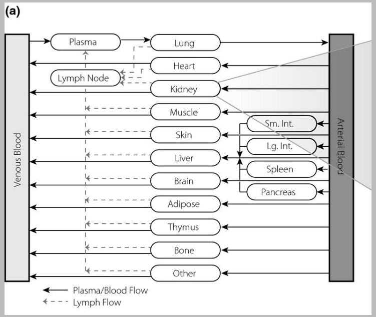
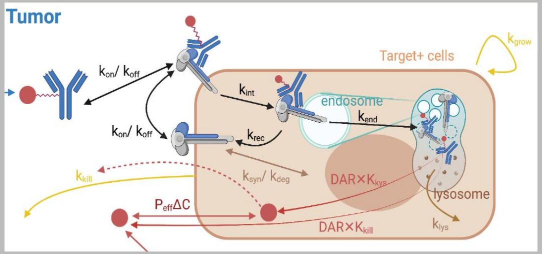
Open-Science Immuno-Oncology QSP Modeling Using Open-Source Julia Solvers

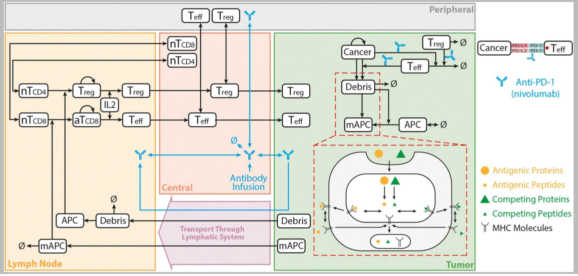
Mechanistic Distribution



Tumor Dynamics



Immune System Modeling



July 28, 2023

Eric Jordie, Ahmed Elmokadem*, Katharina Wilkins, Tim Knab, Yuezhe Li, Kiersten Utsey, Jimena Davis, Ellen Swanson

*Presenter

METRUM
RESEARCH GROUP

Who are we?

- We are a group of quantitative scientists assisting small biotech through large pharmaceutical companies to develop medicines

- We generate mathematical and statistical models to integrate preclinical and clinical knowledge to describe disease progression and inform drug development questions

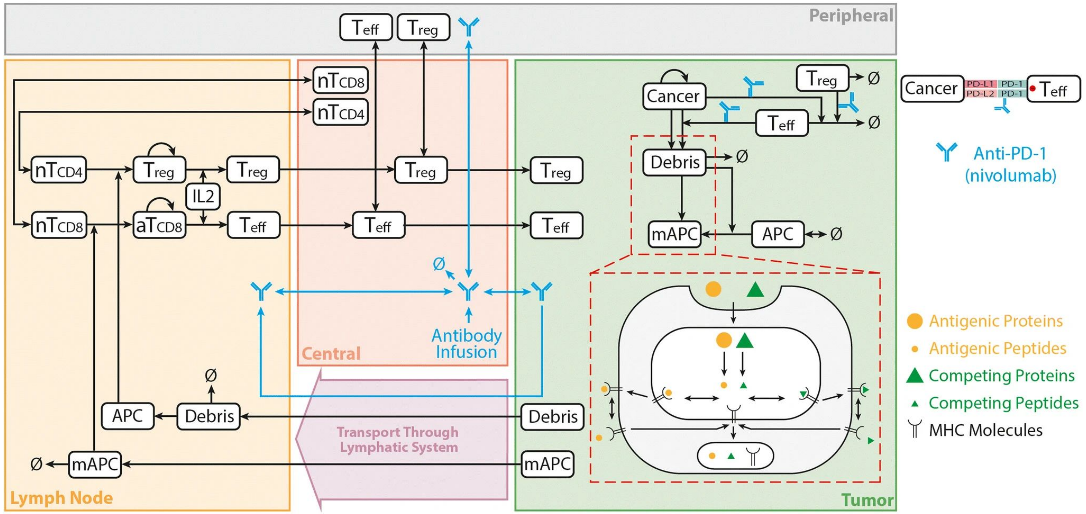
- We are dedicated to fostering and contributing to open science and coding initiatives as a way to accelerate progress

Why did MetrumRG select Julia as their primary systems pharmacology production platform?

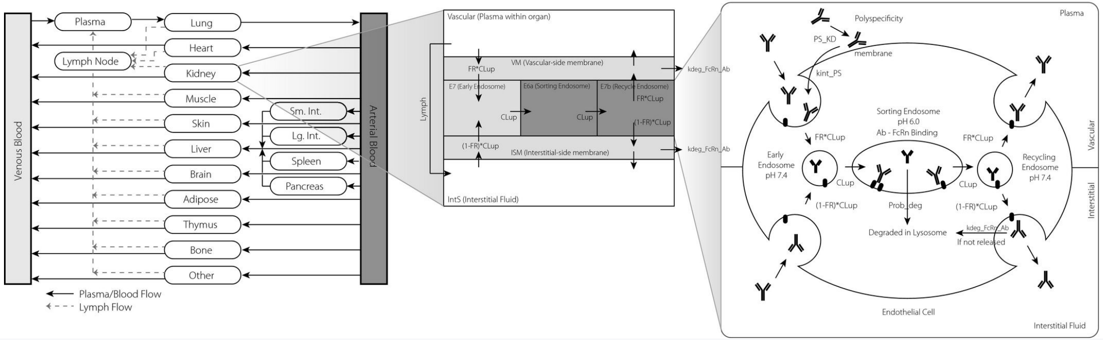
- Needed a fast, scalable, and reproducible open source platform capable of handling large (1000+ equations), multi-scale, spatial, and agent-based, and Bayesian models

- Needed a platform that integrates with other open source platforms (such as R)

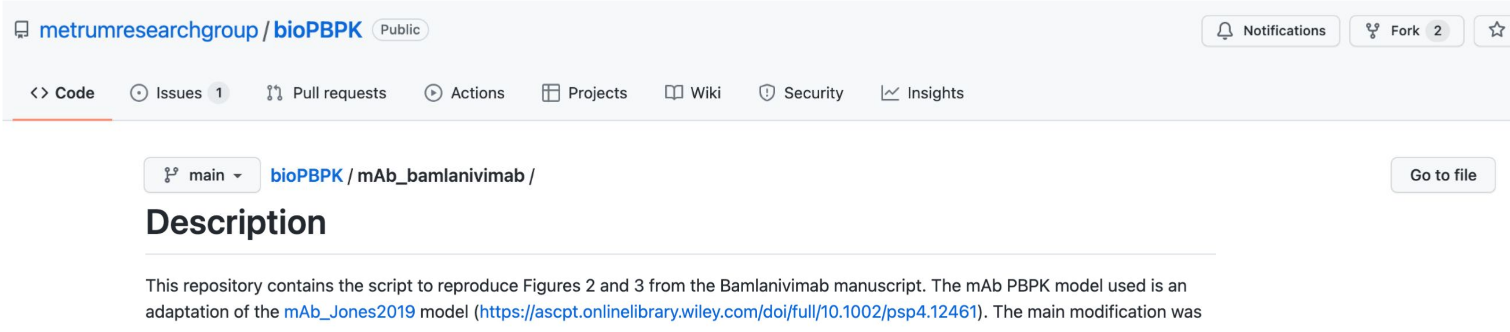
Example Immuno QSP Model:. Jafarnejad, 2019



MetrumRG has already published Open-Source Julia Production Codes for Monoclonal Antibody PBPK



https://ascpt.onlinelibrary.wiley.com/doi/full/10.1002/psp4.12461



Description

This repository contains the script to reproduce Figures 2 and 3 from the Bamlanivimab manuscript. The mAb PBPK model used is an adaptation of the mAb_Jones2019 model (https://ascpt.onlinelibrary.wiley.com/doi/full/10.1002/psp4.12461). The main modification was

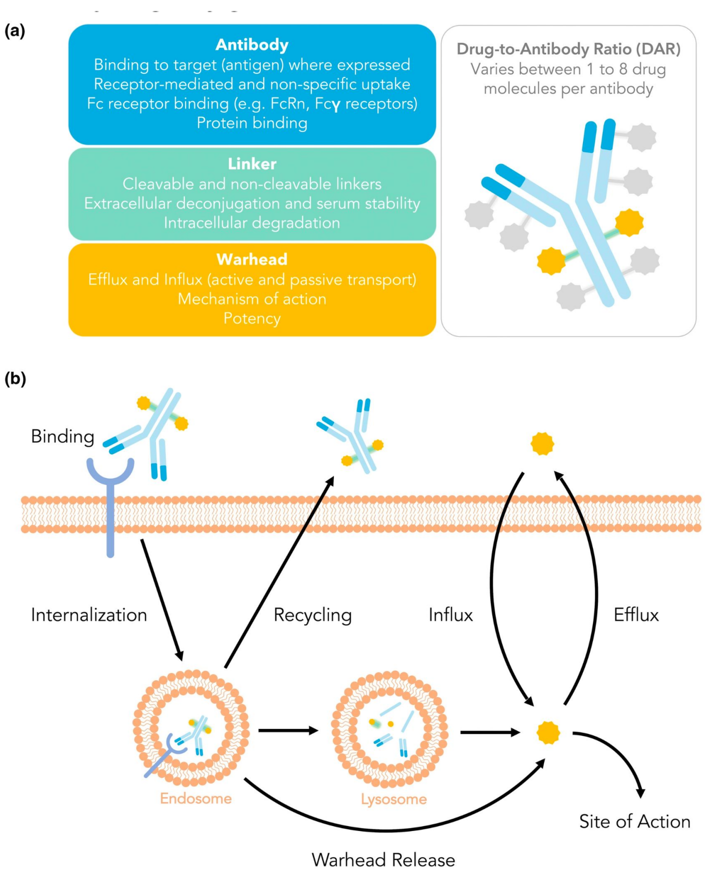
MetrumRG is also in the process of developing Open-Source Immuno-Oncology Models

Today's topic: Key Antibody-Drug Conjugate properties and mechanisms for QSP modeling

(a) The antibody, linker, and warhead components of ADCs each have different design properties that must be considered during modeling. Another key characteristic is the drug-to-antibody ratio (DAR), which typically varies between one and eight.

(b) Key mechanisms of action of the ADC include binding to the target antigen, internalization into the cell, trafficking and recycling of the ADC, endosomal cleavage of the linker or lysosomal degradation of the ADC for warhead release, influx and efflux of the warhead, and cell killing effects at the site of action.

ADC, antibody-drug conjugate; QSP, quantitative systems pharmacology.



Antibody
Binding to target (antigen) where expressed
Receptor-mediated and non-specific uptake
Fc receptor binding (e.g. FcRn, Fcγ receptors)
Protein binding

Linker
Cleavable and non-cleavable linkers
Extracellular deconjugation and serum stability
Intracellular degradation

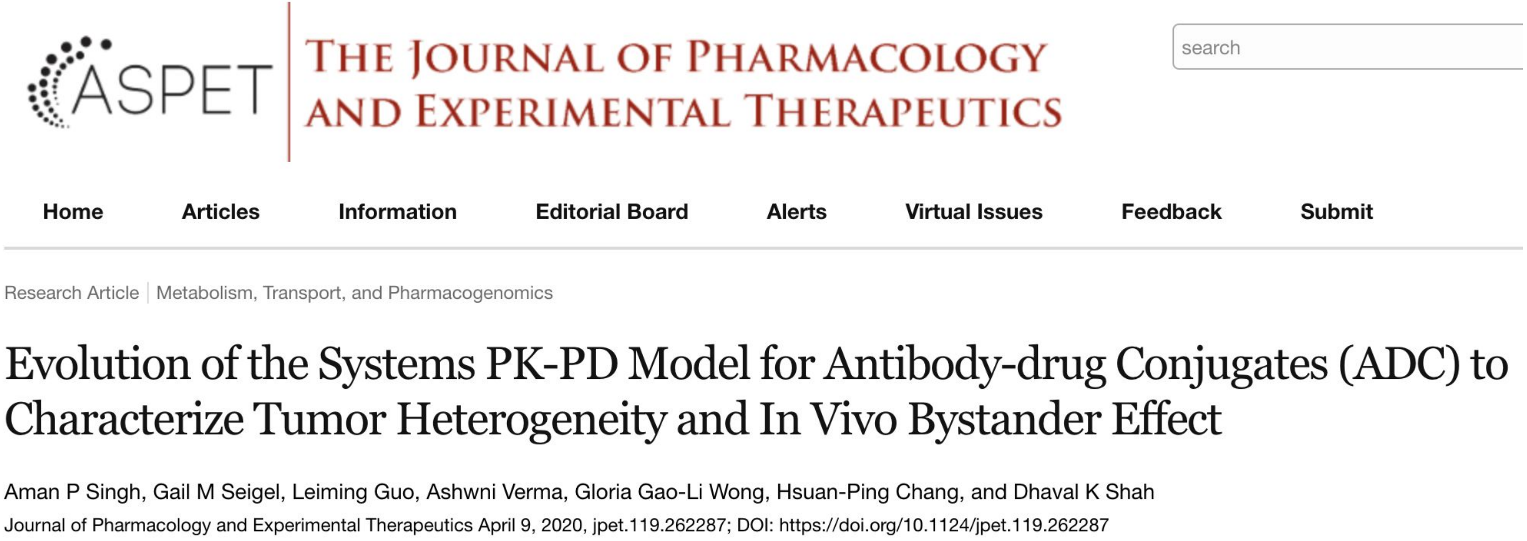
Warhead
Efflux and Influx (active and passive transport)
Mechanism of action
Potency

Drug-to-Antibody Ratio (DAR)
Varies between 1 to 8 drug molecules per antibody

From: Figure 1 of Lam, I., Pilla Reddy, V., Ball, K., Arends, R. H., & Mac Gabhann, F. (2022). Development of and insights from systems pharmacology models of antibody-drug conjugates. CPT: Pharmacometrics & Systems Pharmacology, 11(8), 967–990. https://doi.org/10.1002/psp4.12833

Motivating Example: *in vitro* ADC Modeling using Julia

- Drug development typically progresses from *in vitro* studies, to *in vivo* animal studies, and ultimately human clinical studies
- For today, we will look at the fundamental principles of the ADC *in vitro* system, data, and modeling abstracted from the following paper:



Research Article | Metabolism, Transport, and Pharmacogenomics

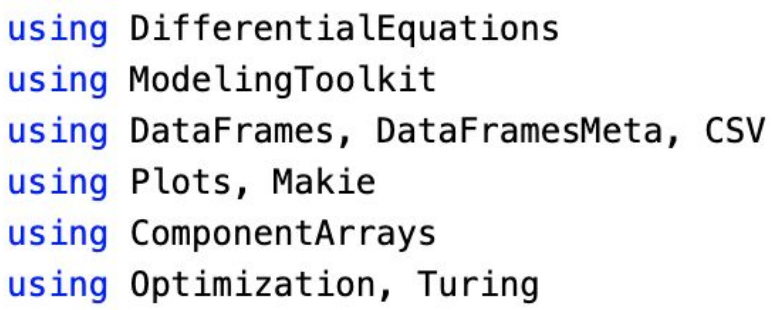
Evolution of the Systems PK-PD Model for Antibody-drug Conjugates (ADC) to Characterize Tumor Heterogeneity and In Vivo Bystander Effect

Aman P Singh, Gail M Seigel, Leiming Guo, Ashwni Verma, Gloria Gao-Li Wong, Hsuan-Ping Chang, and Dhaval K Shah

Journal of Pharmacology and Experimental Therapeutics April 9, 2020, jpet.119.262287; DOI: https://doi.org/10.1124/jpet.119.262287

https://jpet.aspetjournals.org/content/early/2020/04/09/jpet.119.262287

What Julia ecosystem does MetrumRG typically use?



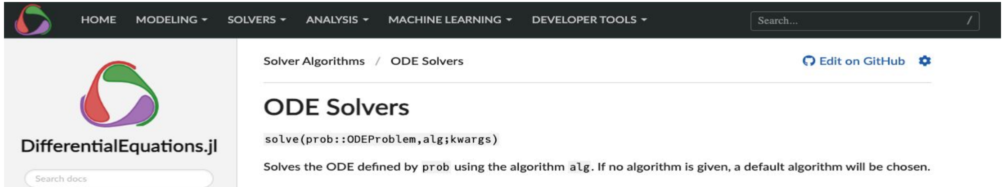
```
using DifferentialEquations
using ModelingToolkit
using DataFrames, DataFramesMeta, CSV
using Plots, Makie
using ComponentArrays
using Optimization, Turing
```

metworx

High Performance Cloud Computation Made Simple

Code editing. Redefined.

Free. Built on open source. Runs everywhere.



ODE Solvers

`solve(prob::ODEProblem,alg;kwargs)`

Solves the ODE defined by `prob` using the algorithm `alg`. If no algorithm is given, a default algorithm will be chosen.

SciML: Open Source Software for Scientific Machine Learning

METRUM
RESEARCH GROUP

Additional resources:

Repository with PBPK modeling example:

- https://github.com/metrumresearchgroup/cptpsp-tutorial-2019

bamlanivimab PBPK paper (Chigutsa et al. Vol 111(3) 2022, p. 595-604):

- https://ascpt.onlinelibrary.wiley.com/doi/10.1002/cpt.2459
- https://github.com/metrumresearchgroup/bioPBPK

Motivating Example: Draw it, Develop it, Deploy it

A schematic diagram of a systems pharmacokinetic model developed to characterize T-vc-MMAE PK in a heterogeneous tumor containing N87 and GFP-MCF7 cells

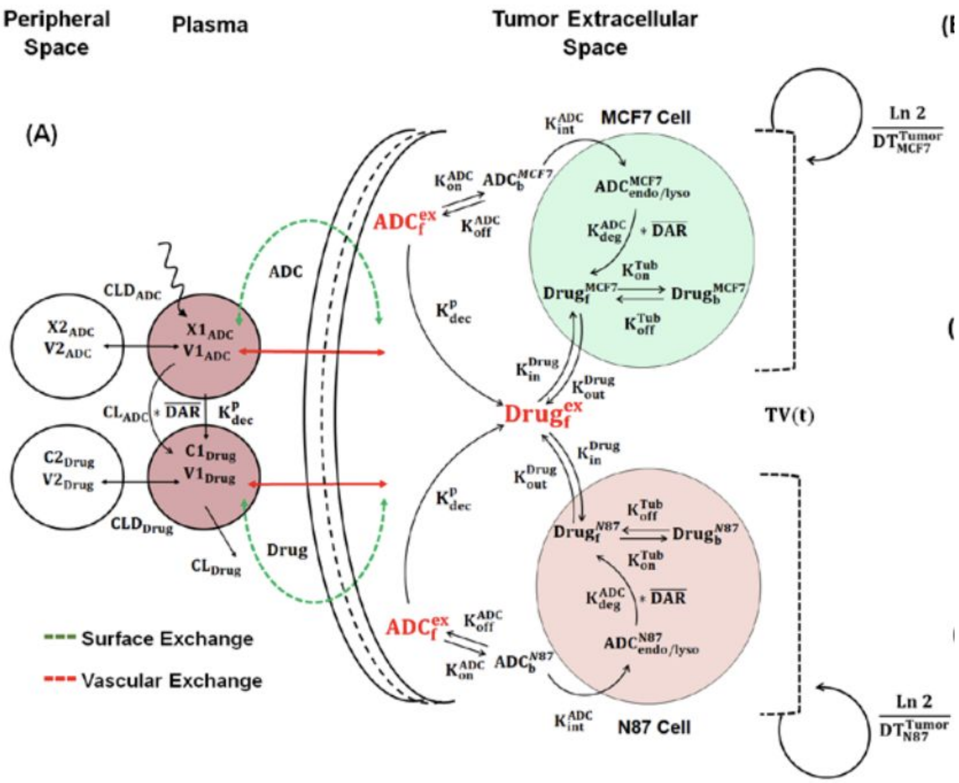
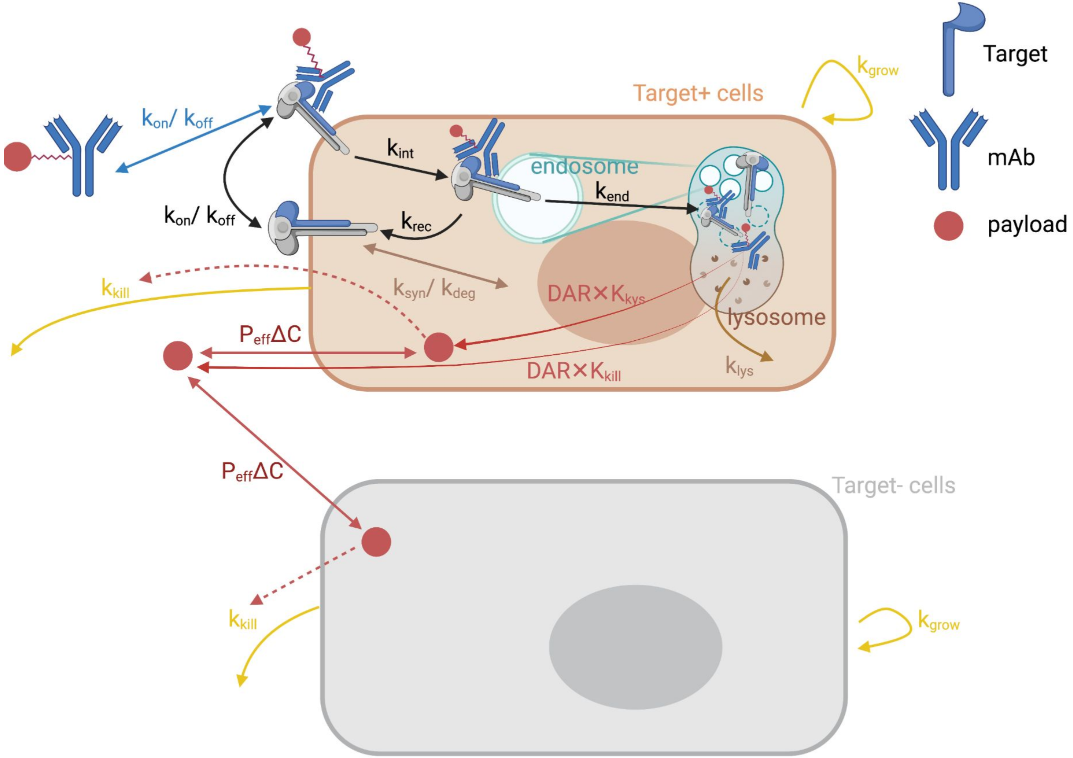


Figure 3 of Singh, A. P., Seigel, G. M., Guo, L., Verma, A., Wong, G. G.-L., Cheng, H.-P., & Shah, D. K. (2020). Evolution of the Systems Pharmacokinetics-Pharmacodynamics Model for Antibody-Drug Conjugates to Characterize Tumor Heterogeneity and In Vivo Bystander Effect. The Journal of Pharmacology and Experimental Therapeutics, 374(1), 184–199. https://jpet.aspetjournals.org/content/early/2020/04/09/jpet.119.262287

Simplified Case example: ADC schematic, *in vitro* only

Understanding the System

- Draw it!

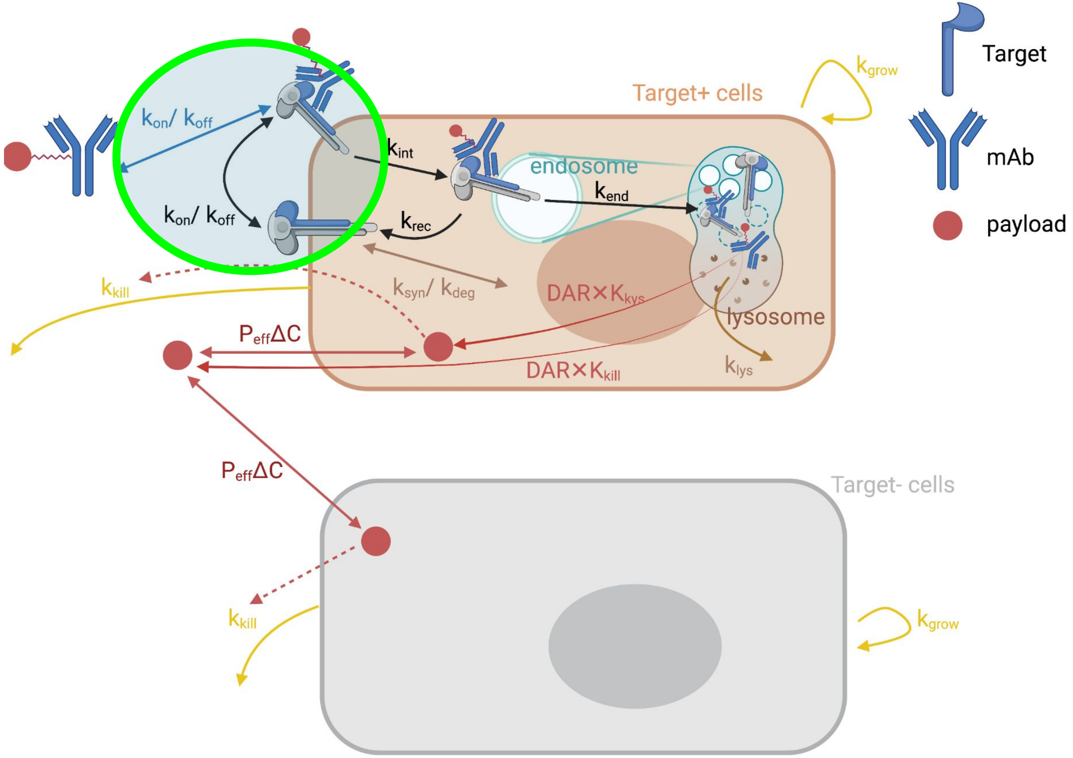


Understanding the data: *in vitro* ADC Modeling

Target Binding

Kon/Koff informed by:

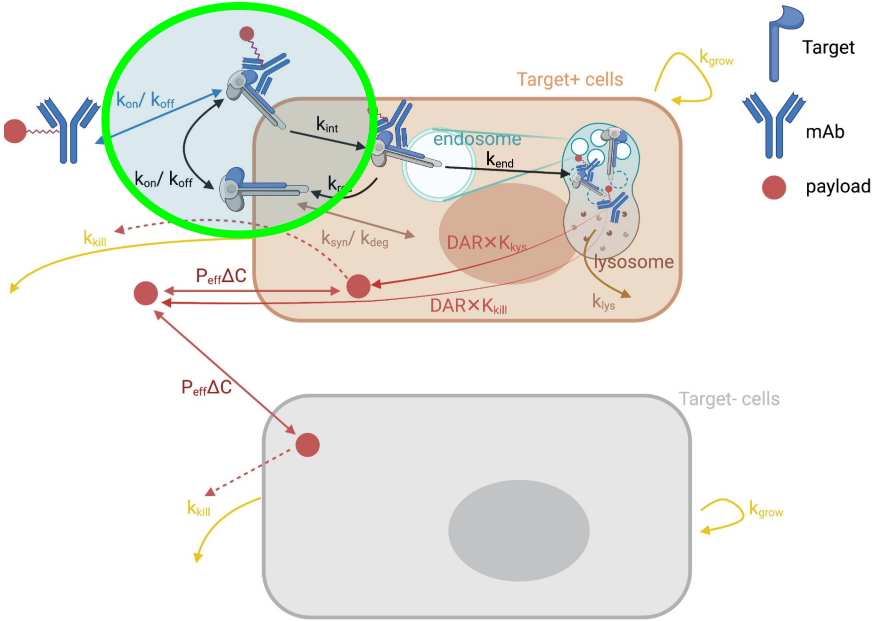
- Target affinity assays
  - SPR/Biacore affinity
  - Cell-based binding assay
  - Any considerations about multiple epitopes, avidity, bivalent binding?



Julia Model Code: Target Binding

```julia
function invitroADC(du, u, p, t)
   ...
   # Compartment volumes and surface areas
   Vm   = p[1]      # Media volume

   # Rate constants
   Kon  = p[2]  # ADC/receptor on rate constant
   Kd   = p[3]   # ADC/receptor dissociation rate

   # Get Koff from Kd and Kon
   Koff = Kd*Kon*6.022e23/1e9 ; # Convert back to nM

   # ADC in media binding to surface receptor
   flux_A_R_s_binding = A_m/Vm*R_s*Kon;
   # Surface ADC/receptor unbinding
   flux_AR_s_unbinding = AR_s*Koff;

   # ADCs in media
   # Flux = unbinding - binding
   du[1] = flux_AR_s_unbinding*Ntot - flux_A_R_s_binding*Ntot;
   ...
end
```

in vitro ADC Modeling

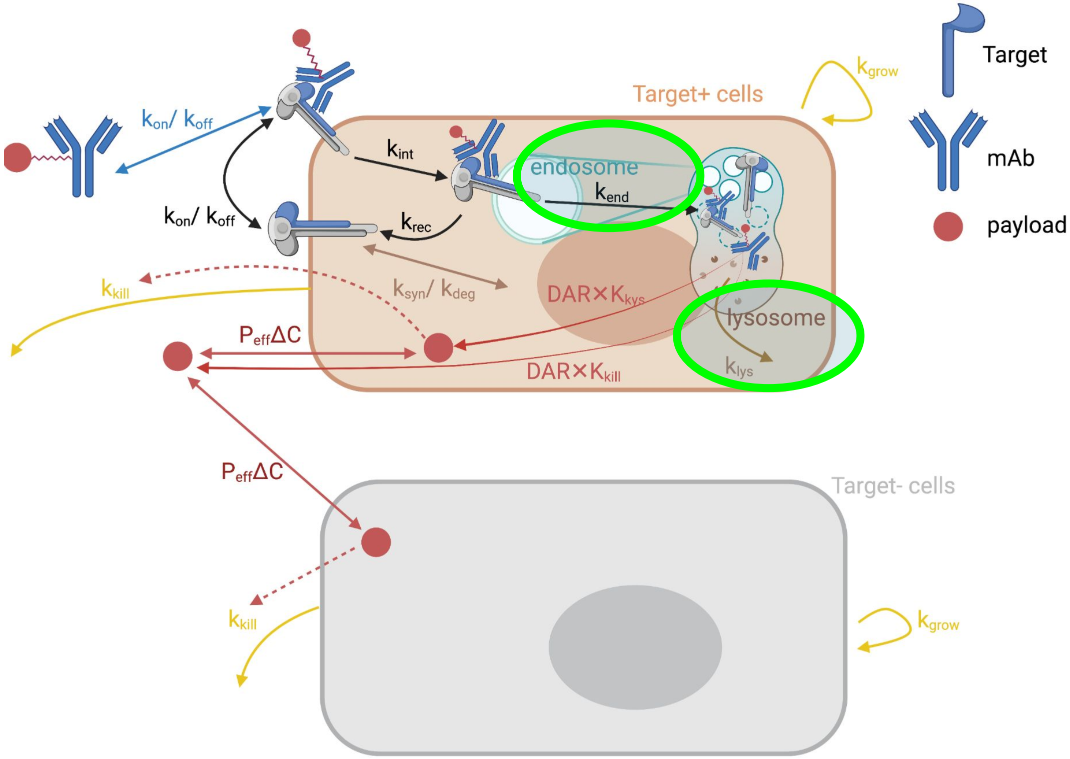


Understanding the data: Lysosomal Degradation

in vitro ADC Modeling

Lysosomal degradation rate (Klys) informed by:

- Lysosomal degradation rate
- Approximated by linker kinetics

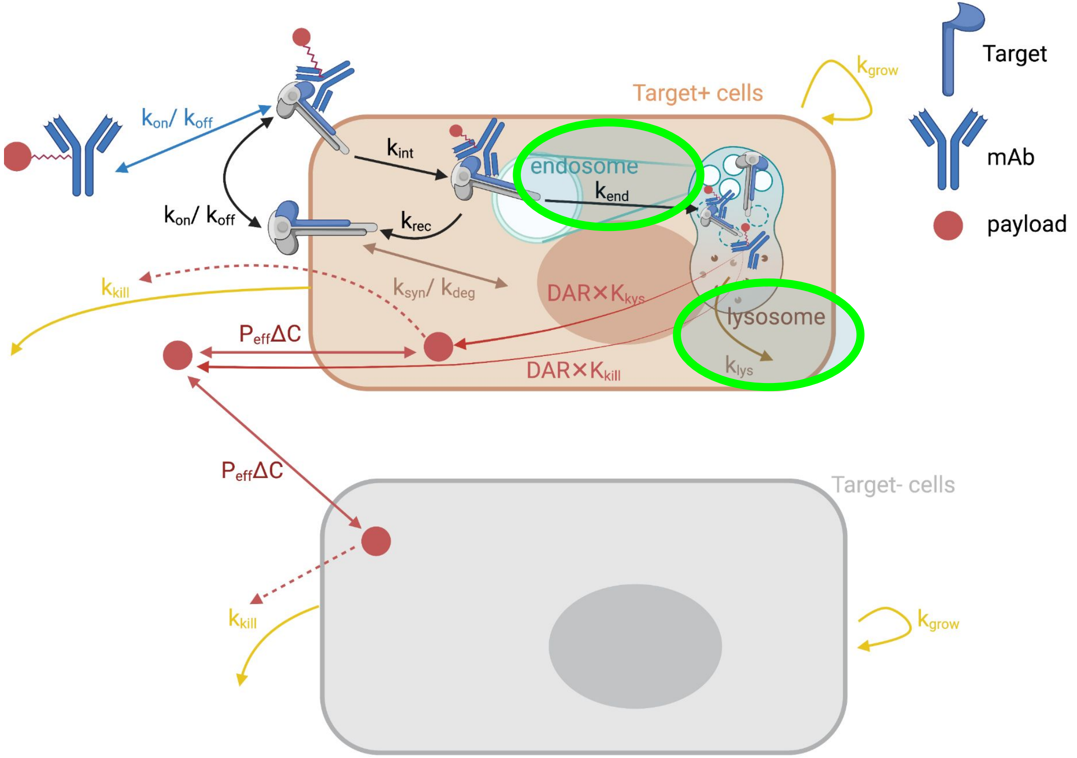


Understanding the data: Endosome-to-lysosome

in vitro ADC Modeling

Endosome-to-lysosome/bystander effect informed by combination of:

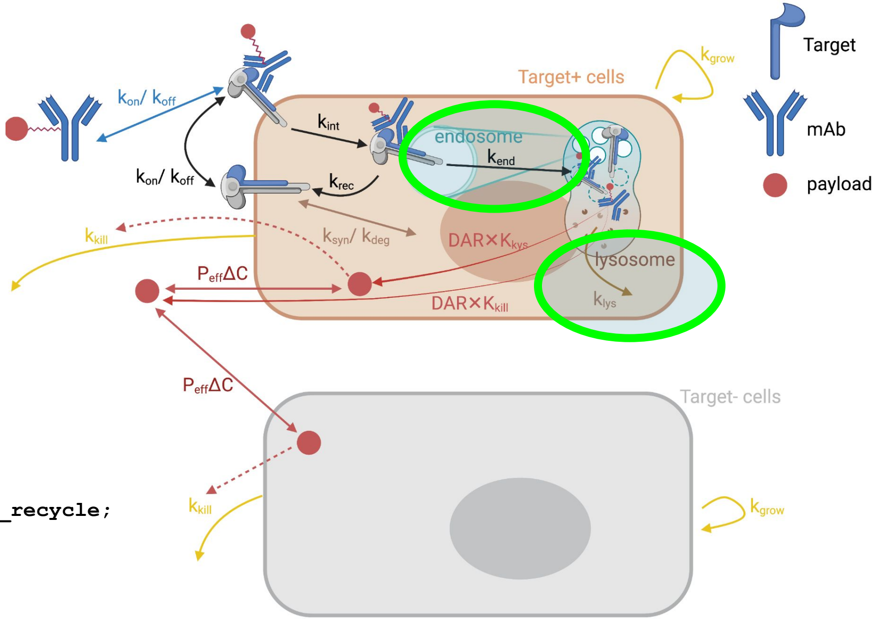
- Degradation rate in lysosome
- IC50 for payload
- Permeability (Peff)
- Concentration of payload in the media
- DAR



Model Code: Lysosomal degradation

```
function invitroADC(du, u, p, t)
    ...
    # Rate constants
    Klys = p[4]  # Lysosomal deg rate const int ADC/receptor
    Kend = p[5]  # Endosomal sorting rate for int ADC/receptor

    # Endosomal ADC/receptor complex transport to lysosome
    flux_AR_e_to_AR_l = AR_e*Kend;
    # Lysosomal ADC/receptor complex catabolized
    flux_AR_l_cat = AR_l*Klys;
    # Endosomal ADC/receptor unbinding
    flux_AR_e_unbinding = AR_e*Koff;

    # ADC/receptor complex in endosome
    # Flux = internalization - unbinding - transport to lysosome - recycling
    du[2] = flux_AR_s_int - flux_AR_e_unbinding - flux_AR_e_to_AR_l - flux_AR_e_recycle;

    # ADC/receptor complex in lysosome
    # Flux = transport from endosome - catabolism
    du[3] = flux_AR_e_to_AR_l - flux_AR_l_cat;

    ...
end
```

in vitro ADC Modeling

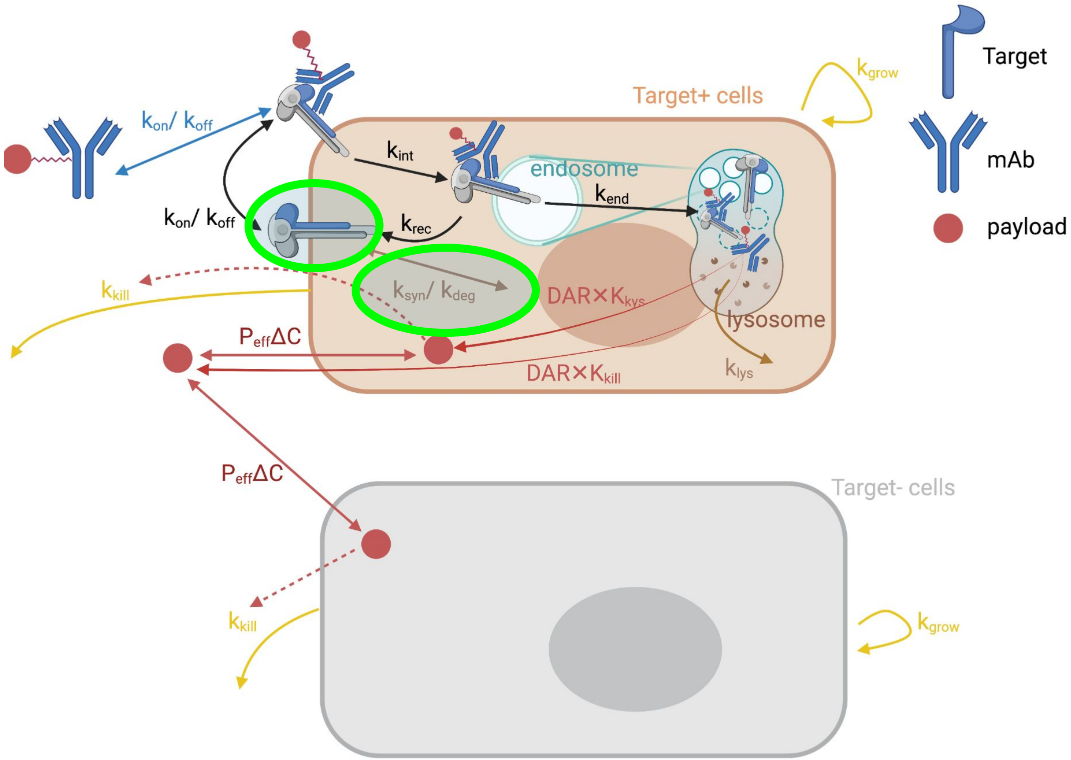


Understanding the data: Receptor Expression and Dynamics

in vitro ADC Modeling

Receptor Expression and Dynamics informed by:

- Receptor expression (immunofluoresence)
- Receptor shedding
- Competition with ligand
- Feedback upregulation/ downregulation
- Effects of receptor dimerization, phosphorylation, signaling

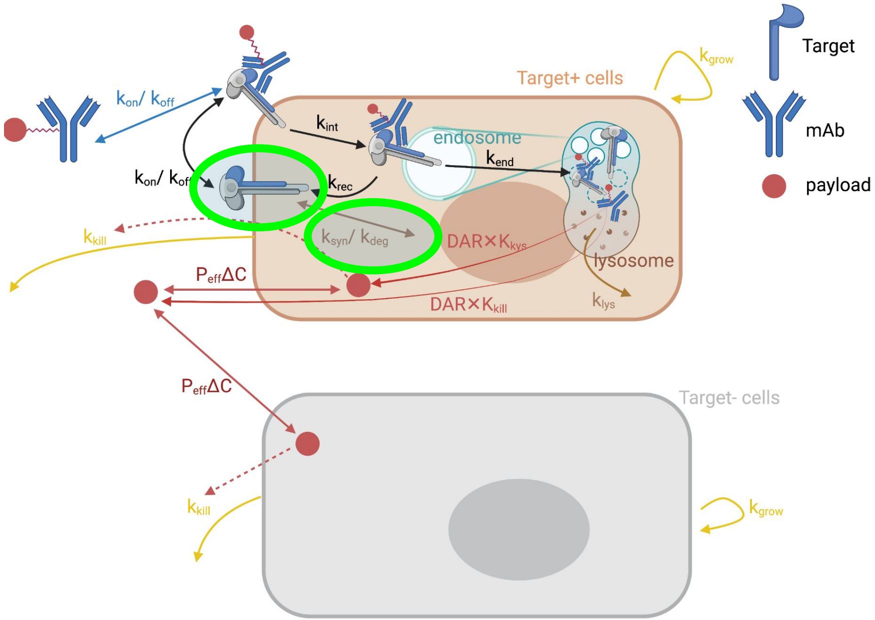


METRUM RESEARCH GROUP

Model Code: Receptor Expression and Dynamics

```
function invitroADC(du, u, p, t)
   ...
   # Calculate Ksyn
   Ksyn = Nr * Kdeg;
   Krec = p[6]  # Rate for receptors recycling to surface
   Krec_AR = p[7] # Rate of AR complex recycling to surface
   Nr = p[8]   # Surface receptor expression (receptors/cell)
   Kdeg = p[9]  # Surface-bound receptor degradation rate

   # Surface receptor synthesis and feedback
   flux_R_s_syn = Ksyn;
   # Endosomal receptor recycles to surface
   flux_R_e_recycle = R_e*Krec;

   # Free receptors on surface
   # Flux = synthesis + unbinding +
   #         recycling - binding - degradation
   du[4] = flux_R_s_syn + flux_AR_s_unbinding +
           flux_R_e_recycle - flux_A_R_s_binding -
           flux_R_s_degrade;
   ...
end
```

in vitro ADC Modeling



Understanding the data: Internalization and Recycling

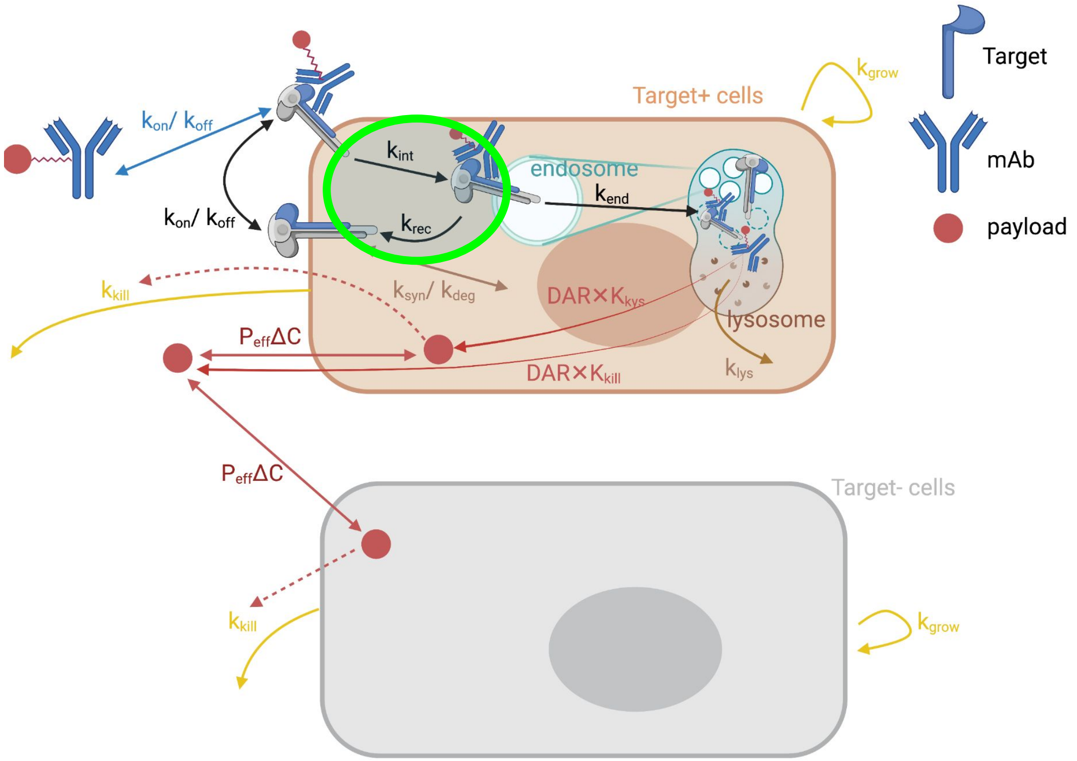
in vitro ADC Modeling

Internalization rate (Kint) informed by:

- Internalization assays, turnover assays

Recycling rates (Krec) informed by:

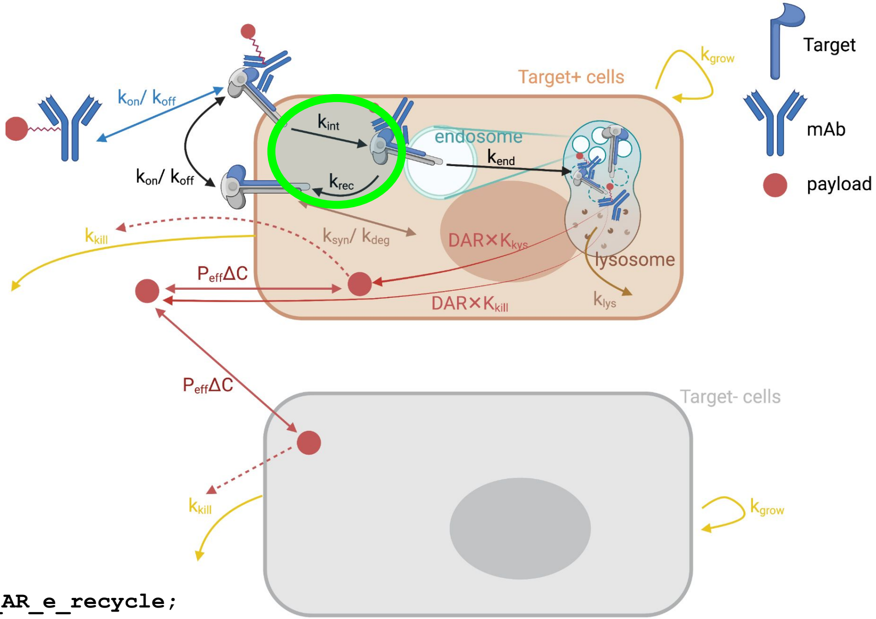
- receptor-alone and complex recycling rates (CHX)
- turnover assays



Model Code: Internalization and Recycling

```
function invitroADC(du, u, p, t)
   ...
   Kint = p[10]  # ADC/receptor complex int rate constant

   # Receptor occupancies (using baseline receptor abundance)
   RO = AR_s/R_s_0;

   # ADC/receptor complex internalizing to endosome
   flux_AR_s_int = AR_s*Kint;

   # Endosomal ADC/receptor recycles to surface
   flux_AR_e_recycle = AR_e*Krec_AR;

   # ADCs bound to cell surface
   # Flux = binding - unbinding - internalization + recycling
   du[5] = flux_A_R_s_binding - flux_AR_s_unbinding - flux_AR_s_int + flux_AR_e_recycle;

   ...
end
```

in vitro ADC Modeling



Understanding the data: Payload Release and Distribution

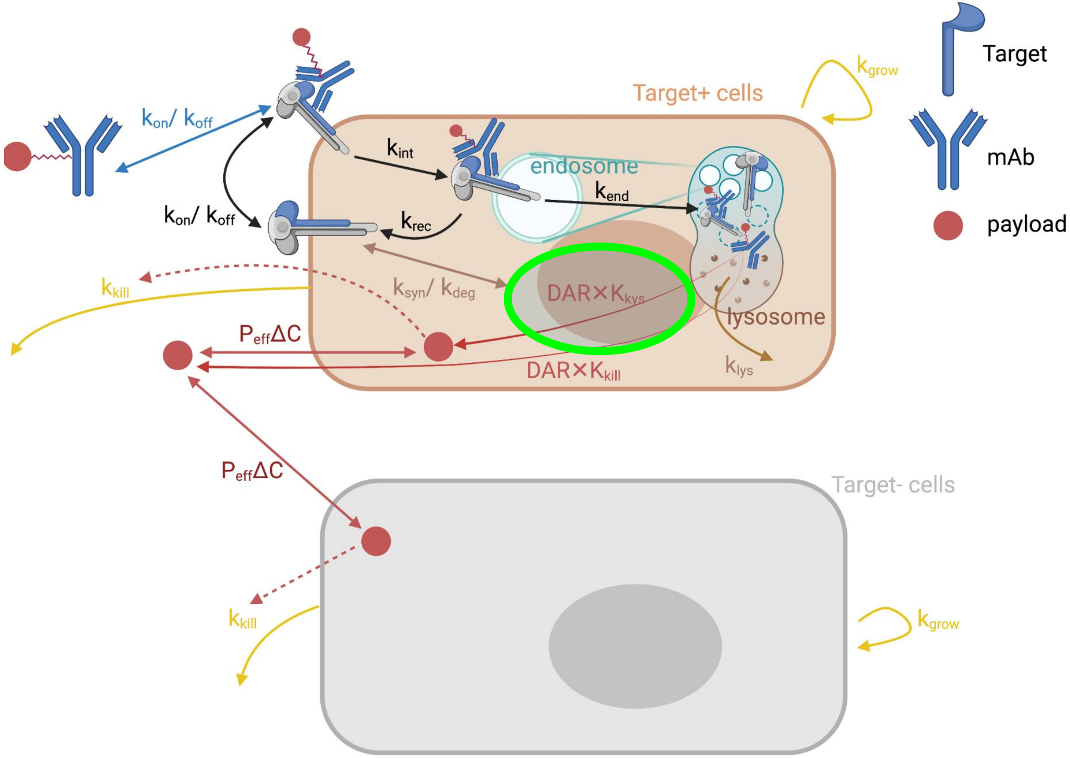
in vitro ADC Modeling

Payload release informed by:

- Linker stability
- pH-dependent linker cleavage?
- Protease linker cleavage?
- Intracellular environment?

Payload distribution informed by payload:

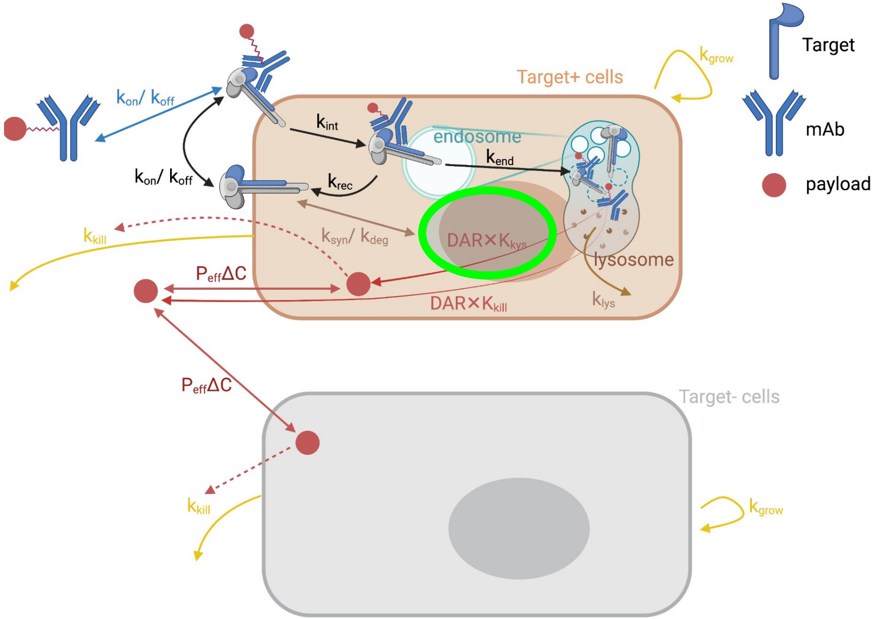
- Physchem, protein binding, cellular permeability, diffusivity, etc.



Model Code: Payload Release & Distribution

in vitro ADC Modeling

```julia
function invitroADC(du, u, p, t)
    ...
    Klys = p[11]  # Lysosomal deg rate constant int ADC/receptor



    # Lysosomal ADC/receptor complex catabolized
    flux_AR_l_cat = AR_l*Klys;

    # Lysosomal antibody catabolized
    flux_A_l_cat = A_l*Klys;

    # ADC/receptor complex in lysosome
    # Flux = transport from endosome - catabolism
    du[6] = flux_AR_e_to_AR_l - flux_AR_l_cat;

    ...
end
```

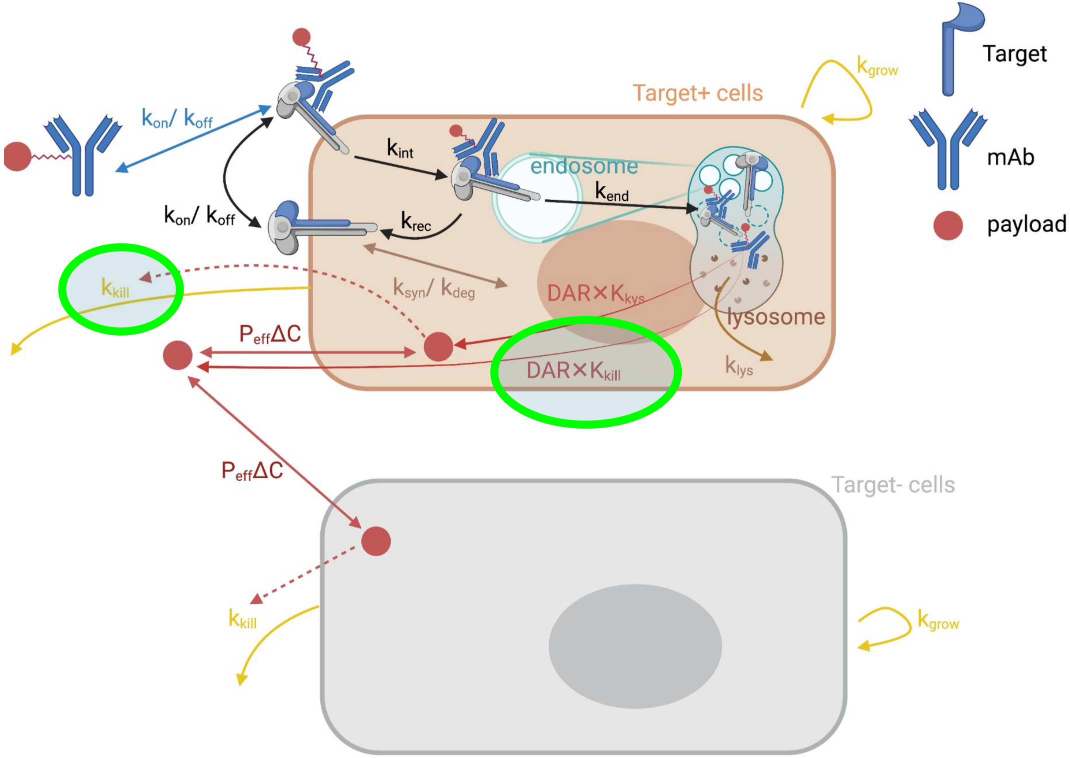


Understanding the data: Cell Killing

Cell killing effect (Kkill) depends on mechanism of action, but generally informed by:

- Payload Release
- IC50s
- Cell half lives

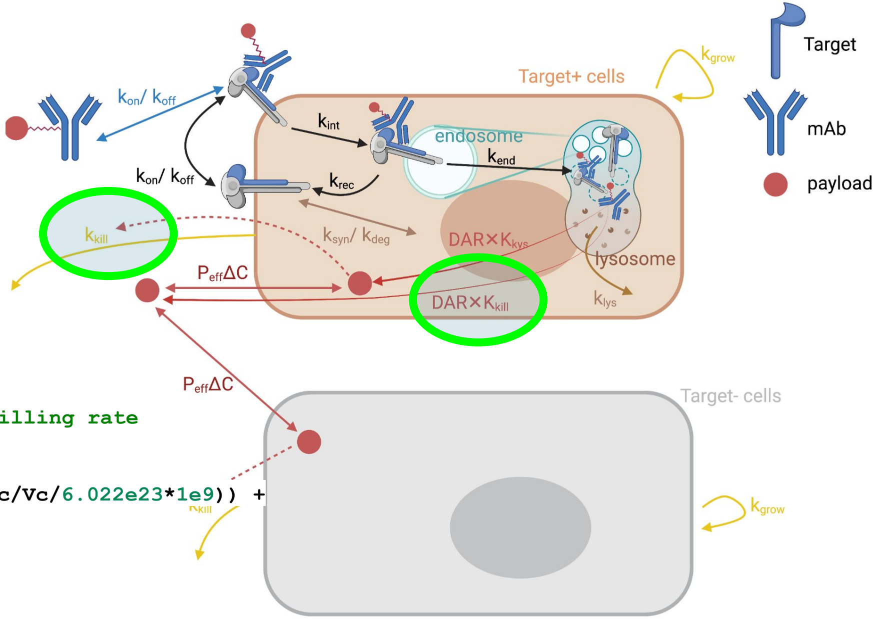
Cell cycle-dependent payload sensitivity data/information is also considered



Model Code: Cell Killing

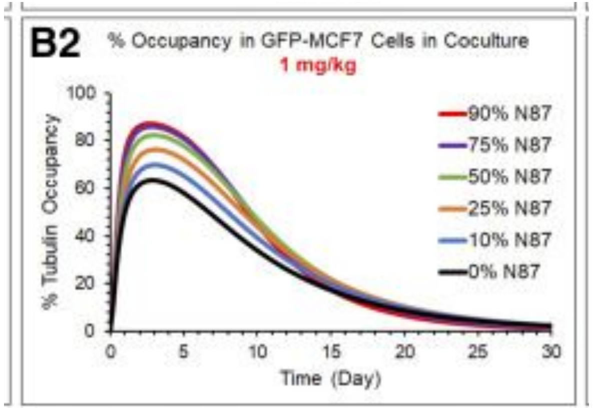
```
function invitroADC(du, u, p, t)
   ...
   Kkill = p[12] # Baseline cell killing/death rate
   # Killing effect model
   EC50_Payload = p[13]  # Killing due to payload
   Emax_Payload = p[14]
   EC50_ADCC    = p[15]  # Killing due to ADCC
   Emax_ADCC    = p[16]
   # Unconjugated payload in cytoplasm
   # Flux = payload escape from lysosome - diffusion from cytoplasm to media
   du[7] = flux_P_l_to_P_c - flux_P_c_to_P_m;
   # Overall cell growth/death
   du[8] = Kgrow_eff*Nc_1 - Kkill_eff*Nc_1;
   # Effective kill rate = baseline death rate + payload killing rate + ADCC killing rate
   # ADCC is assumed to be negligible, here.
   Kkill_eff = Kkill + Emax_Payload *(P_c/Vc/6.022e23*1e9)/(EC50_Payload + (P_c/Vc/6.022e23*1e9)) +
Emax_ADCC*RO/(EC50_ADCC + RO);
   # Transit compartments (non-growing) for cells in process of being killed
   du[9] = Kkill_eff*Nc_1 - Nc_2/tau;
   du[10] = (Nc_2 - Nc_3)/tau;
   du[11] = (Nc_3 - Nc_4)/tau;
   ...
end
```

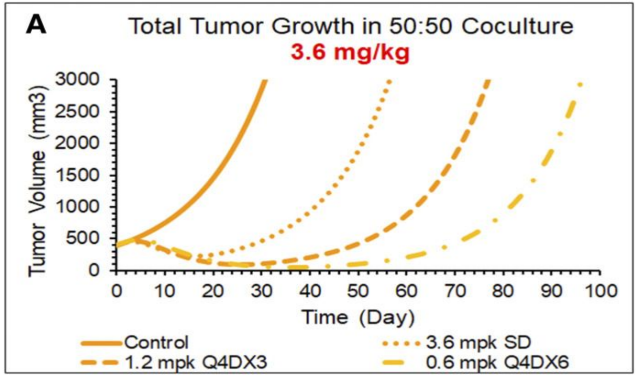
in vitro ADC Modeling



Motivating Example: *in vitro* ADC Modeling using Julia

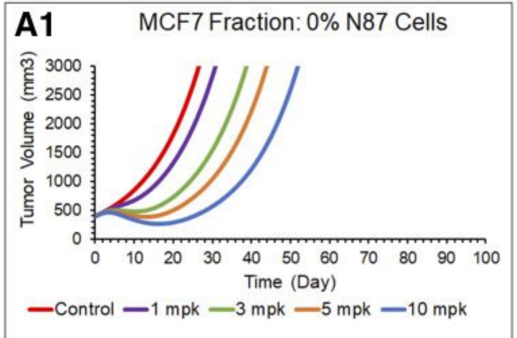
Once the system is drawn and developed (parameterized), you can deploy it to simulate scenarios of interest to the drug development team

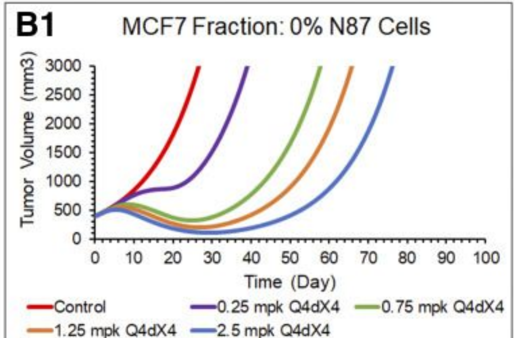




Single Dosing Regimen

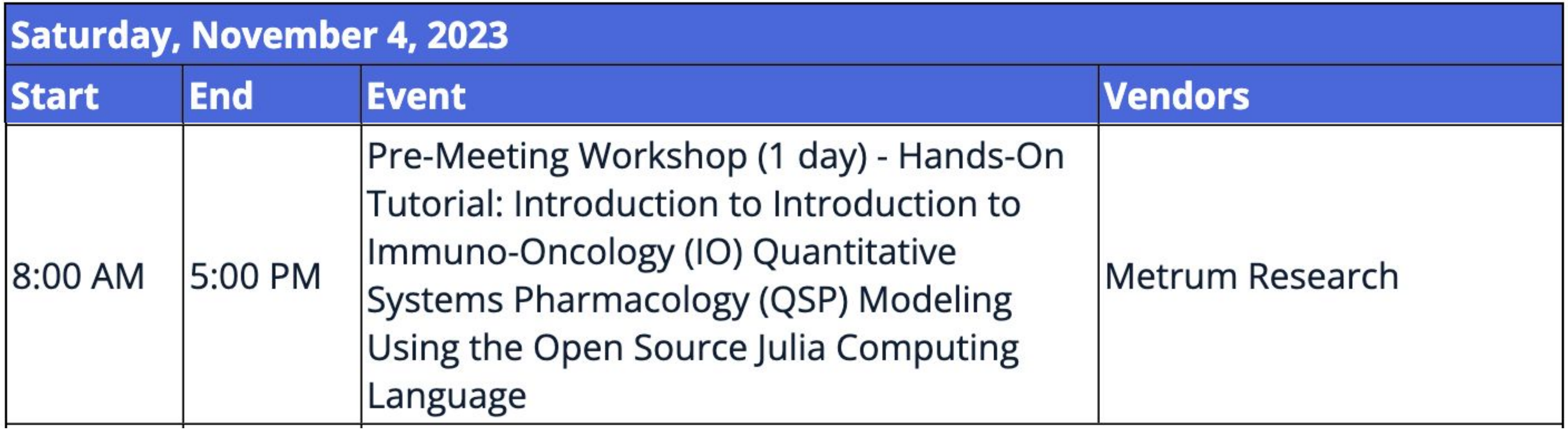
Fractionated Dosing Regimen



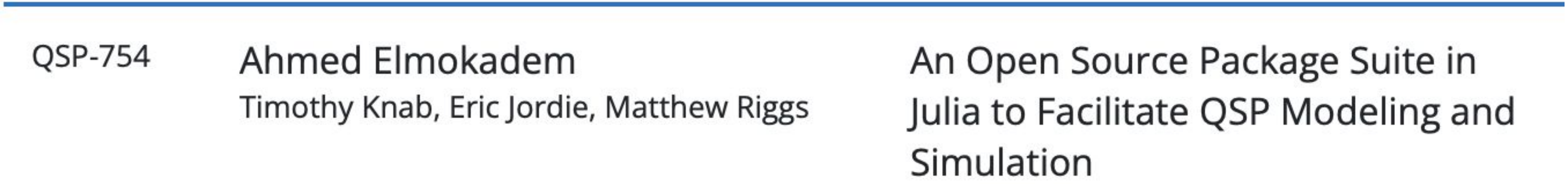


https://jpet.aspetjournals.org/content/early/2020/04/09/jpet.119.262287

Coming Soon: MetrumRG is currently developing open source IO library models and presenting Julia-based content at ACoP 14 (November 5-8th):



Saturday, November 4, 2023			
Start	**End**	**Event**	**Vendors**
8:00 AM	5:00 PM	Pre-Meeting Workshop (1 day) - Hands-On Tutorial: Introduction to Introduction to Immuno-Oncology (IO) Quantitative Systems Pharmacology (QSP) Modeling Using the Open Source Julia Computing Language	Metrum Research



QSP-754

Ahmed Elmokadem
Timothy Knab, Eric Jordie, Matthew Riggs

An Open Source Package Suite in Julia to Facilitate QSP Modeling and Simulation

Coming Soon:
Combined, & parameterized model specs

```
@MRGModel function invitroODE(du, u, p, t)
   @init begin
       # Compartment volumes and surface areas
       @parameter Vm   = 5e-4       # Media volume
       @parameter Vc   = 3.68e-12  # Volume of single cell
       @parameter Sc   = 1.66e-5   # Surface area of a single cell (cm^2)
       @parameter Nc0  = 1.5e5     # Initial number of cells in well
       # Rate constants
       @parameter Kon  = 0.0  # ADC/receptor on rate constant
       @parameter Koff = 1.0  # ADC/receptor off rate constant
...
       # Initial number of cells in well
       @init Nc_1  = Nc0; # All cells are healthy
       @init Nc_2  = 0.0;
       @init Nc_3  = 0.0;
       @init Nc_4  = 0.0;
   end
...
   @ddt Nc_1 = Kgrow_eff*Nc_1 - Kkill_eff*Nc_1;
   # Transit compartments (non-growing) for cells in process of being killed
   @ddt Nc_2 = Kkill_eff*Nc_1 - Nc_2/tau;
...
end
```

Thank you

ahmede@metrumrg.com