Plasma pharmacokinetics

We use a set of ordinary differential equations (ODEs) to describe the drug dynamics in plasma, peripheral tissue and transit compartments [1]:

$$\frac{dC_{t1}}{dt} = -k_a C_{t1}$$

$$\frac{dC_{t2}}{dt} = k_a (C_{t1} - C_{t2})$$

$$\frac{dC_{Pe}}{dt} = Q\left(\frac{C_p}{V_p} - \frac{C_{Pe}}{V_{Pe}}\right)$$

$$\frac{dC_P}{dt} = k_a C_{t2} - Q\left(\frac{C_p}{V_p} - \frac{C_{Pe}}{V_{Pe}}\right) - CL\frac{C_p}{V_p}$$

C_{t1} and C_{t2} are concentrations of antibiotic in first and second transit compartments (mg/kg) respectively, and C_{Pe} and C_P are concentrations in peripheral and plasma compartments (mg/kg) respectively. V_{Pe} and V_P are volumes of distribution for peripheral and plasma compartments (L/kg) respectively. k_a is the absorption rate constant (h^{-1}), Q is the inter-compartmental clearance rate constant between the plasma and peripheral compartments (L/h/kg)
and CL is the clearance rate constant from the plasma compartment (L/h/kg).

The functions that compute these ODEs are BloodFunc::operator() and BloodFuncTwoComp::operator() in grvascular.cpp for 2 transit compartments and 1 transit compartment, respectively.

Flux from plasma to lung

Antibiotics are added to or subtracted from the vascular sources on the *GranSim* grid depending on the concentration difference between the plasma concentration (C_P) and concentration on the lung tissue around the vascular sources on the *GranSim* grid (C_{VSM}) [1]:

$$C_{VSM}(t+\Delta t) = C_{VSM}(t) + p\,A_{VSM}\left(\frac{PC \times \frac{C_p(t)}{V_p} - C_{VSM}(t)}{V_{micro}}\right)\Delta t$$

where C_{VSM} is antibiotic concentration on the grid at the given vascular source (mg/L), C_P is the concentration in blood plasma (mg/kg), V_p is the

volume of distribution for plasma compartment (L/kg), p is permeability (cm/s), A_{VSM} is outside area of the grid micro-compartment (cm^2), PC is permeability coefficient (measure of antibiotic sequestration in the tissue), V_{micro} is the volume of one grid microcompartment in *GranSim* (L) and Δt is time step (s).

The functions that compute the flux are Vascular::solveVascularSources in grvascular.cpp and Vascular::calculateFluxChangeDrugs in grvascular.h.

Tissue pharmacokinetics

Once the drug is in the lung tissue, it degrades, partitions into macrophages and binds to caseum.

Antibiotics are assumed to degrade according to

$$\frac{dC_x}{dt} = -k_{deg,x} C_x$$

where x is intracellular (i) or extracellular (e), $k_{deg,x}$ is the intracellular or extracellular degradation rate constant, and C_x is the intracellular or extracellular antibiotic concentration [1].

We assume cellular accumulation of antibiotics and caseum binding are at pseudosteady state since previous estimated rates of antibiotic uptake are fast relative to diffusion. Intracellular (C_i)
extracellular (C_e) and caseum bound (C_C) concentrations are updated at each diffusion time step based on the total amount of antibiotic in the grid microcompartment where each macrophage is located following diffusion. C_i, C_e and C_c are thus related by:

$$C_e = \frac{A_T}{\left(1 + \left(\frac{1 - f_u^D}{f_u^D}\right) + a\frac{V_{mac}}{V_{micro}}\right)} \quad \text{where} \quad f_u^D = \frac{1}{1 - \left(\frac{1}{D}\right) - \left(\frac{1}{D f_u}\right)} \quad \text{and}$$

$$D = \frac{\text{¿ of killings needed for a total caseation} \in \text{a microcompartment}}{\text{total killings} \in \text{a microcompartment}}$$

$$C_i = a C_e \frac{V_{mac}}{V_{micro}}$$

$$C_c = \left(\frac{1 - f_u^D}{f_u^D} \right) C_e$$

where A_T is the total amount of antibiotic available (the sum of intracellular, extracellular and caseum bound), V_{micro} is the volume of one grid microcompartment, V_{mac} is the volume of a macrophage, a is the cellular accumulation ratio (or intracellular partition coefficient), D is the dilution factor, f_u^D is the measured caseum unbound fraction and f_u is the real caseum unbound fraction [1].

Intracellular degradation is computed in Mac::consumeDrugs in macrophage.cpp, extracellular degradation is computed in AntibioticSimulationGrid::setupDiffusion in antibioticsimulationgrid.cpp and GrDiffusionFFT::setupPMATNoFlux in grdiffusionFFT.cpp, partitioning and caseum binding is computed in the function AntibioticSimulationGrid::partitionAntibioticSteadyState in antibioticsimulationgrid.cpp.

Pharmacodynamics

We calculate the antibiotic killing rate constant (k) using an Emax model (Hill equation):

$$k(C) = E_{max} \frac{C^h}{C^h + C_{50}{}^h}$$

where E_{max} is the maximum killing rate constant, h is the Hill coefficient, C_{50} is the concentration needed to achieve the half maximal killing rate constant ($E_{max}/2$) and C is the drug concentration in a microgrid in *GranSim* [1].

When multiple antibiotics are used and thus present and available on our simulation grid within *GranSim*, we simulate their interaction by adjusting the effective concentration according to their fractional inhibitory concentration (FIC) values predicted by an *in silico* tool, INDIGO-MTB (inferring drug interactions using chemogenomics and orthology optimized for Mtb) [2, 3]. Briefly, we first converted the concentrations of all antibiotics on a small section of the grid (a grid microcompartment or a microgrid) to the equipotent concentration of the antibiotic of the highest maximal killing rate constant (highest E_{max}). For example, if we have *n* antibiotics (drug *i* with the concentration C_i) and drug *m* has the highest E_{max} of all drugs, then we calculate the adjusted concentration for each drug *i* ($C_{i,adj}$), which is the concentration of drug *m* that results in the same antibiotic killing rate constant as drug *i* with the concentration of C_i, with the following equation:

$$C_{i,adj} = \left(\frac{{C_{m,50}}^{h_m} {C_i}^{h_i}}{\frac{E_{max,m}}{E_{max,i}} \left({C_i}^{h_i} + {C_{i,50}}^{h_i} \right) - {C_i}^{h_i}} \right)^{1/h_m}$$

where $C_{m,50}$ and $C_{i,50}$ are the concentration of C_m and C_i at which half maximal killing is
achieved, respectively, $E_{max,m}$ and $E_{max,i}$ are the maximal killing rate constants of drug *m* and
drug *i*, respectively, and h_m and h_i are the Hill coefficients of drug *m* and drug *i*, respectively [4].
Then, we calculated the effective concentration (C_{eff}) as the sum of the adjusted concentrations
of *n* antibiotics that are increased/decreased based on the FIC values to simulate synergistic/antagonistic effects with the following equation:

$$C_{eff} = \left(\sum_{i=1}^{n} {C_{i,adj}}^{FIC} \right)^{1/FIC}$$

where $C_{i,adj}$ is the adjusted concentration of the drug *i* [4]. Then, we used C_{eff} to calculate the antibiotic killing rate constant *k* on that microgrid by using the Hill equation constants of the antibiotic with the highest E_{max}:

$$k\left(C_{eff}\right) = E_{max,m} \frac{{C_{eff}}^{h_m}}{{C_{eff}}^{h_m} + {C_{m,50}}^{h_m}}$$

where $E_{max,m}$, h_m and $C_{m,50}$ are the Hill equation parameters of the antibiotic m, the one with
the highest E_{max} within the regimen [4].

Once we determine the antibiotic killing rate constant *k*, we calculate the probability of killing, *p*, with the following equation:

$$p(k) = 1 - e^{-k}$$

When single drug is present in a microgrid, DrugKill::checkKillExtra and DrugKill::checkKillIntra in drug-kill.cpp compute the killing rate and determine whether bacteria should be killed or not. When multiple drugs are present in a microgrid, DrugKillIndigo::checkKillExtra and DrugKillIndigo::checkKillIntra in drug-kill-indigo.cpp determine whether bacteria should be killed or not.
AntibioticSimulationGrid::computeInteractAdjConc in antibioticsimulationgrid.h computes the adjusted concentrations of each antibiotic and the effective concentration.
AntibioticConcentration::killingRateConstFromEmax in antibioticconcentration.cpp computes the antibiotic killing rate constant based on Hill equation parameters and the effective concentration.

AntibioticSimulationGrid::checkShouldKill in antibioticsimulationgrid.h computes the killing probability and determines whether the bacteria should be killed or not.

References

1. Pienaar E, Cilfone NA, Lin PL, Dartois V, Mattila JT, Butler JR, et al. A computational tool integrating host immunity with antibiotic dynamics to study tuberculosis treatment. J Theor Biol. 2015;367:166-79. Epub 20141209. doi: 10.1016/j.jtbi.2014.11.021. PubMed PMID: 25497475; PubMed Central PMCID: PMCPMC4332617.
2. Chandrasekaran S. Predicting Drug Interactions From Chemogenomics Using INDIGO. Methods Mol Biol. 2019;1888:219-31. doi: 10.1007/978-1-4939-8891-4_13. PubMed PMID: 30519950.
3. Ma S, Jaipalli S, Larkins-Ford J, Lohmiller J, Aldridge BB, Sherman DR, et al. Transcriptomic Signatures Predict Regulators of Drug Synergy and Clinical Regimen Efficacy against Tuberculosis. mBio. 2019;10(6). Epub 20191112. doi: 10.1128/mBio.02627-19. PubMed PMID: 31719182; PubMed Central PMCID: PMCPMC6851285.
4. Cicchese JM, Sambarey A, Kirschner D, Linderman JJ, Chandrasekaran S. A multi-scale pipeline linking drug transcriptomics with pharmacokinetics predicts in vivo interactions of tuberculosis drugs. Sci Rep. 2021;11(1):5643. Epub 20210311. doi: 10.1038/s41598-021-84827-0. PubMed PMID: 33707554; PubMed Central PMCID: PMCPMC7971003.