

Supplement of Earth Surf. Dynam., 5, 557–570, 2017
<https://doi.org/10.5194/esurf-5-557-2017-supplement>
© Author(s) 2017. This work is distributed under
the Creative Commons Attribution 3.0 License.



Supplement of

Deriving principal channel metrics from bank and long-profile geometry with the R package cmgo

Antonius Golly and Jens M. Turowski

Correspondence to: Antonius Golly (antonius.golly@gmail.com)

The copyright of individual parts of the supplement might differ from the CC BY 3.0 License.

1 Supplementary Material

2 I. LIST OF ALL PARAMETERS

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

```

51 plot.planview.voronoi      = FALSE,      # in the plan view plot, add voronoi polygons in plan view plot
52 plot.planview.cl.original = FALSE,      # in the plan view plot, add the rough centerline (before smoothing)
53 plot.planview.cl.smoothed = TRUE,       # in the plan view plot, add the smoothed centerline
54 plot.planview.cl.points   = FALSE,      # when a centerline is plotted should the points representing the line be emphasized
55 plot.planview.cl.tx       = FALSE,      # in the plan view plot, add a label with the number next to the centerline points
56 plot.planview.cl.selection = TRUE,      # if [TRUE] and plot window is determined by cl points (see docu) the cl points are highlighted
57 plot.planview.transects   = FALSE,      # in the plan view plot, add transects (perpendiculars to centerline)
58 plot.planview.transects.len = 20,      # give the length of transects in the unit of the input coordinates
59 plot.planview.dist2banks  = TRUE,      # in the plan view plot, add transect segments from centerline to the banks (left and right)
60 plot.planview.grid        = TRUE,      # in the plan view plot, add a grid in the background
61 plot.planview.grid.dist   = 20,      # the distance of the grid lines in the unit of the input coordinates
62 plot.planview.legend      = TRUE,      # in the plan view plot, add a legend
63 plot.planview.legend.pos  = "topleft",  # keyword to position legend (see ?legend)
64 plot.planview.scalebar    = TRUE,      # in the plan view plot, add a scalebar (width of one plot.planview.grid.dist)
65 plot.planview.use.names   = TRUE,      # if [TRUE] set names will be used for display, otherwise "set1", "set2", etc.
66
67 plot.metrics.use.names    = TRUE,      # if [TRUE] set names will be used for display, otherwise "set1", "set2", etc.
68
69 # plot options
70 plot.zoom                 = TRUE,      # if [TRUE] the plan view plot is zoomed in (see also CM.plotPlanView())
71 plot.zoom.extent.length  = 140,      # zoom window extent for the plan view plot in the unit of the input coordinates
72 plot.zoom.extent         = "e1",     # applied zoom window name (see also CM.plotPlanView())
73 plot.zoom.extents        = list(     # presets (customizable list) of zoom windows
74     e1 = c(400480, 3103130),
75     e2 = c(399445, 3096220),
76     e3 = c(401623, 3105925)
77 ),
78 plot.cl.range            = "c11",     # applied zoom cl range (see also CM.plotPlanView)
79 plot.cl.ranges           = list(     # presets (customizable list) of cl ranges
80     c11 = c(1235, 1260)
81 ),
82 plot.cl.range.use.reference = TRUE,   # determines whether to look for reference centerline [TRUE] or current centerline when centering around cl.range
83 plot.to.file             = FALSE,     # if [TRUE] all plots will be copied to file devices
84 plot.to.pdf              = TRUE,     # if [TRUE] the plot will be saved as pdf
85 plot.to.png              = TRUE,     # if [TRUE] the plot will be saved as png
86 plot.index               = 0,        # numbering for filenames (see also CM.plotPlanView())
87 plot.directory           = "plots/",  # directory for saving plots if plot.to.file = TRUE
88 plot.filename            = "documentation", # plot file name
89
90 # model parameters
91 force.calc.voronoi       = FALSE,     # if [TRUE] the voronoi polygons are always re-calculated and never taken from cache
92 force.calc.cl            = FALSE,     # if [TRUE] the centerline is always re-calculated and never taken from cache
93 bank.interpolate         = TRUE,     # if [TRUE] the provided bank points are linearly interpolated to generate a denser polygon (see CM.generatePolygon())
94 bank.interpolate.max.dist = 6,       # if bank.interpolate is [TRUE] this is the maximum distance all bank points will have
95 bank.reduce              = FALSE,    # if [TRUE] the provided bank points are reduced by points that are closer to each other than bank.reduce.min.dist
96 bank.reduce.min.dist    = 0.5,      # if bank.reduce is [TRUE] this is the minimum distance all bank point will have
97 bank.filter2.max.it     = 12,       # number of the maximum iterations for filter 2 to prevent the program to run infinitely
98 centerline.smoothing.width = 7,     # smoothing window width of mean filter in number of observations (see CM.calculateCenterline())
99 centerline.local.slope.range = 15,
100 transects.span          = 3,        # span of centerline points used for calculating the transects (see CM.processCenterline())
101 centerline.bin.length   = 5,       # for simplifying the centerline give the spacing in the unit of the input coordinates (see CM.reduceCenterline())
102 centerline.use.reference = FALSE,    # sets method for calculating distance centerline to banks, if [FALSE] (default) each river profile will be compared to its own
103 centerline.reference    = "set1",   # sets the reference data set if centerline.use.reference is [TRUE]
104 calculate.metrics       = TRUE,     # if [TRUE] all centerline metrics are calculated (see CM.processCenterline())
105

```

```

106 force.calc.metrics      = FALSE,      # if [TRUE] the metrics are always re-calculated and never taken from cache
107
108 # step identification after Zimmermann et. al 2008 [Zimmermann, A.E., Church, M., and Hassan, M. a., 2008, Identification of steps and pools from stream longitudinal
109 profile data: Geomorphology, v. 102, no. 3-4, p. 395-406, doi: 10.1016/j.geomorph.2008.04.009.]
110 steps.identify          = TRUE,
111 steps.verbose           = FALSE,      # should there be
112 steps.thalweg.dist      = "3d",      # chose method of distance calculation "3d" or "2d"
113 steps.minimum.step.length = 2.25,    # as percentage of Wb [%]
114 steps.maximum.step.length = 200,     # as percentage of Wb [%]
115 steps.minimum.pool.length = 10,      # as percentage of Wb [%]
116 steps.minimum.residual.depth= 0.23,  # as percentage of Wb [%]
117 steps.minimum.drop.height = 3.3,     # as percentage of Wb [%]
118 steps.minimum.step.slope = 10,       # average slope + 10 degree [°]
119 steps.bank.full.width.fix = TRUE,    # TRUE: use a fix bank full width for the whole stream, FALSE: calculate from banks
120 steps.bank.full.width    = 3.7,     # [m]
121 steps.average.slope.fix  = FALSE,
122 steps.average.slope      = 12.5,    #8.34, #12.5,      # [°]
123
124 # ignore
125 dummy = TRUE
126
127 )
128

```