Another logo here

Calcium puff-like dynamics in Markov chain models of instantaneously coupled intracellular calcium channels

Gregory D. Smith

Department of Applied Science, College of William and Mary, Williamsburg, VA 23187

(greg@as.wm.edu)

Abstract

The stochastic activation and inactivation of intracellular calcium (Ca^{2+}) channels gives rise to localized Ca^{2+} elevations known as Ca^{2+} 'puffs' and 'sparks.' Beginning with minimal stochastic models of Ca^{2+} regulation of individual Ca^{2+} channels, we derive a formalism for simulation and analysis of the dynamics of Ca^{2+} regulation at "Ca^{2+} release sites," collections of 5-50 channels that give rise to puffs and sparks. The method involves selecting a continuous time Markov model for an intracellular Ca^{2+} channel of interest, choosing a spatial location for each channel (e.g., from a hypothetical distribution), and calculating the Ca^{2+} 'microdomain' associated with each possible configuation of the release site (i.e., which channels are open or closed). Because stochastic ion channel models have transition probabilities that depend on the local Ca^{2+} concentration, this procedure expands the Markov model for a single channel into a model of the release site as a collective entity. For tightly clustered Ca^{2+} release sites, the Ca^{2+} microdomain can be estimated in a computationally efficient manner as the steady-state solution to a system of reaction-diffusion equations representing the buffered diffusion of intracellular Ca^{2+} from a conselation of point sources. When biophysically reastic models of intracellular Ca^{2+} channels—e.g., the DeYoung-Keizer model of the inositol 1,4,5-trisphosphate receptor (IP_3R)—are used as the starting point, this method allows the analysis of equilibrium and non-equilibrium properties of IP_3-sensitive Ca^{2+} release sites. We find that the equilibrium open probability of IP_3-sensitive Ca^{2+} release sites can depend biphasically on the effective density of the release site, a dimensionless quantity that accounts for release site size, the number of channels per release site, and the length constant associated with Ca^{2+} buffers. The distinct Ca^{2+} regulatory properties of IP_3R subtypes are shown to influence the collective behavior of release sites composed of a homogenous population of IP_3R subtypes.

Q-matrix expansion for instantaneosly coupled channels

If we take as given the interaction matrix $C = (c_{ij})$ giving the increase in local Ca^{2+} experienced by channel j when channel i is open and continue to write the background Ca^{2+} concentration that is experienced by all channels even when all are closed as c_∞ (kept separate from C), then for two interacting two-state channels, C is 2×2 and the expanded Q-matrix becomes

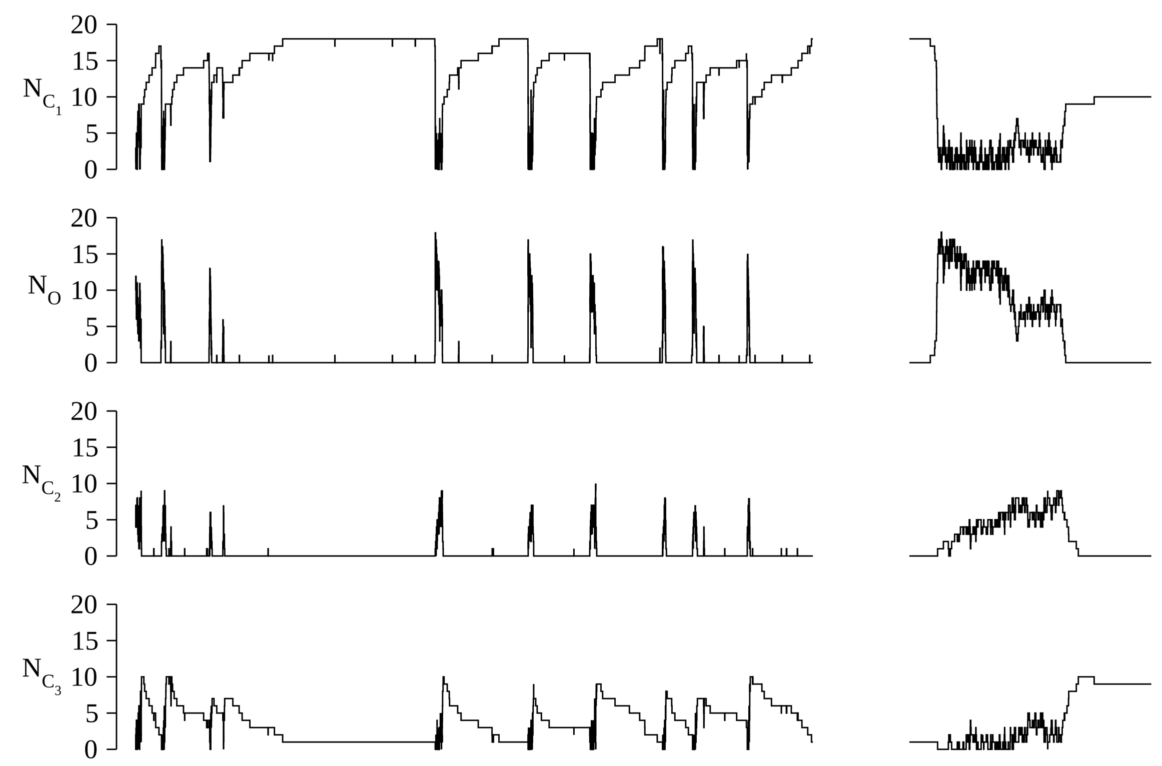
$$Q^{(2)} = \begin{pmatrix} \diamond & k^+c_\infty^\eta & k^+c_\infty^\eta & \cdot \\ k^- & \diamond & \cdot & k^+(c_\infty + c_{21})^\eta \\ k^- & \cdot & \diamond & k^+(c_\infty + c_{12})^\eta \\ \cdot & k^- & k^- & \diamond \end{pmatrix} \quad (1)$$

while for and for three interacting channels C is 3×3 and we have

$$Q^{(3)} = \begin{pmatrix} \diamond & k^+c_\infty^\eta & k^+c_\infty^\eta & \cdot & k^+c_\infty^\eta & \cdot & \cdot & \cdot \\ k^- & \diamond & \cdot & k^+(c_\infty + c_{12})^\eta & \cdot & k^+(c_\infty + c_{13})^\eta & \cdot & \cdot \\ k^- & \cdot & \diamond & k^+(c_\infty + c_{21})^\eta & \cdot & \cdot & k^+(c_\infty + c_{23})^\eta & \cdot \\ \cdot & k^- & k^- & \diamond & \cdot & \cdot & \cdot & k^+(c_\infty + c_{13} + c_{23})^\eta \\ k^- & \cdot & \cdot & \cdot & \diamond & k^+(c_\infty + c_{31})^\eta & k^+(c_\infty + c_{32})^\eta & \cdot \\ \cdot & k^- & \cdot & \cdot & k^- & \diamond & \cdot & k^+(c_\infty + c_{12} + c_{32})^\eta \\ \cdot & \cdot & k^- & \cdot & k^- & \cdot & \diamond & k^+(c_\infty + c_{13} + c_{23})^\eta \\ \cdot & \cdot & \cdot & k^- & \cdot & k^- & k^- & \diamond \end{pmatrix} \quad (2)$$

Notice how the Ca^{2+} microdomain complicates the expanded Q-matrix formalism.

Ca^{2+} release site simulations using 19 four-state models



Ca^{2+} release site simulations using 19 four-state models (Eq. **??** **??**) show repetitive cycles of activation and inactivation reminicent of Ca^{2+} puffs. Total time: 2 s (*left*) and 40 ms (*right*). Parameters used: $R = 0.5$ μm, $r_d = 0.05$ μm, $\eta = 2$, $c_\infty = 50$ nM; and $k_i^\pm$ as in Fig. **??**B.

Memory-efficient Kronecker operations

The stack operator maps an $n \times m$ matrix into an $nm \times 1$ vector. That is, if X is a $n \times m$ matrix comprising m column vectors $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m)$, where each $\mathbf{x}_i$ is an $n \times 1$ vector,

$$X = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \end{pmatrix}_{n \times m}$$

then the stack operator applied to X gives

$$X^S = \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{pmatrix}$$

A matrix-vector product of the form $(A \otimes B)\,\mathbf{x}$ can be efficiently calculated using the stack operation, it's inverse, and the identity,

Your logo here

$$(A \otimes B)\,\mathbf{x} = (A \otimes B)\,X^S = \left(BXA^T\right)^S \quad (3)$$

Reshaping $1 \times m_a m_b$ vector $\mathbf{x}$ to form a $m_b \times m_a$ matrix X and stacking the result of two matrix multiplications allows one to avoid peforming the Kronecker product and constructing the $n_a n_b \times m_a m_b$ matrix $A \otimes B$ and significantly reduces the number of multiplications involved. A matrix-vector product involving a Kronecker sum can be similarly calculated,

$$(A \oplus B)\,\mathbf{x} = (A \otimes I_B + I_A \otimes B)\,X^S = \left(I_B X A^T + B X I_A^T\right)^S = \left(XA^T + BX\right)^S$$

And, finally, a matrix-vector product involving $N-1$ Kronecker sums of the $n_a \times n_a$ matrices $A_1, A_2, \ldots, A_N$, that is,

$$\left(\bigoplus_{i=1}^{N} A_i\right)\mathbf{x} = \left(\sum_{i=1}^{N} I_{A_1} \otimes \cdots \otimes A_i \otimes \cdots \otimes I_{A_N}\right)\mathbf{x}$$

can be calculated....

Example MATLAB Code

```
function [ y ] = mvkronsum(A,B,x)
% MVKRONSUM calculates the matrix-vector product Y = (KRON(A,IB)+KRON(IA,B))*X
% without constructing KRON(A,IB) or KRON(IA,B).  In the above expression, IA
% and IB are identity matrices of the size of A and B, respectively.

[ ma, na ] = size(A);
[ mb, nb ] = size(B);
[ mx, nx ] = size(x);

y = zeros(mx,nx);
for i = 1:nx
   X = reshape(x(:,i),[nb ma]);
   y(:,i) = reshape(X*A'+B*X,[mx 1]);
end
```