

```
1 [maftools.ui.sections]
2 order = ["readfiles", "getFields", "getSampleSummary", "plotmafSummary",
3 "oncoplot_default", "oncoplot_with_cnv", "oncoplot_advanced", "plotTiTv",
4 "lollipopPlot2", "tcgaCompare", "plotVaf",
5 "somaticInteractions", "oncostrip", "plotEnrichmentResults", "plotClusters"]
6 [maftools.ui.sections.ui_basic]
7 readfiles = "title = 'Input files', status = 'primary', width = 12, collapsed
8 = FALSE, collapsible = TRUE"
9 getFields = "title = 'Output of maftools fields summary', status = 'primary',
10 width = 12, collapsed = TRUE, collapsible = TRUE"
11 getSampleSummary = "title = 'Output of maftools sample summary', status =
12 'primary', width = 12, collapsed = TRUE, collapsible = TRUE"
13 getGeneSummary = "title = 'Output of maftools gene summary', status =
14 'primary', width = 12, collapsed = TRUE, collapsible = TRUE"
15 getClinicalData = "title = 'Output of maftools clinical data summary', status
16 = 'primary', width = 12, collapsed = TRUE, collapsible = TRUE"
17 plotmafSummary = "title = 'Output of maftools MAF summary', status =
18 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
19 oncoplot_default = "title = 'Output of maftools oncoplots', status =
20 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
21 oncoplot_with_cnv = "title = 'Output of maftools oncoplots with copy number
22 data', status = 'primary', width = 12, collapsed = TRUE, height='auto',
23 collapsible = TRUE"
24 oncoplot_advanced = "title = 'Output of maftools oncoplots with advanced',
25 status = 'primary', width = 12, collapsed = TRUE, height='auto', collapsible
26 = TRUE"
27 plotTiTv = "title = 'Output of maftools transition and transversions', status
28 = 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
29 lollipopPlot = "title = 'Output of maftools Lollipop plots for amino acid
30 changes', status = 'primary', width = 12, collapsed = TRUE, height='auto',
31 collapsible = TRUE"
32 lollipopPlot2 = "title = 'Output of maftools Lollipop2 plots for amino acid
33 changes', status = 'primary', width = 12, collapsed = TRUE, height='auto',
34 collapsible = TRUE"
35 tcgaCompare = "title = 'Output of maftools comparing mutation load', status =
36 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
37 plotVaf = "title = 'Output of maftools VAF boxplot', status = 'primary',
38 width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
39 gisticChromPlot = "title = 'Output of maftools gistic genome plot', status =
40 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
41 gisticBubblePlot = "title = 'Output of maftools gistic bubble plot', status =
42 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
43 gisticOncoPlot = "title = 'Output of maftools gistic oncoplot plot', status =
44 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
45 somaticInteractions = "title = 'Output of maftools somatic interactions',
46 status = 'primary', width = 12, collapsed = TRUE, height='auto', collapsible
47 = TRUE"
48 oncostrip = "title = 'Output of maftools somatic interactions (oncostrip)',
49 status = 'primary', width = 12, collapsed = TRUE, height='auto', collapsible
50 = TRUE"
51 plotOncodrive = "title = 'Output of maftools driver based on positional
52 clustering', status = 'primary', width = 12, collapsed = TRUE, height='auto',
53 collapsible = TRUE"
54 mafSurvival = "title = 'Output of maftools Survival', status = 'primary',
55 width = 12, collapsed = TRUE, collapsible = TRUE"
56 plotEnrichmentResults = "title = 'Output of maftools Clinical enrichment
57 analysis', status = 'primary', width = 12, collapsed = TRUE, height='auto',
58 collapsible = TRUE"
59 plotClusters = "title = 'Output of maftools heterogeneity analysis', status =
60 'primary', width = 12, collapsed = TRUE, height='auto', collapsible = TRUE"
```

```

29
30 [maftools.paramters.readfiles]
31 # For reading annovarR shiny APP maftools tool input files
32 section_type = "input"
33 rcmd_last = ""
34 laml = read.maf(maf = laml_maf, clinicalData = laml_clin)
35 laml.plus.gistic = read.maf(maf = laml_maf, gisticAllLesionsFile =
all_lesions, gisticAmpGenesFile = amp_genes,
36   gisticDelGenesFile = del_genes, gisticScoresFile = scores_gis, isTCGA =
is_tcga)
37 laml.gistic = readGistic(gisticAllLesionsFile = all_lesions,
gisticAmpGenesFile = amp_genes, gisticDelGenesFile = del_genes,
gisticScoresFile = scores_gis, isTCGA = is_tcga)
38 primary_maf = read.maf(maf = primary_maf)
39 relapse_maf = read.maf(maf = relapse_maf)
40 ""
41 progressbar_message = "Reading related MAF and other files."
42 render_id = "maftools_readfiles"
43
44 #!!!! input_ui_order required related section
45 input_ui_order = ["single_input", "cnv_input", "comparsion_input",
"other_params", "start_analysis"]
46
47 [maftools.paramters.readfiles.input.single_input]
48 title = "Single sample anlayais:"
49 title_control = "class = 'input-section-p'"
50 varname = ["laml_maf", "laml_clin", "laml_mutsig"]
51 input_id = ["input_maf", "input_clin", "input_mutsig"]
52 type = ["shiny::selectInput", "shiny::selectInput", "shiny::selectInput"]
53 label = ["MAF file (laml_maf)", "Clinical annotation file (laml_clin)",
"MutSig reusults (laml_mutsig)"]
54 [maftools.paramters.readfiles.input.single_input.choices]
55 laml_maf = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(maf)', full.names = TRUE), featch_files(c('maf',
'maf.gz'))$file_path)}"
56 laml_clin = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(txt)|(tsv)', full.names = TRUE), featch_files(c('tsv', 'txt',
'txt.gz', 'tsv.gz'))$file_path)}"
57 laml_mutsig = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(txt)|(tsv)', full.names = TRUE), featch_files(c('tsv', 'txt',
'txt.gz', 'tsv.gz'))$file_path)}"
58 [maftools.paramters.readfiles.input.single_input.selected]
59 laml_maf = "@>@system.file('extdata', 'tcga_laml.maf.gz', package =
'maftools')@<@"
60 laml_clin = "@>@system.file('extdata', 'tcga_laml_annot.tsv', package =
'maftools')@<@"
61 laml_mutsig = "@>@system.file('extdata', 'LAML_sig_genes.txt.gz', package =
'maftools')@<@"
62
63 [maftools.paramters.readfiles.input.cnv_input]
64 title = "CNV related analyais:"
65 title_control = "class = 'input-section-p'"
66 varname = ["all_lesions", "amp_genes", "del_genes", "scores_gis"]
67 input_id = ["input_gistic_all_lesions", "input_gistic_amp",
"input_gistic_del", "input_gistic_score"]
68 type = ["shiny::selectInput", "shiny::selectInput", "shiny::selectInput",
"shiny::selectInput"]
69 label = ["Gistic all lesions file (all_lesions):", "Gistic amp genes file
(amp_genes)",

```

```

70         "Gistic del genes file (del_genes)", "Gistic score file
(scores_gis)"]
71 [maftools.paramters.readfiles.input.cnv_input.choices]
72 all_lesions = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(txt)|(tsv)', full.names = TRUE), featch_files(c('txt',
'txt.gz'))$file_path)}"
73 amp_genes = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(txt)|(tsv)', full.names = TRUE), featch_files(c('txt',
'txt.gz'))$file_path)}"
74 del_genes = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(txt)|(tsv)', full.names = TRUE), featch_files(c('txt',
'txt.gz'))$file_path)}"
75 scores_gis = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(gistic)', full.names = TRUE), featch_files(c('txt', 'txt.gz',
'gistic'))$file_path)}"
76 [maftools.paramters.readfiles.input.cnv_input.selected]
77 all_lesions = "@>@system.file('extdata', 'all_lesions.conf_99.txt', package =
'maftools')@<@"
78 amp_genes = "@>@system.file('extdata', 'amp_genes.conf_99.txt', package =
'maftools')@<@"
79 del_genes = "@>@system.file('extdata', 'del_genes.conf_99.txt', package =
'maftools')@<@"
80 scores_gis = "@>@system.file('extdata', 'scores.gistic', package =
'maftools')@<@"
81
82 [maftools.paramters.readfiles.input.comparsion_input]
83 title = "Comparsion between different cohort:"
84 title_control = "class = 'input-section-p'"
85 varname = ["primary_maf", "relapse_maf"]
86 input_id = ["input_primary_maf", "input_relapse_maf"]
87 type = ["shiny::selectInput", "shiny::selectInput"]
88 label = ["Cohort 1 maf file (primary_maf)", "Cohort 2 maf file
(relapse_maf)"]
89 [maftools.paramters.readfiles.input.comparsion_input.choices]
90 primary_maf = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(maf)', full.names = TRUE), featch_files(c('maf',
'maf.gz'))$file_path)}"
91 relapse_maf = "!!glue {c(list.files(system.file('extdata', package =
'maftools'), '(maf)', full.names = TRUE), featch_files(c('maf',
'maf.gz'))$file_path)}"
92 [maftools.paramters.readfiles.input.comparsion_input.selected]
93 primary_maf = "@>@system.file('extdata', 'APL_primary.maf.gz', package =
'maftools')@<@"
94 relapse_maf = "@>@system.file('extdata', 'APL_relapse.maf.gz', package =
'maftools')@<@"
95
96 [maftools.paramters.readfiles.input.other_params]
97 title = "Other paramters"
98 title_control = "class = 'input-section-p'"
99 varname = ["is_tcga"]
100 input_id = ["maftools_is_tcga"]
101 type = ["shiny::checkboxInput"]
102 label = ["Reading file: Maf file is TCGA format? (is_tcga is TRUE)",
103 "Command to read files."]
104
105 [maftools.paramters.readfiles.input.other_params.value]
106 is_tcga = true
107
108 [maftools.paramters.readfiles.input.start_analysis]
109 input_id = "start_maftools_analysis"

```

```
110 type = "shiny::actionButton"
111 label = "Run"
112
113 [maftools.paramters.getFields]
114 section_type = "output"
115 rcmd_last = "getFields(laml)"
116 render_type = "shiny::renderPrint"
117 render_id = "maftools_fields_summary"
118 output_type = "shiny::verbatimTextOutput"
119 progressbar_message = "Maftools getFields"
120
121 [maftools.paramters.getSampleSummary]
122 section_type = "output"
123 rcmd_last = "getSampleSummary(laml)"
124 render_type = "DT::renderDataTable"
125 render_id = "maftools_sample_summary"
126 output_type = "DT::dataTableOutput"
127 progressbar_message = "Maftools getSampleSummary"
128
129 [maftools.paramters.getGeneSummary]
130 section_type = "output"
131 rcmd_last = "getGeneSummary(laml)"
132 render_type = "DT::renderDataTable"
133 render_id = "maftools_gene_summary"
134 output_type = "DT::dataTableOutput"
135 progressbar_message = "Maftools getGeneSummary"
136
137 [maftools.paramters.getClinicalData]
138 section_type = "output"
139 render_type = "DT::renderDataTable"
140 render_id = "maftools_clinical_data"
141 output_type = "DT::dataTableOutput"
142 rcmd_last = "getClinicalData(laml)"
143 progressbar_message = "Maftools getClinicalData"
144
145 [maftools.paramters.plotmafSummary]
146 section_type = "output"
147 render_type = "shiny::renderPlot"
148 render_id = "maftools_plot_maf_summary"
149 output_type = "shiny::plotOutput"
150 export_engine = "Cairo"
151 export_params = "type = 'pdf', width = 21, height = 14, units='cm',
bg='transparent'"
152 rcmd_last = ""plotmafSummary(maf = laml, rmOutlier = TRUE, addStat =
'median', dashboard = TRUE, titvRaw = FALSE)""
153 progressbar_message = "Maftools plotmafSummary"
154
155 [maftools.paramters.oncoplot_default]
156 section_type = "output"
157 render_type = "shiny::renderPlot"
158 render_id = "maftools_plot_oncoplots"
159 output_type = "shiny::plotOutput"
160 export_engine = "Cairo"
161 export_params = "type = 'pdf', width = 21, height = 14, units='cm',
bg='transparent'"
162 rcmd_last = "oncoplot(maf = laml, top = 10, fontSize = 12)"
163 progressbar_message = "Maftools oncoplot_default"
164
165 [maftools.paramters.oncoplot_with_cnv]
166 section_type = "output"
```

```

167 render_type = "shiny::renderPlot"
168 render_id = "maftools_plot_oncoplots_cnv"
169 output_type = "shiny::plotOutput"
170 export_engine = "Cairo"
171 export_params = "type = 'pdf', width = 21, height = 14, units='cm',
  bg='transparent'"
172 rcmd_last = "oncoplot(maf = laml.plus.gistic, top = 10, fontSize = 12)"
173 progressbar_message = "Maftools oncoplot with CNV"
174
175 [maftools.paramters.oncoplot_advanced]
176 section_type = "output"
177 render_type = "shiny::renderPlot"
178 render_id = "maftools_plot_oncoplots_advanced"
179 output_type = "shiny::plotOutput"
180 export_engine = "Cairo"
181 export_params = "type = 'pdf', width = 21, height = 27, units='cm',
  bg='transparent'"
182 rcmd_preprocess = ""
183 col = RColorBrewer::brewer.pal(n = 8, name = 'Paired')
184 names(col) = c('Frame_Shift_Del', 'Missense_Mutation', 'Nonsense_Mutation',
  'Multi_Hit', 'Frame_Shift_Ins',
185   'In_Frame_Ins', 'Splice_Site', 'In_Frame_Del')
186
187 #Color coding for FAB classification; try getAnnotations(x = laml) to see
  available annotations.
188 fabcolors = RColorBrewer::brewer.pal(n = 8, name = 'Spectral')
189 names(fabcolors) = c("M0", "M1", "M2", "M3", "M4", "M5", "M6", "M7")
190 fabcolors = list(FAB_classification = fabcolors)
191 mutsigQval = 0.01
192 clinicalFeatures = "FAB_classification"
193 sortByAnnotation = TRUE
194
195 oncoplot_advanced_params <- list(maf = laml, colors = col, mutsig =
  laml_mutsig,
196   mutsigQval = mutsigQval, clinicalFeatures = clinicalFeatures,
197   sortByAnnotation = sortByAnnotation,
198   annotationColor = fabcolors)
199 ""
200 rcmd_last = ""
201 do.call(oncoplot, oncoplot_advanced_params)
202 ""
203 progressbar_message = "Maftools oncoplot with advanced"
204
205 [maftools.paramters.plotTiTv]
206 section_type = "output"
207 render_type = "shiny::renderPlot"
208 render_id = "maftools_plot_titv"
209 output_type = "shiny::plotOutput"
210 export_engine = "Cairo"
211 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
  bg='transparent'"
212 rcmd_preprocess = "laml.titv = titv(maf = laml, plot = FALSE, useSyn = TRUE)"
213 rcmd_last = "plotTiTv(res = laml.titv)"
214 progressbar_message = "Maftools plotTiTv"
215
216 [maftools.paramters.lollipopPlot]
217 section_type = "output"
218 render_type = "shiny::renderPlot"
219 render_id = "maftools_plot_lollipop"
220 output_type = "shiny::plotOutput"

```

```

221 export_engine = "Cairo"
222 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
223 rcmd_last = ""lollipopPlot(maf = lam1, gene = 'DNMT3A', AACol =
'Protein_Change', showMutationRate = TRUE)""
224 progressbar_message = "Maftools lollipopPlot"
225 [maftools.paramters.lollipopPlot2]
226 section_type = "output"
227 render_type = "shiny::renderPlot"
228 render_id = "maftools_plot_lollipop2"
229 output_type = "shiny::plotOutput"
230 export_engine = "Cairo"
231 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
232 rcmd_last = ""lollipopPlot2(m1 = primary_maf, m2 = relapse_maf, gene =
"PML",
233 AACol1 = "amino_acid_change", AACol2 =
"amino_acid_change", m1_name = "Primary", m2_name = "Relapse")""
234 progressbar_message = "Maftools lollipopPlot2"
235
236 [maftools.paramters.tcgaCompare]
237 section_type = "output"
238 render_type = "shiny::renderPlot"
239 render_id = "maftools_plot_mutation_load"
240 output_type = "shiny::plotOutput"
241 export_engine = "Cairo"
242 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
243 rcmd_last = "lam1.mutload = tcgaCompare(maf = lam1, cohortName = 'Example-
LAML')"
244 progressbar_message = "Maftools tcgaCompare"
245
246 [maftools.paramters.plotVaf]
247 section_type = "output"
248 render_type = "shiny::renderPlot"
249 render_id = "maftools_plot_vaf_box"
250 output_type = "shiny::plotOutput"
251 export_engine = "Cairo"
252 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
253 rcmd_last = "plotVaf(maf = lam1, vafCol = 'i_TumorVAF_WU')"
254 progressbar_message = "Maftools plotVaf"
255
256 [maftools.paramters.gisticChromPlot]
257 section_type = "output"
258 render_type = "shiny::renderPlot"
259 render_id = "maftools_plot_gistic_genome"
260 output_type = "shiny::plotOutput"
261 export_engine = "Cairo"
262 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
263 rcmd_last = ""gisticChromPlot(gistic = lam1.gistic, markBands = "all")""
264 progressbar_message = "Maftools gisticChromPlot"
265
266 [maftools.paramters.gisticBubblePlot]
267 section_type = "output"
268 render_type = "shiny::renderPlot"
269 render_id = "maftools_plot_gistic_bubble"
270 output_type = "shiny::plotOutput"
271 export_engine = "Cairo"

```

```

272 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
273 rcmd_last = """"gisticBubblePlot(gistic = laml.gistic)""""
274 progressbar_message = "Maftools gisticBubblePlot"
275
276 [maftools.paramters.gisticOncoPlot]
277 section_type = "output"
278 render_type = "shiny::renderPlot"
279 render_id = "maftools_plot_gistic_oncoplot"
280 output_type = "shiny::plotOutput"
281 export_engine = "Cairo"
282 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
283 rcmd_last = """"gisticOncoPlot(gistic = laml.gistic, clinicalData =
getClinicalData(x = laml),
284 clinicalFeatures = 'FAB_classification', sortByAnnotation = TRUE, top =
10)""""
285 progressbar_message = "Maftools gisticOncoPlot"
286
287 [maftools.paramters.somaticInteractions]
288 section_type = "output"
289 render_type = "shiny::renderPlot"
290 render_id = "maftools_plot_somatic_inter"
291 output_type = "shiny::plotOutput"
292 export_engine = "Cairo"
293 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
294 rcmd_last = """"somaticInteractions(maf = laml, top = 25, pvalue = c(0.05,
0.1))""""
295 progressbar_message = "Maftools somaticInteractions"
296
297 [maftools.paramters.oncostrip]
298 section_type = "output"
299 render_type = "shiny::renderPlot"
300 render_id = "maftools_plot_somatic_inter_oncostrip"
301 output_type = "shiny::plotOutput"
302 export_engine = "Cairo"
303 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
304 rcmd_last = """"oncostrip(maf = laml, genes = c('TP53', 'FLT3', 'RUNX1'))""""
305 progressbar_message = "Maftools oncostrip"
306
307 [maftools.paramters.plotOncodrive]
308 section_type = "output"
309 render_type = "shiny::renderPlot"
310 render_id = "maftools_plot_oncodrive"
311 output_type = "shiny::plotOutput"
312 export_engine = "Cairo"
313 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
314 rcmd_preprocess = """"laml.sig = oncodrive(maf = laml, AACol =
'Protein_Change', minMut = 5, pvalMethod = 'zscore)""""
315 rcmd_last = """"plotOncodrive(res = laml.sig, fdrCutOff = 0.1, useFraction =
TRUE)""""
316 progressbar_message = "Maftools plotOncodrive"
317
318 [maftools.paramters.mafSurvival]
319 section_type = "output"
320 render_type = "shiny::renderPlot"
321 render_id = "maftools_plot_survival"

```

```
322 output_type = "shiny::plotOutput"
323 export_engine = "Cairo"
324 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
325 rcmd_last = ""mafSurvival(maf = lam1, genes = 'DNMT3A', time =
'days_to_last_followup',
326 Status = 'Overall_Survival_Status', isTCGA = TRUE)""
327 progressbar_message = "Maftools mafSurvival"
328
329 [maftools.paramters.plotEnrichmentResults]
330 section_type = "output"
331 render_type = "shiny::renderPlot"
332 render_id = "maftools_plot_clinical_enrichment"
333 output_type = "shiny::plotOutput"
334 export_engine = "Cairo"
335 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
336 rcmd_preprocess = "fab.ce = clinicalEnrichment(maf = lam1, clinicalFeature =
'FAB_classification')"
337 rcmd_last = ""plotEnrichmentResults(enrich_res = fab.ce, pVal = 0.05)""
338 progressbar_message = "Maftools plotEnrichmentResults"
339
340 [maftools.paramters.plotClusters]
341 section_type = "output"
342 render_type = "shiny::renderPlot"
343 render_id = "maftools_plot_clusters"
344 output_type = "shiny::plotOutput"
345 export_engine = "Cairo"
346 export_params = "type = 'pdf', width = 21, height = 17, units='cm',
bg='transparent'"
347 rcmd_preprocess = "tcga.ab.2972.het = inferHeterogeneity(maf = lam1, tsb =
'TCGA-AB-2972', vafCol = 'i_TumorVAF_WU')"
348 rcmd_last = "plotClusters(clusters = tcga.ab.2972.het)"
349 progressbar_message = "Maftools maftools_plot_clusters"
350
```