SUPPLEMENTARY MATERIAL

Charge Recombination Kinetics of Bacterial Photosynthetic Reaction Centres Reconstituted in Liposomes: Deterministic Versus Stochastic Approach.

Emiliano Altamura [1], Paola Albanese [1], Pasquale Stano [2], Massimo Trotta [3] Francesco Milano [4,*] and Fabio Mavelli [1,*]

[1] Università degli studi di Bari "Aldo Moro", Department of Chemistry, Via Orabona 4, 70125 Bari, Italy; emiliano.altamura@uniba.it (E.A.); paola.albanese@uniba.it (P.A.)
[2] University of Salento, Department of Biological and Environmental Sciences and Technologies (DiSTeBA), Strada Provinciale Lecce-Monteroni, Ecotekne, 73100 Lecce, Italy; pasquale.stano@unisalento.it
[3] IPCF-CNR Istituto per i Processi Chimico Fisici, via Orabona 4, 70125 Bari, Italy; massimo.trotta@cnr.it
[4] National Research Council of Italy (CNR), Institute of Science of Food Production (ISPA), Strada Provinciale Lecce-Monteroni, Ecotekne, 73100 Lecce, Italy
* Correspondence: francesco.milano@cnr.it (F.M.); fabio.mavelli@uniba.it (F.M.)

1. Matlab Code for the numerical integration of the ODE set.

```
function NumericalIntegration
   %-------------------------------------------------
   % Kinetic Constants
   r_kin = [2.45e+005 1.14e+002 1.0e+004 5.44e+002 2.45e+005 1.7e+002...
            9.7 9.7 2.180e-002];
   cRC = 1e-3;
   cQT = 0.5e-3;
   c_t = 0:0.01:2;
   %-------------------------------------------------
   % Get Matrix of Stoichiometric Coefficients
   [mR, mPP] = EM_GetRCMatrixCoeff(r_kin);
   %-------------------------------------------------
   % Calculate intial concentrations
   rY0 = EM_GetRCInitConc(r_kin(5), r_kin(6), cRC, cQT);
   %-------------------------------------------------
   % Integrate the ODE set
   [c_t,m_Cteo] = ode23s('odeKinSysEx',  c_t,rY0,[], r_kin, mR, mPP);
   %-------------------------------------------------
   % Calculate the
   c_Ydet = (m_Cteo(:,1) + m_Cteo(:,2) + m_Cteo(:,3)) / cRC;
   %-------------------------------------------------
   % Do plot
   figure(1)
   hp = plot(c_t, c_Ydet, 'r-');
   ylabel ('\chi*(\itt)')
   xlabel ('time \itt')
   set(hp, 'LineWidth', 2)
return

function [mR, mPP] = EM_GetRCMatrixCoeff(r_kin)
% restituisce le matrici dei coefficienti per reagenti e prodotti
% per il meccanismo seguente
%   Meccanismo                Valori default
% x* + q --> y*              k*(in)  = k(in)    = 2.5e+1   (?)
%    y*  --> x* + q          k*(out) = k(out)   = 2.5e+8   (?)
%    y*  --> z*              k(AB)              = 10000
```

```
%    z*  --> y*              k(BA)              = 650
%  x + q --> y               k(in)    = k*(in)  = 2.5e+1   (?)
%    y   --> x + q           k(out)   = k*(out) = 2.5e+8   (?)
%    x*  --> x               k1       = k(AD)   = 8.33
%    y*  --> y               k2       = k(AD)   = 8.33
%    z*  --> y               k3       = k(BD)   = 0.01

   %     x* y* Z* x  y  q
   mR = [1, 0, 0, 0, 0, 1
         0, 1, 0, 0, 0, 0
         0, 1, 0, 0, 0, 0
         0, 0, 1, 0, 0, 0
         0, 0, 0, 1, 0, 1
         0, 0, 0, 0, 1, 0
         1, 0, 0, 0, 0, 0
         0, 1, 0, 0, 0, 0
         0, 0, 1, 0, 0, 0]';

   mP = [0, 1, 0, 0, 0, 0
         1, 0, 0, 0, 0, 1
         0, 0, 1, 0, 0, 0
         0, 1, 0, 0, 0, 0
         0, 0, 0, 0, 1, 0
         0, 0, 0, 1, 0, 1
         0, 0, 0, 1, 0, 0
         0, 0, 0, 0, 1, 0
         0, 0, 0, 0, 1, 0]';
   mPP  =  mP - mR;
   for j=1:size(mPP,2)
      mPP(:,j) = r_kin(j)*mPP(:,j);
   end
return

function [cY0] = EM_GetRCInitConc(kin, kout, Crc, Cqa)
   D    = Crc - Cqa;
   KQ   = kin/kout;    % dark equilibrium constant
   b = (1/KQ + D);
   q0 = (-b + sqrt(b^2 + 4*(Cqa)/ KQ ) )/2;
   yy0 = (KQ*q0)/(1+KQ*q0)*Crc;
   xx0 = Crc - yy0;
   zz0 = 0;
   x0 = 0;
   y0 = 0;
   cY0 = [xx0; yy0; zz0; x0; y0; q0];
return


function dy = odeKinSysEx(t, y, rK, mR, mS, rT)
   dy = zeros(size(y));    % a column vector
   for jReaz =1:size(rK, 2)
      sV = prod(y .^ mR(:,jReaz)); % velocita reazione jReax
      if sV ~= 0
         dy = dy + mS(:,jReaz) * sV;
      end
   end
return
```