time, the complexity of the *findSegments* function is `length(x)*maxk` in memory and `length(x)*maxk*maxcp` in time. As I am envisaging applications with `length(x)` $\approx 10^5$ and `maxk` ≈ 250, the difference can be substantial.

2 How to choose the number of segments?

Need to assess goodness of fit (contained in `gd$J`) together with model complexity (i.e. the number of change points). Details will follow ...

3 Some more testing

Here is a little for-loop that generates data using random parameters and checks whether *findSegments* can reconstruct them. The purpose of this is for checking the validity of the code.

```
> set.seed(4711)
> for (i in 1:20) {
+     gd = genData(lenx, nrcp)
+     seg = findSegments(gd$x, maxk = maxk, maxcp = maxcp)
+     stopifnot(seg$th[nrcp, 1:nrcp] == gd$cp)
+ }
```

```
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
```

```
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
```

```
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
Assessing arguments...
Computing cost matrix for segmentation...
Running Picard's segmentation algorithm...
```

References

[1] A statistical approach for CGH microarray data analysis. Franck Picard, Stephane Robin, Marc Lavielle, Christian Vaisse, Gilles Celeux, Jean-Jacques Daudin. Rapport de recherche No. 5139, Mars 2004, *Institut National de Recherche en Informatique et en Automatique (INRIA)*, ISSN 0249-6399. http://www.inapg.fr/ens_rech/mathinfo/recherche/mathematique/outil.html