Cardiovascular Dynamics *in-silico*

Pharmacological Targeting of Long QT 3 Syndrome:
Proof of Concept for an *in-silico* Drug Development Platform

Jonathan D. Moreno

MD/PhD Candidate
Colleen Clancy Laboratory
Institute for Computational Biomedicine
Weill Cornell Medical College

Outline

- Biophysics | Clinical Medicine | Cardiovascular Pharmacology | Mathematics | Engineering | HPC
- Ion-channel mutations (long QT)
  - Biophysics, genotype, phenotype and clinical characteristics
- Current treatment strategies
- *In-silico* predictive modeling and computational biomedicine
  - Formulation of Markov models of drug-receptor interactions
  - Channel | cell | cable | tissue | 3D heart
    - Numerical techniques
    - High performance computing

Ion-channel mutations and their arrhythmogenicity

- ***Channelopathy:*** Mutations in genes that encode ion channels can lead to abnormal channel function leading to perturbation of the AP to cause arrhythmias. Specific Na^+ channel mutations:
  - Gain of function during AP plateau (LQT)
  - Overall loss of channel function (Brugada, ICCD, SSS)

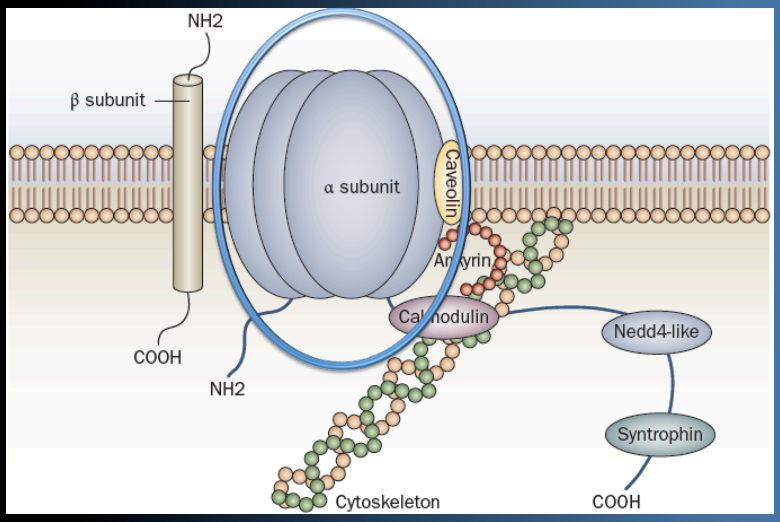
- Hereditary Long QT (LQT) syndrome
  - Prolongation of QT interval on ECG can lead to life-threatening arrhythmias and sudden cardiac death
  - ***LQT3:*** Usually bradycardic; occurring during sleep or relaxation

The Long QT (3) Syndrome

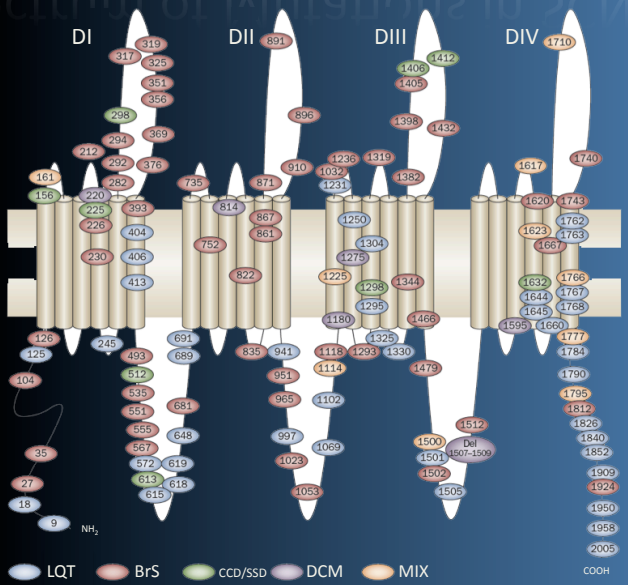
- Heterogeneous group of mutations in cardiac sodium channel α subunit
  - Genetically distinct, clinical presentation similar with subtle differences
  - Overlapping syndromes with clinical characteristics coexisting in a single patient

- ΔKPQ identified in 1995
  - Transient failure of inactivation → persistent I_{Na}
- D1790G: C terminal mutation, LQT3 clinical phenotype, possible distinct mechanism of action

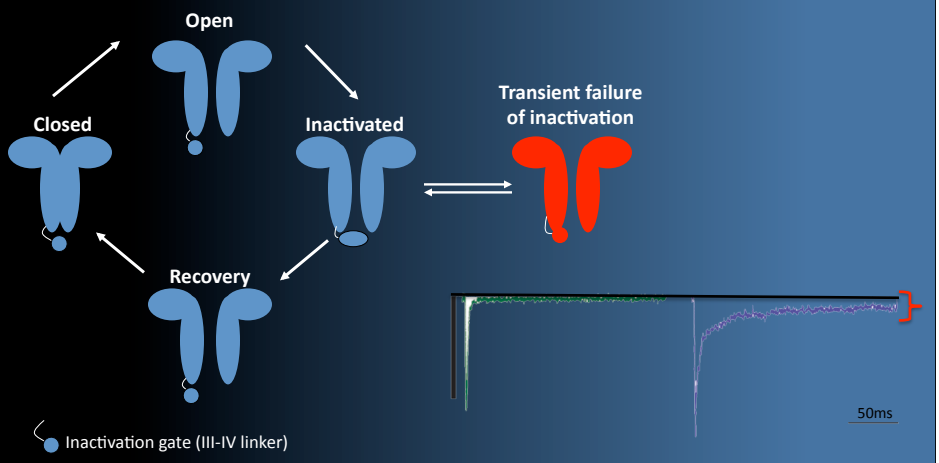
Na^+ Channel Macromolecular Complex



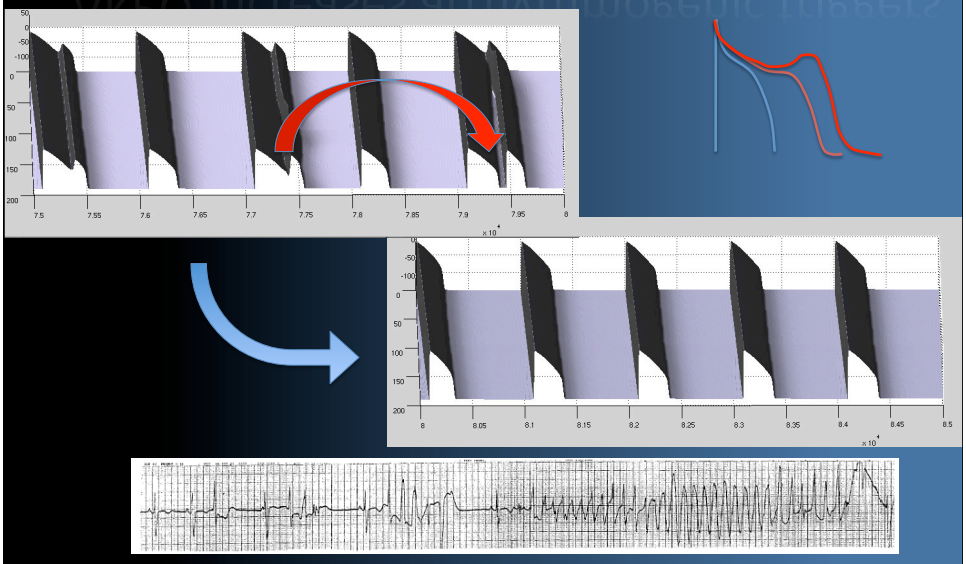
Spectrum of Mutations in SCN5A



The Excitatory Cycle of an Ion Channel



ΔKPQ increases arrhythmogenic triggers



Hodgkin Huxley Formulation

- HH formulation computes conductance for each current as a function of the P(O) of a series of hypothetical gates
  - Conductance is f(t, V) via gates
  - 1st order transitions from C → O and O → C that are *independent* of the other gates
  - Ions can only pass through the open state of the gate
- Based on experimental data, Na^+ activation can be modeled by 3 identical activation gates (m^3)
- Inactivation shortly after activation (h)
- LRd also includes slow inactivation gate (j)

$$I_{Na} = G_{Na} * m^3 * h * j * (V - E_{Na})$$

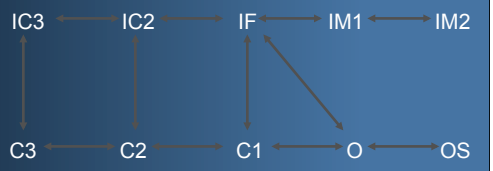
Markov-based Models

Motivation

- Need for models with explicit representation of single ion-channel states
- HH models are limited in their ability to describe specific aspects of single channel behavior
  - Inactivation of the Na^+ channel greater when in state O
- Assumption of independent gating ($m^3 \cdot h \cdot j$) fails
- MM can represent the dependence of a given transition on the occupancy of different states of the channel
  - Assume that transitions between channel states depend on the present conformation of the channel, but not on previous behavior

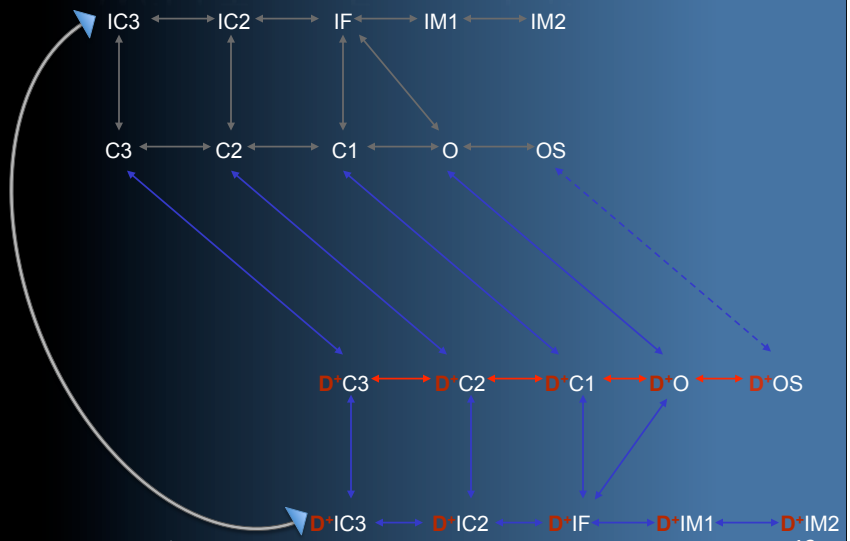
Formulating the model

- Wild Type: Use as many states as necessary to recapitulate the kinetics of channels
- Experimental data:
  - k_{on}, k_{off} rates
  - pKa – for drug partitioning
  - Steady State Availability
  - Tonic Block - kd_{closed}
  - Use Dependent Block - kd_{open}
  - Frequency dependence of UDB
  - Recovery from UDB

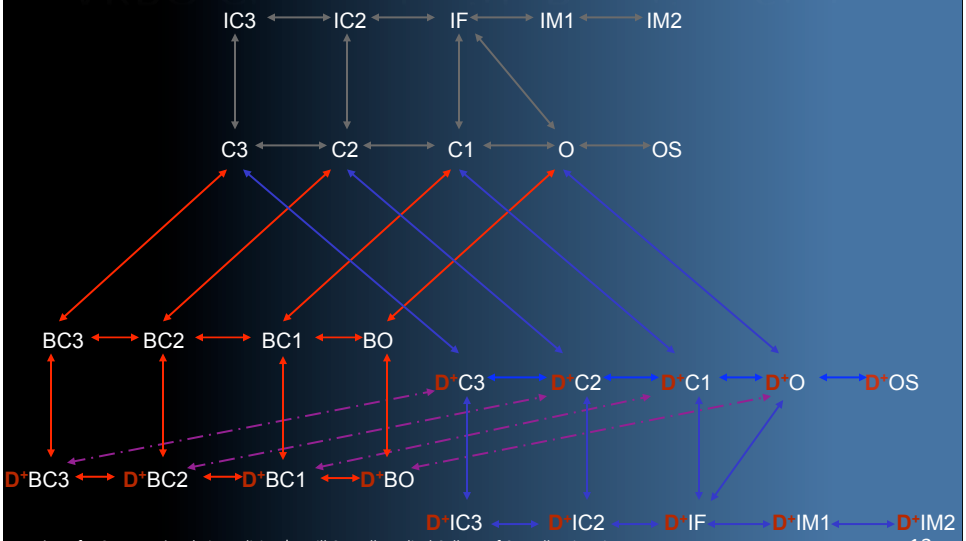


- Models adhere to 2nd law of thermodynamics (microscopic reversibility)
- Start playing!

Wild Type Formulation



ΔKPQ Channel with Bursting States



Runge Kutta Formulation

Forward Euler
- Error $O(h^1)$

$$y_{n+1} = y_n + h * f(x_n, y_n)$$

Fourth Order Runge Kutta
- Error $O(h^4)$

$$\frac{dy}{dt} = f(t,y)$$

$$k_1 = h * f(t_n, y_n)$$

$$k_2 = h * f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = h * f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = h * f\left(t_n + h, y_n + k_3\right)$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + Error$$

…some code

```
//Drug Free States
k1_O = hh*(a13*Cell_ptr->mC1 + b2*Cell_ptr->mIF +koff*Cell_ptr->DmO + k_off*Cell_ptr->D_mO + bx*Cell_ptr->mOS + mu2*Cell_ptr->mBO - Cell_ptr->mO*(b13 + a2 + mu1 + kon + k_on + ax));

k1_C1 = hh*(a12*Cell_ptr->mC2 + a3*Cell_ptr->mIF + b13*Cell_ptr->mO + mu2*Cell_ptr->mBC1 + kcoff*Cell_ptr->DmC1 + kc_off*Cell_ptr->D_mC1 - Cell_ptr->mC1*(b12 + b3 + a13 + mu1 + kcon + kc_on));
k1_C2 = hh*(a11*Cell_ptr->mC3 + a3*Cell_ptr->mIC2 + b12*Cell_ptr->mC1 + mu2*Cell_ptr->mBC2 + kcoff*Cell_ptr->DmC2 + kc_off*Cell_ptr->D_mC2 - Cell_ptr->mC2*(b11 + b3 + a12 + mu1 + kcon + kc_on));
k1_C3 = hh*(a3*Cell_ptr->mC3 + b11*Cell_ptr->mC2 + mu2*Cell_ptr->mBC3 + kcoff*Cell_ptr->DmC3 + kc_off*Cell_ptr->D_mC3 - Cell_ptr->mC3*(b3 + a11 + mu1 + kcon + kc_on));
k1_IC3 = hh*(b3*Cell_ptr->mC3 + b11*Cell_ptr->mIC2 + ki_off*Cell_ptr->D_mIC3 - Cell_ptr->mIC3*(a11 + a3 + ki_on));
k1_IC2 = hh*(a11*Cell_ptr->mIC3 + b3*Cell_ptr->mC2 + b12*Cell_ptr->mIF + ki_off*Cell_ptr->D_mIC2 - Cell_ptr->mIC2*(b11 + a3 + a12 + ki_on));
k1_IF = hh*(a12*Cell_ptr->mIC2 + b3*Cell_ptr->mC1 + b4*Cell_ptr->mIM1 + a2*Cell_ptr->mO + ki_off*Cell_ptr->D_mIF - Cell_ptr->mIF*(b12 + b2 + a3 + a4 + ki_on));
k1_IM1 = hh*(a4*Cell_ptr->mIF + b5*Cell_ptr->mIM2 - Cell_ptr->mIM1*(b4 + a5));
k1_IM2 = hh*(a5*Cell_ptr->mIM1 - Cell_ptr->mIM2*(b5));

k1_OS = hh*(ax*Cell_ptr->mO + ki_off*Cell_ptr->D_mOS - Cell_ptr->mOS*(bx + ki_on));
k1_BC3 = hh*(mu1*Cell_ptr->mC3 + b11*Cell_ptr->mBC2 + kcboff*Cell_ptr->DmBC3 + kc_off*Cell_ptr->D_mBC3 - Cell_ptr->mBC3*(mu2 + a11 + kcon + kc_on));
k1_BC2 = hh*(mu1*Cell_ptr->mC2 + a11*Cell_ptr->mBC3 + b12*Cell_ptr->mBC1 + kcboff*Cell_ptr->DmBC2 + kc_off*Cell_ptr->D_mBC2 - Cell_ptr->mBC2*(mu2 + b11 + a12 + kcon + kc_on));
k1_BC1 = hh*(mu1*Cell_ptr->mC1 + a12*Cell_ptr->mBC2 + b13*Cell_ptr->mBO + kcboff*Cell_ptr->DmBC1 + kc_off*Cell_ptr->D_mBC1 - Cell_ptr->mBC1*(mu2 + b12 + a13 + kcon + kc_on))
k1_BO = hh*(mu1*Cell_ptr->mO + a13*Cell_ptr->mBC1 + kboff*Cell_ptr->DmBO + k_off*Cell_ptr->D_mBO - Cell_ptr->mBO*(mu2 + b13 + kon + k_on));

//Charged Drug Bound States
k1_DO = hh*(kon*Cell_ptr->mO + a13c*Cell_ptr->DmC1 + bx1*Cell_ptr->DmOS + b22*Cell_ptr->DmIF + mu2*Cell_ptr->DmBO - Cell_ptr->DmO*(koff + b13c + a22 + ax1 + mu1));
k1_DC1 = hh*(kon*Cell_ptr->mC1 + a12*Cell_ptr->DmC2 + a33*Cell_ptr->DmIF + b13c*Cell_ptr->DmO + mu2*Cell_ptr->DmBC1 - Cell_ptr->DmC1*(koff + b12 + b33 + a13c + mu1));
k1_DC2 = hh*(kon*Cell_ptr->mC2 + a11*Cell_ptr->DmC3 + a33*Cell_ptr->DmIC2 + b12*Cell_ptr->DmC1 + mu2*Cell_ptr->DmBC2 -Cell_ptr->DmC2*(koff + b11 + b33 + a12 + mu1));
k1_DC3 = hh*(kon*Cell_ptr->mC3 + b11*Cell_ptr->DmC2 + a33*Cell_ptr->DmIC3 + mu2*Cell_ptr->DmBC3 - Cell_ptr->DmC3*(koff+ b33 + a11 + mu1));
k1_DOS = hh*(ax1*Cell_ptr->DmO - bx1*Cell_ptr->DmOS);
k1_DIC3 = hh*(b33*Cell_ptr->DmC3 + b11*Cell_ptr->DmIC2 - Cell_ptr->DmIC3*(a11 + a33));
k1_DIC2 = hh*(b33*Cell_ptr->DmC2 + a11*Cell_ptr->DmIC3 + b12*Cell_ptr->DmIF - Cell_ptr->DmIC2*(a33 + b11 + a12));
k1_DIF = hh*(b33*Cell_ptr->DmC1 + a12*Cell_ptr->DmIC2 + bx4*Cell_ptr->DmIM1 + a22*Cell_ptr->DmO - Cell_ptr->DmIF*(a33 + b12 + a44 + b22));
k1_DIM1 = hh*(a44*Cell_ptr->DmIF + b55*Cell_ptr->DmIM2 - Cell_ptr->DmIM1*( b44 + a55));
k1_DIM2 = hh*(a55*Cell_ptr->DmIM1 - b55*Cell_ptr->DmIM2);
k1_DBO = hh*(kbon*Cell_ptr->mBO + a13*Cell_ptr->DmBC1 + mu1*Cell_ptr->DmO - Cell_ptr->DmBO*(kboff + b13 + mu2));
k1_DBC1 = hh*(kcbon*Cell_ptr->mBC1 + a12*Cell_ptr->DmBC2 + b13*Cell_ptr->DmBO + mu1*Cell_ptr->DmC1 - Cell_ptr->DmBC1*(kcboff + b12 + a13 + mu2));
k1_DBC2 = hh*(kcbon*Cell_ptr->mBC2 + a11*Cell_ptr->DmBC3 + b12*Cell_ptr->DmBC1 + mu1*Cell_ptr->DmC2 - Cell_ptr->DmBC2*(kcboff + b11 + a12 + mu2));
k1_DBC3 = hh*(kcbon*Cell_ptr->mBC3 + b11*Cell_ptr->DmBC2 + mu1*Cell_ptr->DmC3 - Cell_ptr->DmBC3*(kcboff + a11 + mu2));

//Neutral Drug Bound States
k1_D_O = hh*(k_on*Cell_ptr->mO + a13n*Cell_ptr->D_mC1 + b_22*Cell_ptr->D_mIF + bx2*Cell_ptr->D_mOS + mu2*Cell_ptr->D_mBO - Cell_ptr->D_mO*(k_off + b13n + a_22 + ax2 + mu1));
k1_D_C1 = hh*(kc_on*Cell_ptr->mC1 + a12*Cell_ptr->D_mC2 + a_33*Cell_ptr->D_mIF + b13n*Cell_ptr->D_mO + mu2*Cell_ptr->D_mBC1 - Cell_ptr->D_mC1*(kc_off + b12 + b_33 + a13n + mu1));
k1_D_C2 = hh*(kc_on*Cell_ptr->mC2 + a11*Cell_ptr->D_mC3 + a_33*Cell_ptr->D_mIC2 + b12*Cell_ptr->D_mC1 + mu2*Cell_ptr->D_mBC2 - Cell_ptr->D_mC2*(kc_off + b11 + b_33 + a12 + mu1));
k1_D_C3 = hh*(kc_on*Cell_ptr->mC3 + a_33*Cell_ptr->D_mIC3 + b11*Cell_ptr->D_mC2 + mu2*Cell_ptr->D_mBC3 - Cell_ptr->D_mC3*(kc_off + b_33 + a11 + mu1));
k1_D_OS = hh*(ax2*Cell_ptr->D_mO + ki_on*Cell_ptr->mOS - Cell_ptr->D_mOS*(bx2 + ki_off));
k1_D_IC3 = hh*(b_33*Cell_ptr->D_mC3 + b11*Cell_ptr->D_mIC2 + ki_on*Cell_ptr->mIC3 - Cell_ptr->D_mIC3*(a_33 + a11 + ki_off));
k1_D_IC2 = hh*(b_33*Cell_ptr->D_mC2 + a11*Cell_ptr->D_mIC3 + b12*Cell_ptr->D_mIF + ki_on*Cell_ptr->mIC2 - Cell_ptr->D_mIC2*(a_33 + b11 + a12 + ki_off));
k1_D_IF = hh*(b_33*Cell_ptr->D_mC1 + a12*Cell_ptr->D_mIC2 + a_22*Cell_ptr->D_mO + ki_on*Cell_ptr->mIF - Cell_ptr->D_mIF*(a_33 + b_22 + b12 + ki_off));
k1_D_BO = hh*(k_on*Cell_ptr->mBO + a13*Cell_ptr->D_mBC1 + mu1*Cell_ptr->D_mO - Cell_ptr->D_mBO*(k_off + b13 + mu2));
k1_D_BC1 = hh*(kc_on*Cell_ptr->mBC1 + a12*Cell_ptr->D_mBC2 + b13*Cell_ptr->D_mBO + mu1*Cell_ptr->D_mC1 - Cell_ptr->D_mBC1*(kc_off + b12 + a13 + mu2));
k1_D_BC2 = hh*(kc_on *Cell_ptr->mBC2 + a11*Cell_ptr->D_mBC3 + b12*Cell_ptr->D_mBC1 + mu1*Cell_ptr->D_mC2 - Cell_ptr->D_mBC2*(kc_off + b11 + a12+ mu2));
k1_D_BC3 = hh*(kc_on*Cell_ptr->mBC3 + b11*Cell_ptr->D_mBC2 + mu1*Cell_ptr->D_mC3 - Cell_ptr->D_mBC3*(kc_off + a11 + mu2));
```

```
O_k1 = Cell_ptr->mO + k1_O/2;
C1_k1 = Cell_ptr->mC1 + k1_C1/2;
C2_k1 = Cell_ptr->mC2 + k1_C2/2;
C3_k1 = Cell_ptr->mC3 + k1_C3/2;
IC3_k1 = Cell_ptr->mIC3 + k1_IC3/2;
IC2_k1 = Cell_ptr->mIC2 + k1_IC2/2;
IF_k1 = Cell_ptr->mIF + k1_IF/2;
IM1_k1 = Cell_ptr->mIM1 + k1_IM1/2;
IM2_k1 = Cell_ptr->mIM2 + k1_IM2/2;
OS_k1 = Cell_ptr->mOS + k1_OS/2;
BC3_k1 = Cell_ptr->mBC3 + k1_BC3/2;
BC2_k1 = Cell_ptr->mBC2 + k1_BC2/2;
BC1_k1 = Cell_ptr->mBC1 + k1_BC1/2;
BO_k1 = Cell_ptr->mBO + k1_BO/2;

DO_k1 = Cell_ptr->DmO + k1_DO/2;
DC1_k1 = Cell_ptr->DmC1 + k1_DC1/2;
DC2_k1 = Cell_ptr->DmC2 + k1_DC2/2;
DC3_k1 = Cell_ptr->DmC3 + k1_DC3/2;
DOS_k1 = Cell_ptr->DmOS + k1_DOS/2;
DIC3_k1 = Cell_ptr->DmIC3 + k1_DIC3/2;
DIC2_k1 = Cell_ptr->DmIC2 + k1_DIC2/2;
DIF_k1 = Cell_ptr->DmIF + k1_DIF/2;
DIM1_k1 = Cell_ptr->DmIM1 + k1_DIM1/2;
DIM2_k1 = Cell_ptr->DmIM2 + k1_DIM2/2;
DBO_k1 = Cell_ptr->DmBO + k1_DBO/2;
DBC1_k1 = Cell_ptr->DmBC1 + k1_DBC1/2;
DBC2_k1 = Cell_ptr->DmBC2 + k1_DBC2/2;
DBC3_k1 = Cell_ptr->DmBC3 + k1_DBC3/2;

D_O_k1 = Cell_ptr->D_mO + k1_D_O/2;
D_C1_k1 = Cell_ptr->D_mC1 + k1_D_C1/2;
D_C2_k1 = Cell_ptr->D_mC2 + k1_D_C2/2;
D_C3_k1 = Cell_ptr->D_mC3 + k1_D_C3/2;
D_OS_k1 = Cell_ptr->D_mOS + k1_D_OS/2;
D_IC3_k1 = Cell_ptr->D_mIC3 + k1_D_IC3/2;
D_IC2_k1 = Cell_ptr->D_mIC2 + k1_D_IC2/2;
D_IF_k1 = Cell_ptr->D_mIF + k1_D_IF/2;
D_BO_k1 = Cell_ptr->D_mBO + k1_D_BO/2;
D_BC1_k1 = Cell_ptr->D_mBC1 + k1_D_BC1/2;
D_BC2_k1 = Cell_ptr->D_mBC2 + k1_D_BC2/2;
D_BC3_k1 = Cell_ptr->D_mBC3 + k1_D_BC3/2;
```


The failure of antiarrhythmics

- Sudden Cardiac Death is the #1 cause of death in the US, CVD in general will be global cause of death by 2020
- CAST, CAST II trials:
  - an incomplete understanding and inability to predict interaction of intrinsically complex drug pharmacology with complexity of cardiac tissue
    - Drug receptor dynamics
    - Anatomical and electrical heterogeneity
    - Mutant channel interactions
- Off target effects of noncardiovascular therapeutics
  - 40% of all new pharmaceuticals affect repolarization process
  - Vast majority of drugs pulled off market (Vioxx)
  - Side effects may occur <1% of patient population
    - Predict vulnerable populations?
    - Discard compounds earlier in the product development pipeline?
  - 50:1 failure to success rate

"All you have to do is to decrease our attrition rate from around 98% to 96% and you would make us twice as successful."

Cardiovascular Dynamics *in-silico*

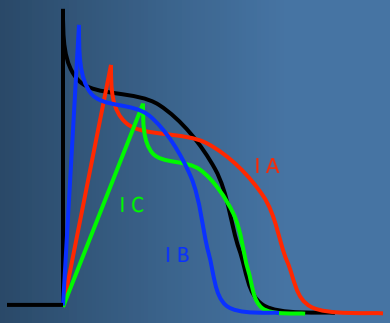
- Single channel → single cell → 1D fiber → 2,3D tissues
  - *Spatiotemporal trends that emerge in higher dimensions that allow for propagation of arrhythmia are missed at smaller spatial scales*
- Develop a computational framework for an *in-silico* drug screen of cardiovascular medications
  - Predictive simulations of pharmacodynamics in an engineered virtual cardiac tissue
  - Identification of key parameters for drug-receptor dynamics

Model "wish list"

- Transport phenomena: Diffusion
- Kinetics of drug receptor interactions
- Macroscopic wave propagation
- Scalability: 0D → 3D
- Optimize with high performance computing
  - Numerical optimizations
  - Parallel computations
  - GPGPU

Sodium Channel Block Subclassification

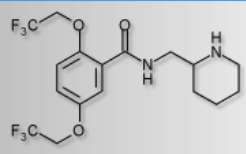
- Class IA (e.g., quinidine)
  - Moderate Na^+ channel blockade
  - ↑ ERP
- Class IB (e.g., lidocaine)
  - Weak Na^+ channel blockade
  - ↓ ERP
- Class IC (e.g., flecainide)
  - Strong Na^+ channel blockade
  - → ERP

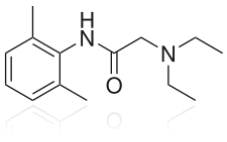


Antiarrhythmics

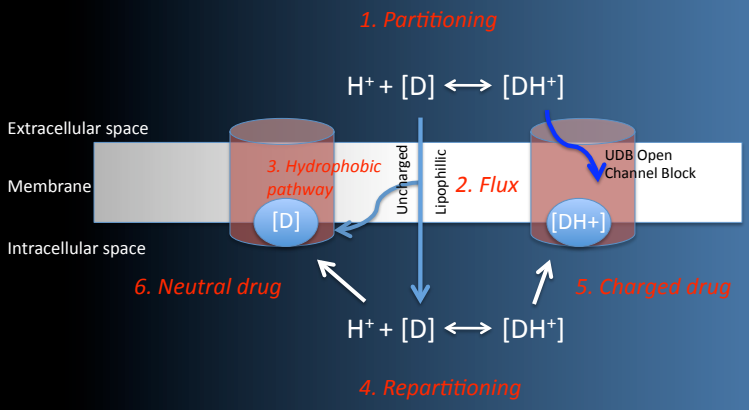
Flecainide
- Class 1C antiarrhythmic
- pKa ~ 9.3
- 99% charged at physiologic pH

Lidocaine
- Class 1B antiarrhythmic
- pKa ~ 7.6
- 50% charged at physiologic pH





pH Dependent Partitioning



Clancy, CE.

Channel | Cell | Cable | Tissue | 3D Heart