Supplemental Digital Content

Table A. Equations used to estimate oxygen desaturation

Equation	Term Definitions
Functional Residual Capacity (FRC)[1]	
Normal Body Habitus‡	
FRC = (5.48*Height in cm) – 7.05	
Obese Habitus‡	
FRC = 231.9 * EXP(-0.07 * BMI) + 55.2	
Exponential Oxygen Wash-in[2]	
$FAO_{2(n+1)} = FAO_{2(n)} + (FIO_2 - FAO_{2(n)})*(1 - EXP(-TV_{alv\_cumulative}/FRC))$	$FAO_{2(n+1)}$ and $FAO_{2(n)}$ are the alveolar fraction of oxygen in the current breath and the breath preceding pre-oxygenation, respectively FIO_2 = Fraction of inspired oxygen $TV_{alv\_cumulative}$ is the total alveolar tidal ventilation during pre-oxygenation up to the current breath
Alveolar Oxygen Partial Pressure	

$PAO_2 = FAO_2 * PB – PH2O$

PAO2 is Alveolar oxygen partial pressure (mmHg)

PB is barometric pressure (mmHg)

PH2O is water partial pressure (mmHg)

Arterial Oxygen Partial Pressure

PaO2 = PAO2 – (age in years) + 10)/4

PaO2 = Arterial oxygen partial pressure (mmHg)

Harris Benedict Equation[3]

BMR = 66.5 + (13.7 * weight in kg) + (5 * height in cm) – (6.8 * age in years)

BMR = Basal Metabolic Rate in calories per day

Weir Equation[4]

REE = [3.94*VO2 + 1.1 * VCO2] * 1.44

REE = Resting Energy Expenditure. (calories/day)

VO2 = oxygen consumption (L/min)

VCO2 = carbon dioxide output (mL/min)

Respiratory Quotient

R = VCO2/VO2

Assumed to be 0.8.

Estimated Oxygen Consumption[5]

VO2 = 1.21*(REE/7.02)

To solve for VO2:

1. REE assumed to be BMR.
2. VCO2 = R * VO2
3. Convert from STPD to BTPS

Kelman Equation[6]

$$SpO2 = 100 * (a1*x + a2*x^2 + a3*x^3 + x^4) / (a4 + a5*x + a6*x^2 + a7*x^3 + x^4)$$

SpO2 = Oxygen saturation (%)

x = PaO2

a1 = -8532

a2 = 2121

a3 = -67

a4 = 935961

a5 = -31346

a6 = 2396

a7 = -67

ǂ Equations are for male gender. Equations for female gender are not shown.

STPD = Standard Temperature Pressure Dry.

BTPS = Body Temperature Pressure Saturated (water vapor).

Table B. Assumptions and limitations of predicting oxygen saturation during apnea.

Assumptions
Pre oxygenation with a loosely sealed facemask on an anesthesia machine yields an inspired oxygen fraction (FiO_2) of 0.6
Spontaneous respiratory rate is 14 breaths per minute
Sea level is assumed for all calculations
For each breath, the tidal volume is 7 mL/kg, anatomic plus facemask dead space is 4.0 mL/kg, and alveolar tidal volume is 3.0 mL/kg
Lung volumes weight are normalized to ideal body weight and reported in BTPS
Cardiac output, hemoglobin concentration, arterial pH, arterial carbon dioxide, shunt and temperature are within normal limits
The airway is assumed to be closed during apnea
Exponential oxygen wash-in during pre-oxygenation[7]
Estimates of oxygen consumption assume the basal metabolic rate[5] resting energy expenditure, and respiratory quotient are all within normal limits[4]
The Harris Benedict formula estimates of basal metabolic rate are reasonable in morbidly obese individuals[3]
Basal metabolic rate is reduced by 30% during anesthesia
The functional residual capacity in obese individuals is reduced[8]
The alveolar – arterial oxygen gradient is increased by 5 mmHg in obese individuals[9]
Estimates of SpO2 can be made from arterial oxygen tension levels[6]
Limitations
Ventilatory response to elevated arterial carbon dioxide levels may be decreased in

#

obese individuals

Blood recirculation is not accounted for

Values for cardiac output and arterial carbon dioxide levels are likely to be abnormal in obese individuals

---

References

1. Ibanez J, Raurich JM. Normal values of functional residual capacity in the sitting and supine positions. Intensive Care Med 1982;8:173-7.
2. Fowler KT, Hugh-Jones P. Mass spectrometry applied to clinical practice and research. Br Med J 1957;1:1205-11.
3. Müller B, Merk S, Bürgi U, Diem P. Calculating the basal metabolic rate and severe and morbid obesity. Praxis (Bern 1994) 2001;90:1955-63.
4. Weir JB. New methods for calculating metabolic rate with special reference to protein metabolism. J Physiol 1949;109:1-9.
5. Harris JA, Benedict FG. A Biometric Study of Human Basal Metabolism. Proc Natl Acad Sci U S A 1918;4:370-3.
6. Kelman GR. Digital computer subroutine for the conversion of oxygen tension into saturation. J Appl Physiol 1966;21:1375-6.
7. Hardman JG, Wills JS, Aitkenhead AR. Factors determining the onset and course of hypoxemia during apnea: an investigation using physiological modelling. Anesth Analg 2000;90:619-24.
8. Jones RL, Nzekwu MM. The effects of body mass index on lung volumes. Chest 2006;130:827-33.

#

#

9. Taylor RR, Kelly TM, Elliott CG, Jensen RL, Jones SB. Hypoxemia after gastric bypass surgery for morbid obesity. Arch Surg 1985;120:1298-302.

#

We are providing a Metlab routine that was created for sugammadex. This routine was made possible after an exchange of emails with Dr. H. Kleijn, the first author of reference #12. Interested *A&A* readers may request access to the drug display created by Applied Medical Visualizations (Medvis) at (http://www.medvis.com/contacts/new).

Sugammadex code

```
clear all


% Covariates

BW = 75; % kg

AGE = 40.0; % yrs

CR = 100; % ml/min

SEX = 0; % m=0

asian_flag = 0;

sevo_flag = 0;


MW_roc = 609.7; % g/M

MW_sug = 2178; % g/M


t_start = 0; % mins

t_stop = 100; % mins

t_delta = 0.01; % mins
```

#

```
% Dosing info
inf.RocDose = [1.2];
inf.RocTime = [0];
inf.SugDose = [16];
inf.SugTime = [3];
inf.bolus_step = 0.2; % mins

[Pk, Pd] = PkPd_params('BW',BW,'AGE',AGE,'CR',CR,'race',asian_flag,'sevo',sevo_flag);

tspan = t_start:t_delta:t_stop;
Y = zeros(1,8);
T=tspan';
for loop = 2:length(tspan)
    t=tspan(loop);

    y = RK4(t,Y(end,:),t_delta,Pk,Pd,inf);

    Y = [Y; y];
end

CM_roc = Y(:,1)/Pk.Roc.V1; % uM/ml
CM_sug = Y(:,3)/Pk.Sug.V1; % uM/ml
CM_cplx = Y(:,5); % uM
```

#

#

CP_roc = (CM_roc + CM_cplx)*MW_roc; % ug/ml

CP_sug = ((CM_sug + CM_cplx)*MW_sug)/1000; % mg/ml

CP_roc_free = CM_roc * MW_roc; % ug/ml

CP_roc_cplx = CM_cplx * MW_roc; % ug

TOFI = Pd.Roc.E0 -

(Pd.Roc.Emax*(Y(:,7).^Pd.Roc.gamma))./(((Pd.Roc.EC50^Pd.Roc.gamma)+(Y(:,7).^Pd.Roc.ga

mma)));

T2I = Pd.Roc.E0T -

(Pd.Roc.EmaxT*(Y(:,8).^Pd.Roc.gammaT))./(((Pd.Roc.EC50T^Pd.Roc.gammaT)+(Y(:,8).^Pd.R

oc.gammaT)));

%% Output description

% Col A = Time (decimal minutes). Step size is 0.01 min (0.6 seconds)

% Col B = [Ce,Roc] (uM/ml. Multiply by 609.7 g/M to get ug/ml)

% Col C = TOF twitch ratio (T4/T1). (divide by 104 to get 0-1)

% Col D = [Cm,Roc] (uM/ml. Multiply by 609.7 g/M to get ug/ml)

% Col E = [Cm,Sug] (uM/ml. Multiply by 2178 g/M to get ug/ml)

% Col F = [Cm,Roc free + cmplx] (uM/ml. Multiply by 609.7 g/M to get ug/ml)

results = [T,Y(:,7),TOFI,CM_roc,CM_sug,CM_roc+CM_cplx];

Dose Code

function drugs = dose(t,Pk,inf)

MW_roc = 609.7; % g/M

#

```
#

MW_sug = 2178; % g/M

F1 = 1/MW_roc; % M/g

F3 = 1000/MW_sug; % M/mg


inf_tm = inf.bolus_step;


drugs.roc = 0;

drugs.sug = 0;


%% Rocuronium

for loop = 1:length(inf.RocTime)

    Rdose_tm = inf.RocTime(loop);

    % Rdose2_tm = 20;


    if t >= Rdose_tm && t <= Rdose_tm+inf_tm

        % Dose 1

        if length(inf.RocDose) == 1

            RocDose = inf.RocDose; % mg/kg

        else

            RocDose = inf.RocDose(loop);

        end

        TotRocDose = RocDose*Pk.BW*1000; % ug

        drugs.roc = F1*TotRocDose/inf_tm; % uM/min


#
```

```
#

        % dosing1 = pulse(F1*TotRocDose,0,2000);
    end

end

%% Sugammadex
for loop = 1:length(inf.SugTime)
    Sdose_tm = inf.SugTime(loop);

    if t >= Sdose_tm && t < Sdose_tm+inf_tm

        % Dose 1
        if length(inf.SugDose) == 1
            SugDose = inf.SugDose; % mg/kg
        else
            SugDose = inf.SugDose(loop);
        end
        TotSugDose = SugDose*Pk.BW; % mg
        drugs.sug = F3*TotSugDose/inf_tm; % uM/min

        % dosing3 = pulse(F3*TotSugDose,Sugtime,2000);
    end
end

#
```

#

PKPD Parameters code

```
function [Pk, Pd] = PkPd_params(varargin)

%% Compute the parameters necessary for computing pk/pd of roc & sugammadex

%         [Pk Pd] = PkPd_params('Name',value,...)

% If no inputs are provided, the population estimates will be returned

%       (Pk.BW = 70 kg, Pk.AGE = 43.5, Pk.CR = 119 ml/min)

%

% Input options are 'BW' (in kg), 'AGE' (in yrs), 'CR' (in ml/min), 'race'

% (binary for if asian) and 'sevo' (binary for if sevoflurance given).

% Default values are

%             BW = 70; % kg

%             AGE = 43.0; % yrs

%             CR = 119; % ml/min

%             race = 0;

%             sevo = 0;


% Values from H.J.Kleign et al, Population Pk-Pd analysis for sugammadex-mediated reversal

of

% rocuronium-induced neuromuscular blocakde. BJCP 2011. 72(3). 415-433


%% Default values

Pk.BW = 70; % kg

Pk.AGE = 43.0; % yrs
```

#

#

```
Pk.CR = 119; % ml/min
asian_flag = 0;
sevo_flag = 0;
M = 1; % for TOF gamma


%% Process inputs
n_inputs = nargin;


if mod(n_inputs,2) ~= 0
    error('myApp:argChk', 'Number of input arguments must be even')
else
    for loop = 2:2:n_inputs
        if strcmpi(varargin{loop-1},'BW')
            Pk.BW = varargin{loop};
        elseif strcmpi(varargin{loop-1},'AGE')
            Pk.AGE = varargin{loop};
        elseif strcmpi(varargin{loop-1},'CR')
            Pk.CR = varargin{loop};
        elseif strcmpi(varargin{loop-1},'race')
            asian_flag = logical(varargin{loop});
        elseif strcmpi(varargin{loop-1},'sevo')
            asian_flag = logical(varargin{loop});
        else
```

#

```
#

            error('myApp:inpChk', 'Input name %s not recognized. Valid names are "Pk.BW",
"Pk.AGE", "Pk.CR", "race" and "sevo".', varargin{loop-1})
        end
    end
end

%% Population Estimates of Pk parameters %%

% Rocuronium

Pk.Roc.CL = 0.269;                 % TH1, TCLR                         ml/min
Pk.Roc.CLage = -0.00678;           % TH13, CLRAGE1
Pk.Roc.IIV_CL = 32.4;
Pk.Roc.V1 = 4.73;                  % TH2, TV1R                         ml
Pk.Roc.V1cr = -0.00143;            % TH20, V2RCR1 ** SHOULD BE V1?
Pk.Roc.IIV_V1 = 23.8;
Pk.Roc.Q2 = 0.279;                 % TH3, TQ2R                         ml/min
if asian_flag == 0
    Pk.Roc.Q2rac = 1; % non_asian
else
    Pk.Roc.Q2rac = -0.212; % asian  % TH15, Q2RRAC1
end
Pk.Roc.V2 = 6.76;                  % TH4, TV2R                         ml

#
```

```
#

Pk.Roc.V2age = 0.00613;            % TH19, V2RRAGE1

Pk.Roc.IIV_V2 = 32.2;

Pk.Roc.RE = 20.0;


% Sugammadex


Pk.Sug.CL = 0.093;                 % TH5, TCLO                        ml/min

Pk.Sug.REN = 1.29;                 % TH11, CLclcr power function

Pk.Sug.CLbw = 0.00378;             % TH12, CLOBW1

Pk.Sug.IIV_CL = 22.4;

Pk.Sug.V1 = 4.42;                  % TH6, TV1O                        ml

Pk.Sug.V1bw = -0.00354;            % TH16, V1OBW1

if asian_flag == 0

   Pk.Sug.V1rac = 1; % non_asian

else

   Pk.Sug.V1rac = -0.16; % asian   % TH17, V1ORAC1

end

Pk.Sug.Q2 = 0.206;                 % TH7, TQ2O                        ml/min

Pk.Sug.V2 = 6.35;                  % TH8, TV2O                        ml

Pk.Sug.V2cr = -0.00305;            % TH18, V2OCR1

Pk.Sug.RE = 36.3;



#
```

```
#

% Complex


Pk.Cplx.kd = 0.0559;                % TH9, Kd                              uM

Pk.Cplx.loge_k2 = -3.38;            % TH10, K2                             1/min


Pk.Cplx.k2 = exp(Pk.Cplx.loge_k2);  %                                      1/min

Pk.Cplx.k1 = Pk.Cplx.k2/Pk.Cplx.kd; %                                      1/(min*uM)


%% Population Estimates of Pd parameters %%

% Parameters are for a 40 yr old, 75 kg, Pk.CReatine clearance 100 ml/min


% Rocuronimum

% T2

if sevo_flag == 0

   Pd.Roc.ke0T_sev = 1;

   Pd.Roc.EC50Tsev = 1;

else

   Pd.Roc.ke0T_sev = -0.567;       % TH27, Ke0SEV

   Pd.Roc.EC50Tsev = -0.395;       % TH25, EC50SEV

end

Pd.Roc.ke0T = 0.148;                % TH21, TKe0 T2                        1/min

Pd.Roc.EC50T = 1.61;                % TH22, TEC50 T2                       uM

Pd.Roc.gammaT = 6.83; % Hill coeff  % TH23, Tgam T2


#
```

```
#

Pd.Roc.E0T = 0.982;                % TH24, TE0 T2                    T4/T2


% TOF

if sevo_flag == 0

   Pd.Roc.ke0_sev = 1;

   Pd.Roc.EC50sev = 1;

else

   Pd.Roc.ke0_sev = -0.567;       % TH34, Ke0SEV

   Pd.Roc.EC50sev = -0.395;       % TH32, EC50SEV

end

Pd.Roc.ke0 = 0.134;                % TH28, TKe0 TOFratio                1/min

Pd.Roc.IIV_ke0 = 41.7;

Pd.Roc.EC50 = 1.62;                % TH29, TEC50 TOFratio               uM

Pd.Roc.IIV_EC50 = 24.9;

Pd.Roc.r = 0.37;

Pd.Roc.gamma = 7.52*M;             % TH30, Tgam TOFratio

Pd.Roc.IIV_Hill = 41.1;

Pd.Roc.E0 = 1.04;                  % TH31, TE0 TOFratio                 T4/T1

Pd.Roc.IIV_E0 = 11.1;

Pd.Roc.RE = 2.70;


% Sugammadex



#
```

```
#

Pd.Sug.ks = -3.43;                % TH35, Ks                    1/(min*uM)

Pd.Sug.IIV_ks = 1.14;

Pd.Sug.RE = 3.24;


%% Compute params

%################################################################################

if n_inputs > 0

   %% Population Estimates of Pk parameters %%

   % Parameters are for a 43 yr old, 70 kg, Pk.CReatine clearance 119 ml/min


   % Rocuronium


   if asian_flag == 0

      Pk.Roc.Q2rac = 1; % non_asian

   else

      Pk.Roc.Q2rac = 1 + Pk.Roc.Q2rac; % asian

   end


   Pk.Roc.CLage = 1 + Pk.Roc.CLage * (Pk.AGE - 43);

   Pk.Roc.CL = Pk.Roc.CLage * Pk.Roc.CL * (Pk.BW/70)^0.75; %          ml/min

   Pk.Roc.V1cr = exp(Pk.Roc.V1cr * (Pk.CR - 118.9));

   Pk.Roc.V1 = Pk.Roc.V1cr * Pk.Roc.V1 * (Pk.BW/70)^1; %              ml

   Pk.Roc.Q2 = Pk.Roc.Q2rac * Pk.Roc.Q2 * (Pk.BW/70)^0.75; %          ml/min


#
```

#

```
    Pk.Roc.V2age = exp(Pk.Roc.V2age * (Pk.AGE - 43.0));
    Pk.Roc.V2 = Pk.Roc.V2age * Pk.Roc.V2 * (Pk.BW/70)^1; %                  ml


    Pk.Roc.S1 = Pk.Roc.V1;
    Pk.Roc.k12 = Pk.Roc.Q2/Pk.Roc.V1; %                                  1/min
    Pk.Roc.k21 = Pk.Roc.Q2/Pk.Roc.V2; %                                  1/min


    % Sugammadex


    if asian_flag == 0
        Pk.Sug.V1rac = 1; % non_asian
    else
        Pk.Sug.V1rac = 1 + Pk.Sug.V1rac; % asian
    end
    Pk.Sug.REN = (2 * Pk.CR/(Pk.CR + 119))^Pk.Sug.REN;
    Pk.Sug.CLbw = 1 + Pk.Sug.CLbw * (Pk.BW - 74.5);
    Pk.Sug.CL = Pk.Sug.CLbw * (Pk.Sug.REN * Pk.Sug.CL); %                  ml/min
    Pk.Sug.V1bw = (1 + Pk.Sug.V1bw * (Pk.BW - 74.5))*Pk.Sug.V1rac;
    Pk.Sug.V1 = Pk.Sug.V1bw * Pk.Sug.V1rac * (Pk.Sug.V1 * (Pk.BW/70)^1); %  ml
    Pk.Sug.Q2 = Pk.Sug.Q2 * (Pk.BW/70)^0.75; %                           ml/min
    Pk.Sug.V2cr = exp(Pk.Sug.V2cr * (Pk.CR - 118.9));
    Pk.Sug.V2 = Pk.Sug.V2cr * (Pk.Sug.V2 * (Pk.BW/70)^1); %                ml
```

#

#

```
Pk.Sug.S3 = Pk.Sug.V1;

Pk.Sug.k12 = Pk.Sug.Q2/Pk.Sug.V1; %                    1/min

Pk.Sug.k21 = Pk.Sug.Q2/Pk.Sug.V2; %                    1/min


% Complex

Pk.Cplx.CL = Pk.Sug.CL; %                        ml/min

Pk.Cplx.V1 = Pk.Sug.V1; %                        ml

Pk.Cplx.Q2 = Pk.Sug.Q2; %                        ml/min

Pk.Cplx.V2 = Pk.Sug.V2; %                        ml


Pk.Cplx.k12 = Pk.Cplx.Q2/Pk.Cplx.V1; %                 1/min

Pk.Cplx.k21 = Pk.Cplx.Q2/Pk.Cplx.V2; %                 1/min


%% Population Estimates of Pd parameters %%

% Parameters are for a 40 yr 0ld, 75 kg, Pk.CReatine clearance 100 ml/min


% Rocuronimum

% T2

if sevo_flag == 0

   Pd.Roc.ke0T_sev = 1;

   Pd.Roc.EC50Tsev = 1;

else

   Pd.Roc.ke0T_sev = 1 + Pd.Roc.ke0T_sev;
```

#

#

```
    Pd.Roc.EC50Tsev = 1 + Pd.Roc.EC50Tsev;
  end
  Pd.Roc.ke0T = Pd.Roc.ke0T_sev * (Pd.Roc.ke0T * (Pk.BW/70)^-0.25); %     1/min
  Pd.Roc.EC50T = Pd.Roc.EC50Tsev * Pd.Roc.EC50T; %                       uM
  Pd.Roc.gammaT = Pd.Roc.gammaT; % Hill coefficient
  Pd.Roc.E0T = Pd.Roc.E0T * 100; %                                       T4/T2

  Pd.Roc.EmaxT = Pd.Roc.E0T;


  % TOF
  if sevo_flag == 0
    Pd.Roc.ke0_sev = 1;
    Pd.Roc.EC50sev = 1;
  else
    Pd.Roc.ke0_sev = 1 + Pd.Roc.ke0_sev;
    Pd.Roc.EC50sev = 1 + Pd.Roc.EC50sev;
  end

  Pd.Roc.ke0 = Pd.Roc.ke0_sev * (Pd.Roc.ke0 * (Pk.BW/70)^-0.25); %       1/min
  Pd.Roc.EC50 = Pd.Roc.EC50sev * Pd.Roc.EC50; %                          uM
  Pd.Roc.gamma = Pd.Roc.gamma; % Hill coefficient
  Pd.Roc.E0 = Pd.Roc.E0 * 100; %                                         T4/T1
```

#

#

```
    Pd.Roc.Emax = Pd.Roc.E0;

    % Sugammadex

    Pd.Sug.ks = exp(Pd.Sug.ks) * (Pk.BW/70)^-0.25; %                1/(min*uM)
End
```

RK4 Code

```
function y = RK4(t,yn,dt,Pk, Pd, inf)

% h = dt

k1 = dif_eqs(t,t-dt,yn,Pk,Pd,inf)*dt;
k2 = dif_eqs(t,t - 0.5*dt, yn + 0.5*k1,Pk,Pd,inf)*dt;
k3 = dif_eqs(t,t - 0.5*dt, yn + 0.5*k2,Pk,Pd,inf)*dt;
k4 = dif_eqs(t,t, yn + k3,Pk,Pd,inf)*dt;

y = yn + 1/6*(k1 + 2*k2 + 2*k3 + k4);
end

% Equations
function dy = dif_eqs(t,t_rk,y,Pk,Pd,inf)
CM_roc = y(1)/Pk.Roc.V1;
```

#

```
#

CM_sug = y(3)/Pk.Sug.V1;

CM_cplx = y(5);


drugs = dose(t,Pk,inf);


dy(1) = (-Pk.Roc.CL*CM_roc - Pk.Roc.k12*y(1) + Pk.Roc.k21*y(2)) + (-
CM_roc*CM_sug*Pk.Cplx.k1*Pk.Cplx.V1 + CM_cplx*Pk.Cplx.k2*Pk.Cplx.V1) + drugs.roc;
dy(2) = (Pk.Roc.k12*y(1) - Pk.Roc.k21*y(2));


dy(3) = (-Pk.Sug.CL*CM_sug - Pk.Sug.k12*y(3) + Pk.Sug.k21*y(4)) + (-
CM_roc*CM_sug*Pk.Cplx.k1*Pk.Cplx.V1 + CM_cplx*Pk.Cplx.k2*Pk.Cplx.V1) + drugs.sug;
dy(4) = (Pk.Sug.k12*y(3) - Pk.Sug.k21*y(4));


dy(5) = (CM_roc*CM_sug*Pk.Cplx.k1 - CM_cplx*Pk.Cplx.k2 -
(CM_cplx*Pk.Cplx.CL)/Pk.Cplx.V1) + (-Pk.Cplx.k12*y(5)+Pk.Cplx.k21*y(6));
dy(6) = (Pk.Cplx.k12*y(5) - Pk.Cplx.k21*y(6));


dy(7) = (Pd.Roc.ke0*CM_roc - Pd.Roc.ke0*y(7) - Pd.Sug.ks*y(7)*CM_sug);


dy(8) = (Pd.Roc.ke0T*CM_roc - Pd.Roc.ke0T*y(8) - Pd.Sug.ks*y(8)*CM_sug);
end




#
```