

```

1 % Clean
2 close all
3 clear all
4 clc
5
6 % All volumes are expressed as STP equivalents
7 % Prefixes and suffixes:
8 % f - filtered
9 % c - compensated
10 % d - diluent
11 % i - injected
12 % m - metabolized
13 % r - Rate [1/s]
14 % V - volume (STP) [L]
15 % n - quantity [mol]
16 % t - time [s]
17
18 % Define data directories
19 base_dir='..';
20 combined_dir=[base_dir '/' 'combined'];
21
22 % Load indexing data
23 timing=readtable([combined_dir '/' 'combined_timing.csv']);
24
25 % Load manual diluent addition info
26 DVd=readtable('DVd.csv');
27
28 % Constants
29 T0=273.15; % STP temperature (0 Celcius) [K]
30 P0=1.01325; % STP pressure (1 atm) [bar]
31 Twater=T0+35.5; % Water temp [K]
32 Tloop=37+T0; % Circuit temperature (Same as body; water was 35.5 C)
33 mspers=1000; % Time unit conversion factor [ms/s]
34 f02d=0.21; % Diluent oxygen fraction
35
36 % For filtering purposes
37 T5=(5*60)/log(2); % Fastest Thalmann time constant
38 Tf=round(2*T5/2); % Filter window width [s]. The factor 2 is for matching -3 dB cutoff with first
order filter of time constant T5/2
39
40 dives=size(timing,1);
41
42 % Pre-allocate arrays for saving
43 VN2f_array=cell(2,dives);
44 VN2f_offset_array=cell(2,dives);
45
46 for k=1:dives
47     % Load data
48     data=readtable([combined_dir '/' 'dive_' sprintf('%02d',k) '.csv']);
49     n=size(data,1); % Number of samples
50
51     % Bottom or deco
52     for stage=1:2
53         if stage==1
54             idx=timing.loadcell_bottom_start_idx(k):timing.loadcell_bottom_end_idx(k);
55         else
56             idx=timing.loadcell_deco_start_idx(k):timing.loadcell_deco_end_idx(k);
57         end
58
59         offset=0;
60         if stage==2
61             offset=VN2f_array(stage,end);
62         end
63
64         % Read time-aligned rebreather and loadcell log data
65         m=data.m(idx); % Loadcell mass change [kg]
66         P=data.Pemod(idx); % Ambient total pressure during bottom stage [bar]
67         pO2=data.PO2_1(idx); % Circuit oxygen partial pressure [bar]

```

```

68 irr=data.Inj_rate(idx); % Valve injection rate per open duration [L/s/s]
69 tO2i=data.MET_O2(idx)/mspers; % Oxygen valve opening durations [s]
70 HP02=data.HP_02(idx); % Oxygen cylinder pressure [bar]
71 t=data.t_posix(idx); % Time vector [s]
72
73 % Times used for metabolic rate estimation
74 t1=5*60;
75 t2=numel(t);
76
77 % Manual ADV compensation
78 Vd=0*m;
79 Vd(DVd.idx(DVd.dive==k & DVd.stage==stage))=...
80     DVd.V(DVd.dive==k & DVd.stage==stage);
81 Vd=cumsum(Vd);
82 VN2d=(1-fO2d)*Vd;
83 V02d=fO2d*Vd;
84
85 % Offset time vector to start at 0
86 t=t-t(1); % [s]
87
88 % Isotonic regression
89 HP02=-lsqisotonic(t,-HP02); % [bar]
90
91 % Loadcell mass change converted to displacement volume change
92 V=T0/Tloop*P./P0.*m; % [L]
93 [~,Vf]=filter_offset(V,Tf); % [L]
94
95 % Stabilized oxygen fraction
96 fO2=mean(pO2(t1:t2))/mean(P);
97
98 % Oxygen injection
99 V02ir=irr.*P/P0*Twater/Tloop;
100 V02i=cumsum(tO2i).*V02ir; % Oxygen injection volume [L]
101 V02i=V02i+V02d;
102
103 % Oxygen circuit volume change (to maintain pO2)
104 V02f=fO2*Vf;
105
106 % Average O2 Metabolism
107 Dt=t(t2)-t(t1); % Included duration [s]
108 V02c=V02f(t2)-V02f(t1); % Circuit volume compensation [L]
109 V02mr=(V02i(t2)-V02i(t1)+V02c)/Dt; % [L/s]
110 V02m=V02mr*t;
111
112 % Estimate and filtered version
113 VN2=V+V02i-V02m+VN2d;
114 [VN2,VN2f]=filter_offset(VN2,Tf);
115
116 % First-order model regression
117 [params,model]=FO_model(VN2,1);
118 VN2mod=model(params,t);
119
120 % Align stages
121 if stage==1
122     offset=VN2f(end);
123 else
124     VN2=VN2+offset;
125     VN2f=VN2f+offset;
126     VN2mod=VN2mod+offset;
127 end
128
129 % Store values for all-in-one plots
130 VN2f_array{stage,k}=VN2f;
131 if stage==2
132     VN2f_offset_array{stage,k}=VN2f;
133 end
134
135 % Extract optimized parameters

```

```

136     V0=params(1);
137     V=params(2);
138     T=params(3)/60*log(2); % Convert T to half-time [min]
139
140     % Store
141     data.HP02(idx)=HP02;
142     data.V(idx)=V;
143     data.VN2(idx)=VN2;
144     data.VN2f(idx)=VN2f;
145     data.VN2mod(idx)=VN2mod;
146 end
147
148 % Save individual dive
149 writetable(data,['dive_' sprintf('%02d',k) '.csv']);
150
151 % Store cumulative O2 injection
152 data.MET_O2_int=cumsum(data.MET_O2);
153
154 % Write to individual files
155 writetable(data,['dive_' sprintf('%02d',k) '.csv']);
156 end
157
158 % Collect data for plots with all data
159 Ms = cell(1,2);
160 for stage = 1:2
161     vs=VN2f_array(stage,:);
162     lens=cellfun(@numel,vs);
163     n=max(lens);
164     M=nan(n,dives);
165     for d=1:dives
166         v=vs{d};
167         if ~isempty(v)
168             v=v(:);
169             M(1:numel(v),d)=v;
170         end
171     end
172     Ms{stage}=M;
173 end
174
175 % Write table to CSV
176 vars='dive'+string(arrayfun(@(x) sprintf('%02d',x),1:dives,'UniformOutput',false)); % Variable
names
177 writetable(array2table(Ms{1},'VariableNames',vars),'VN2f_bottom_all.csv'); % Bottom
178 writetable(array2table(Ms{2},'VariableNames',vars),'VN2f_deco_all.csv'); % Deco
179 writetable(array2table(Ms{2},'VariableNames',vars),'VN2f_deco_offset_all.csv'); % Deco offset

```