

1 Supplementary Material

2 I. LIST OF ALL PARAMETERS

```
3 # name of the parameter set
4 name = "default",
5
6 # workspace
7 workspace.read = TRUE, # if [TRUE] it is tried to load the global data object from a workspace file in CM.ini()
8 workspace.write = FALSE, # if [TRUE] a workspace with the global data object will be written in CM.writeData()
9 workspace.replace = FALSE, # if [TRUE] a workspace will be replaced when existing in CM.writeData()
10 workspace.filename = "user_workspace.RData", # the filename used in CM.ini() and CM.writeData()
11
12 # input settings
13 input.dir = "input", # the directory from which all input files will be read in by CM.ini()
14 input.sep = "\t", # the column separator sign, e.g. ",", ";", "\t" (tab) passed to read.table (?read.table for more information)
15 input.col.easting = "POINT_X", # the column name for the x-value
16 input.col.northing = "POINT_Y", # s.a.
17 input.col.elevation = "POINT_Z", # s.a.
18 input.col.bank = "Name", # the column name of the side (left/right bank)
19 bank.code.left = "left", # the string code used for the left bank
20 bank.code.right = "right", # the string code used for the right bank
21
22 # output settings
23 output.write = FALSE, # if [TRUE] output ASCII files will be written
24 output.replace = FALSE, # if [TRUE] the output files are replaced when existing in CM.writeFiles()
25 output.write.centerline = FALSE, # if [TRUE] the geometry of the centerline will be written in CM.writeFiles()
26 output.write.metrics = TRUE, # if [TRUE] the calculated channel metrics will be written in CM.writeFiles()
27 output.write.metrics.d = TRUE, # switch on/off the variable d.r and d.l (distances from centerline to banks)
28 output.write.metrics.w = TRUE, # switch on/off the variable w (channel width)
29 output.write.metrics.r = TRUE, # switch on/off the variable r.r and r.l (direction factor of d.r and d.l)
30 output.write.metrics.diff = TRUE, # switch on/off the variable diff.r and diff.l (distances between two banks)
31
32 output.dir = "output",
33 output.sep = "\t",
34
35 # enable/disable plots
36 plot.polygoncheck = TRUE, # if [TRUE], a three-column plot is generated showing the entire river and both ends to roughly check the polygon consistency (see
37 also CM.generatePolygon())
38
39 plot.planview = TRUE, # create a plan view overview plot
40 plot.planview.secondary = TRUE, # in the plan view plot, add a secondary data set for comparison (will be displayed in dashed lines)
41 plot.planview.bankpoints = FALSE, # in the plan view plot, add the bank points of a data set
42 plot.planview.polygons = TRUE, # in the plan view plot, add the channel borders
43 plot.planview.voronoi = TRUE, # in the plan view plot, add voronoi polygons in plan view plot
44 plot.planview.cl.original = FALSE, # in the plan view plot, add the rough centerline (before smoothing)
45 plot.planview.cl.smoothed = TRUE, # in the plan view plot, add the smoothed centerline
46 plot.planview.cl.tx = FALSE, # in the plan view plot, add a label with the number next to the centerline points
47 plot.planview.transects = FALSE, # in the plan view plot, add transects (perpendiculars to centerline)
48 plot.planview.transects.len = 20, # give the length of transects in the unit of the input coordinates
49 plot.planview.dist2banks = TRUE, # in the plan view plot, add transect segments from centerline to the banks (left and right)
50 plot.planview.grid = TRUE, # in the plan view plot, add a grid in the background
```

```

51 plot.planview.grid.dist = 20, # the distance of the grid lines in the unit of the input coordinates
52 plot.planview.legend = TRUE, # in the plan view plot, add a legend
53 plot.planview.scalebar = TRUE, # in the plan view plot, add a scalebar (width of one plot.planview.grid.dist)
54
55 # plot options
56 plot.zoom = TRUE, # if [TRUE] the plan view plot is zoomed in (see also CM.plotPlanView())
57 plot.zoom.extent.length = 140, # zoom window extent for the plan view plot in the unit of the input coordinates
58 plot.zoom.extent = "e1", # applied zoom window name (see also CM.plotPlanView())
59 plot.zoom.extents = list( # presets (customizable list) of zoom windows
60   e1 = c(400480, 3103130),
61   e2 = c(399445, 3096220),
62   e3 = c(401623, 3105925),
63   all = NULL
64 ),
65 plot.cl.range = "c11", # applied zoom cl range (see also CM.plotPlanView)
66 plot.cl.ranges = list( # presets (customizable list) of cl ranges
67   c11 = c(1235, 1260)
68 ),
69 plot.cl.range.use.reference = TRUE, # determines whether to look for reference centerline [TRUE] or current centerline when centering around cl.range
70 plot.to.file = FALSE, # if [TRUE] all plots will be copied to file devices
71 plot.to.pdf = TRUE, # if [TRUE] the plot will be saved as pdf
72 plot.to.png = TRUE, # if [TRUE] the plot will be saved as png
73 plot.index = 0, # numbering for filenames (see also CM.plotPlanView())
74 plot.directory = "plots/", # directory for saving plots if plot.to.file = TRUE
75 plot.filename = "documentation", # plot file name
76
77 # model parameters
78 force.calc.voronoi = FALSE, # if [TRUE] the voronoi polygons are always re-calculated and never taken from cache
79 force.calc.cl = FALSE, # if [TRUE] the centerline is always re-calculated and never taken from cache
80 bank.interpolate = TRUE, # if [TRUE] the provided bank points are linearly interpolated to generate a denser polygon (see CM.generatePolygon())
81 bank.interpolate.max.dist = 6, # if bank.interpolate is [TRUE] this is the minimum distance all bank points will have
82 bank.filter3.max.it = 12, # number of the maximum iterations for filter 3 to prevent the program to run infinitely
83 centerline.smoothing.width = NULL, # smoothing window width in the unit of the input coordinates, if [NULL] (default) smoothing equals bank.interpolate.max.dist (see
84 CM.calculateCenterline())
85 transects.span = 3, # span of centerline points used for calculating the transects (see CM.processCenterline())
86 centerline.bin.length = 1, # for simplifying the centerline give the spacing in the unit of the input coordinates (see CM.reduceCenterline())
87 centerline.use.reference = FALSE, # sets method for calculating distance centerline to banks, if [FALSE] (default) each river profile will be compared to its own
88 centerline, if [TRUE] the centerline of centerline.reference will be taken (see CM.processCenterline())
89 centerline.reference = "set1", # sets the reference data set if centerline.use.reference is [TRUE]
90 calculate.metrics = TRUE, # if [TRUE] all centerline metrics are calculated (see CM.processCenterline())
91 force.calc.metrics = FALSE, # if [TRUE] the metrics are always re-calculated and never taken from cache
92 use.elevation = FALSE, # if a column "elevation" exists set this to TRUE, this allows the calculation of the channel gradient

```