

Natascha Bergo



Oceanógrafa, aluna de Doutorado em Oc. Biológica no **Laboratório de Ecologia de Micro-organismos**, no **Instituto Oceanográfico** da USP. Trabalho com micro-organismos marinhos de mar profundo e sua relação com a formação de minerais marinhos de valor comercial.

[Currículo Lattes](#)

nataschabergo@usp.br

Meus Exercícios

[Exercícios](#)

Trabalho Final

[trabalhofinal](#)

==== Proposta A ====

Função **cito.analysis()**

A técnica de citometria de fluxo permite quantificar micro-organismos marinhos autotróficos e heterotróficos por meio da fluorescência natural de pigmentos acessórios ou pela adição de corantes, no caso as bactérias heterotróficas totais. Resultando em dados quantitativos muitas vezes não normais.

Atualmente, na Oceanografia Microbiana, existe uma grande demanda pela análise dos dados gerados pela técnica de citometria de fluxo. Com o intuito de evitar o uso de vários programas proponho a criação de uma função.

A função fará uma análise exploratória dos dados em busca de NA's, outliers, homogeneidade, normalidade, relações e interações dos dados. Caso sejam registrados muitos outliers, estes serão excluídos.

O próximo passo seria investigar os padrões e as relações dos dados. Comparar a distribuição dos micro-organismos, se os dados forem normais, o teste paramétrico de hipótese baseado na média e desvio padrão dos dados, One-way Anova seria empregado. Caso os dados não forem normais, os mesmos serão transformados em $\log(x+1)$ e seria aplicado um teste baseado na mediana e amplitude inter-quartil, o teste de Friedman.

Para finalizar, um teste multivariado seria aplicado com o intuito de avaliar a distribuição dos dados de abundância comparando-os dados físico-químicos, no caso a PCA.

Input: Matriz de dados contendo as informações abaixo:

- Abundância de *Synechococcus spp* - SYN;
- Abundância de *Prochlorococcus spp* - PRO;

- Abundância de Nanoplâncton eucarioto – NANOEUK;
- Abundância de Picoplâncton eucarioto – PICOEUK;
- Bactérias Heterotróficas – BACH;
- Dados físico-químicos: temperatura, pressão e salinidade.

Output:

Retornará os gráficos de análise exploratória, teste de comparação de distribuição dos dados (valor de p) e uma PCA.

Olá,

Gostei bastante da proposta. A função parece ser bem geral e acredito que você consiga fazer esta função sem muitos problemas, uma vez que utilizaria diversos elementos vistos em aula. Além disso você mostra conhecimento sobre o tema, o que sempre ajuda na hora de implementar a função.

Alguns pontos que podem melhorar:

* Nas variáveis de entrada é necessário colocar a abundância de *Synechococcus spp* e *Prochlorococcus spp* separadamente? Não seria interessante permitir ao usuário a possibilidade de colocar a abundância de espécies de outros gêneros?

* Faltou especificar, em “compute”, os objetos de entrada e saída. As abundâncias precisam ser em um vetor ou a função aceitaria outros tipos de dados (e.g., matriz). O teste de comparação será retornado em qual formato? Um valor de p seria suficiente? Aqui vale pensar em como estes dados são normalmente trabalhados (e.g., como estão organizados os dados da citometria de fluxo).

* Não entendi a relação entre o número de outliers e transformação dos dados [$\log(x+1)$].

—[Mauro Sugawara](#)

Olá Natascha, Você conseguiu apresentar bem o passo a passo da sua função. Porém toda a descrição parece mais um script para análise exploratória de um dado específico do que uma função. Recomendo você pensar em como tornar sua proposta mais geral: podendo o usuário colocar outras espécies e outras variáveis ambientais, por exemplo. Do contrário me parece que sua função servirá apenas para você. Sobre as transformações. Não entendi o porquê de fazer as transformações em si depois você vai fazer testes diferentes caso os dados sejam normais ou não. Eu deixaria a identificação de outliers como um passo diferente da transformação. E de repente um argumento: excluir outliers

da análise ou não. E uma das coisas que sua função poderia retornar é qual a posição dos outliers. Do jeito que está a função é muito específica. Você ainda precisa trabalhar para torná-la mais geral. — [Sara Mortara](#)

Resposta Natascha

Obrigada pelas correções e comentários, Mauro e Sara! Tentei mudar a Proposta "A" para os argumentos de vocês. Sara, sim é um dado específico, mas se for pensar em Oceanografia Microbiana, não seria tão específico assim. A entrada de grupos de organismos específicos é justificada pela saída do equipamento que só gera informações para esses organismos, por isso o nome da função de `cito.analysis()`.

==== Proposta B ====

Função: **ciano.correct()**

Synechococcus spp e *Prochlorococcus spp* são as cianobactérias mais abundantes dos oceanos e responsáveis por mais da metade da produção primária marinha. Ultimamente a técnica de citometria de fluxo tem sido empregada para quantificar estes micro-organismos e estimar a biomassa de carbono dos oceanos.

No entanto, a técnica de citometria de fluxo quantifica as células de *Synechococcus spp* e *Prochlorococcus spp* por meio da fluorescência natural dos pigmentos acessórios. Se as amostras forem oriundas de águas superficiais de oceanos oligotróficos, as células de cianobactérias terão baixa concentração de clorofila-a e consequentemente emitirão pouca fluorescência. O que resulta na sobreposição do ruído do equipamento sobre a contagem geral de células dificultando a real quantificação desses micro-organismos marinhos nas amostras. O que leva a exclusão destas amostras do estudo.

Para solucionar o problema de sobreposição dos dados gerados pelo citômetro de fluxo proponho uma função que possa calcular este valor, por meio da estimativa do complemento do histograma de abundância, ou seja, a função completará a parte faltante dos dados do histograma para cada amostra.



Figura exemplo de sobreposição do número de eventos (células, preto) sobre o ruído (transparente).

Input: Matriz com dados de abundância de todas as amostras de SYN, sendo cada amostra com dados em forma de histograma. Matriz com dados de abundância de todas as amostras de PRO, sendo cada amostra com dados em forma de histograma.

Output:

- Matrix com dados corrigidos de abundância de SYN.
- Matrix com dados corrigidos de abundância de PRO.
- Histograma completo com dados de SYN.
- Histograma completo com dados de PRO.

Olá,

Não entendi muito bem esta proposta. Qual técnica exatamente você pretende utilizar? Você conseguiria explicar passo a passo (como a proposta A)? Os histogramas que você pretende utilizar para fazer esta sobreposição vieram da mesma amostra analisada diversas vezes? —[Mauro Sugawara](#)

Natascha, de novo, esta função funcionaria apenas para estas duas cianobactérias? Este caso não pareceu tão problemático quanto a sua proposta anterior, mas ainda assim pense nisto. Acho que você sabe o que fazer aqui mas faltou explicar melhor como é esta estimativa do complemento do histograma de abundância. Você precisa explicar no help como isto funciona e deixar mais claro quais informações a função retorna. Apenas o gráfico?

Suas duas propostas são muito específicas. Para as duas, faça o exercício de torná-las mais geral, mesmo dentro da sua questão específica. Você precisa especificar qual a estrutura dos objetos de entrada e o que eles devem conter. Tente pensar na estrutura mais geral possível para executar a tarefa que você propõe. Sugiro prosseguir com uma super atenção à generalização. — [Sara Mortara](#)

Resposta Natascha

Obrigada pelas correções e comentários, Mauro e Sara! Mauro, não pretendo usar uma técnica. Eu só quero completar o histograma de cada amostra. E assim, não elimina-lá dos meus dados devido a sobreposição do histograma da amostra com o do ruído. Realmente é bem difícil explicar. Por isso quero trabalhar com a proposta A. =D Sara, este caso é super problemático e mais complexo que a Proposta A, simplesmente pelo fato da perda das amostras que são sobrepostas pelo ruído, estas amostras não podem ser utilizadas. A Proposta B, tentaria solucionar este problema, validando amostras que normalmente são excluídas dos trabalhos.

Tive vários problemas com a função A. Acabei trabalhando com a proposta B. —[Natascha](#)

Help da Função `ciano_correct()`

ciano_correct

package:unknown

R Documentation

Correções de dados de citômetros de fluxo em Oceanografia Microbiana #

Description:

#Esta função determina em qual caso há necessidade de correção de dados de abundância de cianobactérias *Synechococcus* spp (SYN) e *Prochlorococcus* spp (PRO) oriundos de citômetros de fluxo Canto e C6 e retorna a abundância celular corrigida ("no correction", " cell in noise" ou "correction").

Usage:

```
ciano.correct=function(dados, citometro)
```

Arguments:

#dados: Entrada de dados

#Citometro: tipo de equipamento canto & C6

Details:

Essa função fornece correção de dados de abundância oriundos citômetros de fluxo com dificuldade

detecção de células com baixa concentração de clorofila intracelular, no caso das cianobactérias

Synechococcus spp (SYN) e *Prochlorococcus* spp (PRO).

Ela permite a correção dos dados complementando os histogramas com a parte faltante a partir da

parte completa dele. Ideal para dados dos citômetros de fluxo: FaxCanto e C6.

Os dados introduzidos na função devem ser arquivos .txt para cada cianobactérias (ex. Syn._c6.txt

ou Syn_canto.txt) contendo o número de frequência do histograma de cada amostra, sendo as frequências

nas linhas e as amostras nas colunas. A primeira coluna deve se chamar "channel" e conter 500 linhas, ou seja, o total de canais.

Importante que os arquivos a serem analisados sejam .txt.

Value:

Retorna um arquivos .txt contendo os valores de abundância com correções quando necessários e sem

correções para as cianobactérias SYN e PRO (ex. stats_Syn_C6) dos citômetros C6 e FaxCanto para cada amostra.

Author:

Natascha M Bergo

nataschabergo@usp.br

References:

MARIE, D.; BRUSSAARD, C.; PARTENSKY, F.; VAULOT, D.; WILEY, J. Flow cytometric analysis of phytoplankton, bacteria and viruses. Current protocols in cytometry, v. 11, p. 1-15, 1999.

John Wiley & Sons, Inc.

MARIE, D.; PIERRE, U.; CURIE, M.; SIMON, N.; VAULOT, D. Phytoplankton Cell Counting by Flow

Cytometry. p. 1-17, 2005.

~~objects to See Also as 'help', ~~~

Examples:

Arquivo .txt de Input

ciano.correct=function(dados, citometro)

ciano.correct

channel	sample135_C6_PRO_5m	sample136_C6_PRO_50m	sample137_C6_PRO_110m	sample138_C6_PRO_130m	sample139_C6_PRO_170m
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0	0	0
33	0	0	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0

44	0	0	0	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0
51	0	0	0	0	0
52	0	0	0	0	0
53	0	0	0	0	0
54	0	0	0	0	0
55	0	0	0	0	0
56	0	0	0	0	0
57	0	0	0	0	0
58	0	0	0	0	0
59	0	0	0	0	0
60	0	0	0	0	0
61	0	0	0	0	0
62	0	0	0	0	0
63	0	0	0	0	0
64	0	0	0	0	0
65	0	0	0	0	0
66	0	0	0	0	0
67	0	0	0	0	0
68	0	0	0	0	0
69	0	0	0	0	0
70	0	0	0	0	0
71	0	0	0	0	0
72	0	0	0	0	0
73	0	0	0	0	0
74	0	0	0	0	0
75	0	0	0	0	0
76	0	0	0	0	0
77	0	0	0	0	0
78	0	0	0	0	0
79	0	0	0	0	0
80	0	0	0	0	0
81	0	0	0	0	0
82	0	0	0	0	0
83	0	0	0	0	0
84	0	0	0	0	0
85	0	0	0	0	0
86	0	0	0	0	0
87	0	0	0	0	0
88	0	0	0	0	0
89	0	0	0	0	0
90	0	0	0	0	0
91	0	0	0	0	0
92	0	0	0	0	0
93	0	0	0	0	0
94	0	0	0	0	0

95	0	0	0	0	0
96	0	0	0	0	0
97	0	0	0	0	0
98	0	0	0	0	0
99	0	0	0	0	0
100	0	0	0	0	0
101	0	0	0	0	0
102	0	0	0	0	0
103	0	0	0	0	0
104	0	0	0	0	0
105	0	0	0	0	0
106	0	0	0	0	0
107	0	0	0	0	0
108	0	0	0	0	0
109	0	0	0	0	0
110	0	0	0	0	0
111	0	0	0	0	0
112	0	0	0	0	0
113	0	0	0	0	0
114	0	0	0	0	0
115	0	0	0	0	0
116	0	0	0	0	0
117	0	0	0	0	0
118	0	0	0	0	0
119	0	0	0	0	0
120	0	0	0	0	0
121	0	0	0	0	0
122	0	0	0	0	0
123	0	0	0	0	0
124	0	0	0	0	0
125	0	0	0	0	0
126	0	0	0	0	0
127	0	0	0	0	0
128	0	0	0	0	0
129	0	0	0	0	0
130	0	0	0	0	0
131	0	0	0	0	0
132	0	0	0	0	0
133	0	0	0	0	0
134	0	0	0	0	0
135	0	0	0	0	0
136	0	0	0	0	0
137	0	0	0	0	0
138	0	0	0	0	0
139	0	0	0	0	0
140	0	0	0	0	0
141	0	0	0	0	0
142	0	0	0	0	0
143	0	0	0	0	0
144	0	0	0	0	0

145	0	0	0	0	0
146	0	0	0	0	0
147	0	0	0	0	0
148	0	0	0	0	0
149	0	0	0	0	0
150	0	0	0	0	0
151	0	0	0	0	0
152	0	0	0	0	0
153	0	0	0	0	0
154	0	0	0	0	0
155	0	0	0	0	0
156	0	0	0	0	0
157	0	0	0	0	0
158	0	0	0	0	0
159	0	0	0	0	0
160	0	0	0	0	0
161	0	0	0	0	0
162	0	0	0	0	0
163	0	0	0	0	0
164	0	0	0	0	0
165	0	0	0	0	0
166	0	0	0	0	0
167	0	0	0	0	0
168	0	0	0	0	0
169	0	0	0	0	0
170	0	0	0	0	0
171	0	0	0	0	0
172	0	0	0	0	0
173	0	0	0	0	0
174	0	0	0	0	0
175	0	0	0	0	0
176	0	0	0	0	0
177	0	0	0	0	0
178	0	0	0	0	0
179	0	0	0	0	0
180	0	0	0	0	0
181	0	0	0	0	0
182	0	0	0	0	0
183	0	0	0	0	0
184	0	0	0	0	0
185	0	0	0	0	0
186	0	0	0	0	0
187	0	0	0	0	0
188	0	0	0	0	0
189	0	0	0	0	0
190	0	0	0	0	0
191	0	0	0	0	0
192	0	0	0	0	0
193	0	0	0	0	0
194	0	0	0	0	0
195	0	0	0	0	0

196	442	440	303	0	0
197	1104	1055	963	0	0
198	1647	1615	1519	9	0
199	1664	1604	1696	203	0
200	1385	1362	1440	545	0
201	1158	1123	1241	851	0
202	933	939	1048	882	0
203	748	760	898	716	0
204	590	591	745	543	0
205	479	474	661	421	0
206	376	362	566	322	0
207	286	283	529	239	0
208	203	198	494	184	0
209	147	149	488	135	0
210	103	102	478	100	0
211	80	77	463	71	0
212	56	56	475	53	0
213	43	41	486	36	0
214	35	30	491	22	0
215	24	21	489	16	0
216	16	12	492	13	1
217	10	7	488	12	6
218	9	5	501	12	9
219	7	5	491	9	11
220	6	6	509	7	10
221	3	5	507	6	10
222	2	4	511	7	9
223	2	3	491	7	8
224	3	3	472	5	8
225	2	3	450	5	9
226	2	2	435	5	8
227	1	1	408	8	9
228	0	0	384	9	8
229	0	0	357	11	9
230	1	0	343	10	7
231	2	1	330	11	8
232	2	1	302	11	8
233	1	1	271	14	8
234	1	0	237	16	6
235	1	1	224	18	6
236	1	0	205	17	6
237	0	0	190	20	7
238	1	0	180	22	7
239	1	0	175	31	7
240	1	0	166	34	7
241	0	0	160	35	8
242	0	0	156	43	11
243	0	0	157	48	14
244	0	0	149	58	16
245	0	0	140	59	20

246	0	0	143	66	22
247	0	0	148	71	24
248	0	0	148	84	25
249	0	0	142	91	33
250	0	0	134	94	40
251	0	0	136	101	46
252	0	0	131	112	48
253	0	0	124	126	54
254	0	0	112	130	56
255	0	0	107	126	67
256	0	0	107	127	68
257	0	0	106	127	77
258	0	0	97	125	75
259	0	0	85	121	74
260	0	0	68	114	75
261	0	0	56	120	73
262	0	0	51	117	79
263	0	0	48	123	75
264	0	0	46	108	73
265	0	0	39	107	71
266	0	0	34	93	68
267	0	0	27	89	64
268	0	0	19	75	61
269	0	0	13	73	60
270	0	0	10	73	59
271	0	0	11	65	55
272	0	0	10	56	48
273	0	0	8	43	43
274	0	0	6	39	36
275	0	0	5	35	36
276	0	0	3	27	32
277	0	0	3	26	28
278	0	0	3	18	22
279	0	0	3	17	18
280	0	0	3	13	20
281	0	0	2	11	18
282	0	0	3	9	18
283	0	0	3	6	13
284	0	0	4	5	10
285	0	0	3	5	7
286	0	0	2	5	9
287	0	0	2	5	10
288	0	0	1	3	12
289	0	0	2	2	8
290	0	0	1	2	6
291	0	0	1	3	4
292	0	0	1	3	4
293	0	0	0	2	3
294	0	0	0	1	2
295	0	0	0	1	1
296	0	0	0	1	2

297	0	0	0	1	2
298	0	0	0	0	1
299	0	0	0	0	2
300	0	0	0	0	3
301	0	0	0	0	3
302	0	0	0	0	2
303	0	0	0	0	1
304	0	0	0	0	1
305	0	0	0	0	2
306	0	0	0	0	1
307	0	0	0	0	0
308	0	0	0	0	0
309	0	0	0	0	0
310	0	0	0	0	0
311	0	0	0	0	0
312	0	0	0	0	0
313	0	0	0	0	0
314	0	0	0	0	0
315	0	0	0	0	0
316	0	0	0	0	0
317	0	0	0	0	0
318	0	0	0	0	0
319	0	0	0	0	0
320	0	0	0	0	0
321	0	0	0	0	0
322	0	0	0	0	0
323	0	0	0	0	0
324	0	0	0	0	0
325	0	0	0	0	0
326	0	0	0	0	0
327	0	0	0	0	0
328	0	0	0	0	0
329	0	0	0	0	0
330	0	0	0	0	0
331	0	0	0	0	0
332	0	0	0	0	0
333	0	0	0	0	0
334	0	0	0	0	0
335	0	0	0	0	0
336	0	0	0	0	0
337	0	0	0	0	0
338	0	0	0	0	0
339	0	0	0	0	0
340	0	0	0	0	0
341	0	0	0	0	0
342	0	0	0	0	0
343	0	0	0	0	0
344	0	0	0	0	0
345	0	0	0	0	0
346	0	0	0	0	0

347	0	0	0	0	0
348	0	0	0	0	0
349	0	0	0	0	0
350	0	0	0	0	0
351	0	0	0	0	0
352	0	0	0	0	0
353	0	0	0	0	0
354	0	0	0	0	0
355	0	0	0	0	0
356	0	0	0	0	0
357	0	0	0	0	0
358	0	0	0	0	0
359	0	0	0	0	0
360	0	0	0	0	0
361	0	0	0	0	0
362	0	0	0	0	0
363	0	0	0	0	0
364	0	0	0	0	0
365	0	0	0	0	0
366	0	0	0	0	0
367	0	0	0	0	0
368	0	0	0	0	0
369	0	0	0	0	0
370	0	0	0	0	0
371	0	0	0	0	0
372	0	0	0	0	0
373	0	0	0	0	0
374	0	0	0	0	0
375	0	0	0	0	0
376	0	0	0	0	0
377	0	0	0	0	0
378	0	0	0	0	0
379	0	0	0	0	0
380	0	0	0	0	0
381	0	0	0	0	0
382	0	0	0	0	0
383	0	0	0	0	0
384	0	0	0	0	0
385	0	0	0	0	0
386	0	0	0	0	0
387	0	0	0	0	0
388	0	0	0	0	0
389	0	0	0	0	0
390	0	0	0	0	0
391	0	0	0	0	0
392	0	0	0	0	0
393	0	0	0	0	0
394	0	0	0	0	0
395	0	0	0	0	0
396	0	0	0	0	0
397	0	0	0	0	0

398	0	0	0	0	0
399	0	0	0	0	0
400	0	0	0	0	0
401	0	0	0	0	0
402	0	0	0	0	0
403	0	0	0	0	0
404	0	0	0	0	0
405	0	0	0	0	0
406	0	0	0	0	0
407	0	0	0	0	0
408	0	0	0	0	0
409	0	0	0	0	0
410	0	0	0	0	0
411	0	0	0	0	0
412	0	0	0	0	0
413	0	0	0	0	0
414	0	0	0	0	0
415	0	0	0	0	0
416	0	0	0	0	0
417	0	0	0	0	0
418	0	0	0	0	0
419	0	0	0	0	0
420	0	0	0	0	0
421	0	0	0	0	0
422	0	0	0	0	0
423	0	0	0	0	0
424	0	0	0	0	0
425	0	0	0	0	0
426	0	0	0	0	0
427	0	0	0	0	0
428	0	0	0	0	0
429	0	0	0	0	0
430	0	0	0	0	0
431	0	0	0	0	0
432	0	0	0	0	0
433	0	0	0	0	0
434	0	0	0	0	0
435	0	0	0	0	0
436	0	0	0	0	0
437	0	0	0	0	0
438	0	0	0	0	0
439	0	0	0	0	0
440	0	0	0	0	0
441	0	0	0	0	0
442	0	0	0	0	0
443	0	0	0	0	0
444	0	0	0	0	0
445	0	0	0	0	0
446	0	0	0	0	0
447	0	0	0	0	0

448	0	0	0	0	0
449	0	0	0	0	0
450	0	0	0	0	0
451	0	0	0	0	0
452	0	0	0	0	0
453	0	0	0	0	0
454	0	0	0	0	0
455	0	0	0	0	0
456	0	0	0	0	0
457	0	0	0	0	0
458	0	0	0	0	0
459	0	0	0	0	0
460	0	0	0	0	0
461	0	0	0	0	0
462	0	0	0	0	0
463	0	0	0	0	0
464	0	0	0	0	0
465	0	0	0	0	0
466	0	0	0	0	0
467	0	0	0	0	0
468	0	0	0	0	0
469	0	0	0	0	0
470	0	0	0	0	0
471	0	0	0	0	0
472	0	0	0	0	0
473	0	0	0	0	0
474	0	0	0	0	0
475	0	0	0	0	0
476	0	0	0	0	0
477	0	0	0	0	0
478	0	0	0	0	0
479	0	0	0	0	0
480	0	0	0	0	0
481	0	0	0	0	0
482	0	0	0	0	0
483	0	0	0	0	0
484	0	0	0	0	0
485	0	0	0	0	0
486	0	0	0	0	0
487	0	0	0	0	0
488	0	0	0	0	0
489	0	0	0	0	0
490	0	0	0	0	0
491	0	0	0	0	0
492	0	0	0	0	0
493	0	0	0	0	0
494	0	0	0	0	0
495	0	0	0	0	0
496	0	0	0	0	0
497	0	0	0	0	0
498	0	0	0	0	0

```
499 0      0      0      0      0
500 0      0      0      0      0
```

Arquivo .txt de Output

	sample	cell_tot	cell_tot_correc
1	sample135_C6_PRO_5m	134	cells in noise
2	sample136_C6_PRO_50m	111	cells in noise
3	sample137_C6_PRO_110m	13072	20240
4	sample138_C6_PRO_130m	3598	no correction
5	sample139_C6_PRO_170m	2211	no correction

Código da Função

```
#####
##### Função: ciano.correct() #####
### Correção de dados de citometros de fluxo em Oceanografia Microbiana ###
#####
#
ciano.correct=function(dados, citometro)
# Cria a função
#
{
  if ("ggplot2" %in% rownames(installed.packages())==FALSE) {
    stop("A função requer os pacotes 'ggplot2', 'reshape2', 'plyr' e
'scales'")
  } else {
    library("ggplot2") }
  if ("reshape2" %in% rownames(installed.packages())==FALSE) {
    stop("A função requer os pacotes 'ggplot2', 'reshape2', 'plyr' e
'scales'")
  } else {
    library("reshape2") }
  if ("plyr" %in% rownames(installed.packages())==FALSE) {
    stop("A função requer os pacotes 'ggplot2', 'reshape2', 'plyr' e
'scales'")
  } else {
    library("plyr") }
  if ("scales" %in% rownames(installed.packages())==FALSE) {
    stop("A função requer os pacotes 'ggplot2', 'reshape2', 'plyr' e
'scales'")
  } else {
    library("scales") }
# Verifica se os pacotes necessários para o cálculo estão instalados. Caso
# não estejam, a função para e informa que os mesmos são necessários. Caso
# estejam, carrega-os.
```

```
#
  if (!is.data.frame(dados) && nrow(dados) > 500)
    stop("'dados' deve ser um data.frame com até 500 linhas, que é o
número máximo de canais dos citômetros.")
# Define o valor e o formato da entrada de dados

  if (citometro != "canto" && citometro != "c6") {
    stop("Essa função suporta apenas os citômetros 'canto' e 'c6'.")
  } else if (citometro == 'canto') {
    decades = 5
    channel_min = 100
    channel_max = 500
    xmin = 10
    xmax = 10000
  } else if (citometro == "c6") {
    decades = 7
    channel_min = 214
    channel_max = 500
    xmin = 100
    xmax = 100000
  }
# Define qual citômetro será utilizado e quais os parâmetros de cada um.
#
  d1 = dados[dados$channel>=channel_min,]
  dm = melt(d1, id.vars=c("channel"), variable.name="sample",
value.name="cell_number")
  dm1 = ddply(dm, c("sample"), transform,
cell_smooth=as.vector(smooth(cell_number)))
  cor_factor=function(channel, cell_number, cell_smooth)
  {
    i_min = which.min(channel)
    i_max = which.max(channel)
    i_cell_max = which.max(cell_smooth)
    if (cell_smooth[i_cell_max] > 5*cell_smooth[i_min]) {
      cor_factor = "no correction"
    } else if (i_cell_max == i_min) {
      cor_factor = "cells in noise"
    } else {
      cor_factor = 2*sum(cell_number[i_cell_max:i_max]) }
    return(cor_factor)
  }
  dm2 = ddply(dm1, c("sample"), transform,
cell_norm=cell_smooth/max(cell_smooth))
  dm3 = ddply(dm2, c("sample"), transform,
fluo=(10^decades)^(channel/channel_max))
  stats = ddply(dm3, c("sample"), summarise, cell_tot=sum(cell_number),
cell_tot_correc=cor_factor(sample, cell_number, cell_smooth))
  return(stats)
}
```

Arquivos da Função:

Função: [ciano.correct.r](#)

Help: [help.r](#)

Input: [pro_c6.txt](#) [pro_canto.txt](#) [syn_c6.txt](#) [syn_canto.txt](#)

Output: [stats_pro_c6.txt](#) [stats_pro_canto.txt](#) [stats_syn_c6.txt](#) [stats_syn_canto.txt](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:nataschabergo:start 

Last update: **2020/08/12 06:04**