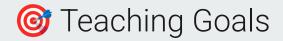


Pharmacokinetics Modeling Course Structural Models



Dr. Matthias König Humboldt-University Berlin Systems Medicine of the Liver koenigmx@hu-berlin.de https://livermetabolism.com

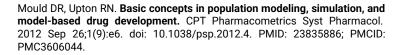


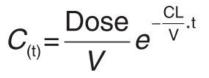
By the end of this section, you should be able to:

- Explain the relationship between Dose, Volume of Distribution (V), and Clearance (CL) using a simple algebraic model.
- 2. **Recognize variability** in pharmacokinetic parameters (such as CL and V) between different drugs and how this impacts drug exposure.
- 3. Develop a **basic understanding of physicochemical properties** (e.g., lipophilicity, molecular weight, solubility) and how they relate to pharmacokinetic behavior.
- 4. Understand the concept and utility of a **parameter scan** in pharmacokinetic modeling.
- 5. **Explore the effects** of changing V, Dose, or CL on drug concentration.

Structural models as algebraic equations

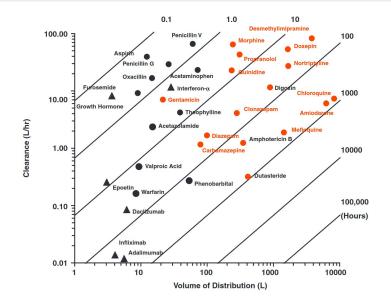
- simplest representation is an algebraic equation representing a one-compartment model
- drug being administered as single intravenous dose
- relationship between
 - independent variable time (t)
 - dependent variable concentration
 (C)
 - C depends on Dose, clearance (CL), and distribution volume (V)

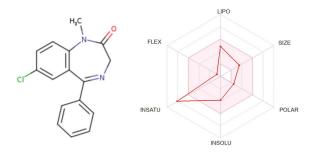




🏃 Physicochemical properties 🏃

- Drugbank lookup (properties & SMILES) <u>https://go.drugbank.com/</u>
- Prediction tool
 <u>http://www.swissadme.ch</u>





State Experimental Properties	Solid	
	PROPERTY	VALUE
	melting point (°C)	138-140
	boiling point (°C)	140 °C
	water solubility	10 mg/mL
	logP	1.18
	рКа	3.5

Structural models as algebraic equations

- Parametrization with clearance (CL) and volume of distribution (Vd)
- Simulation of concentration time course after given Dose

$$C_{(t)} = \frac{\text{Dose}}{V} e^{-\frac{CL}{V} \cdot t}$$
(1)

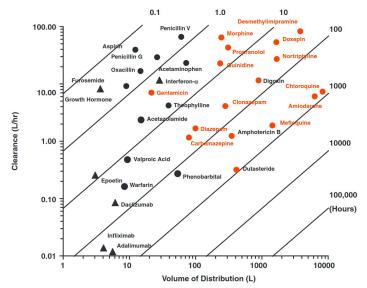


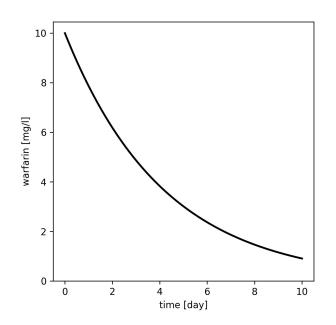
FIGURE 5-8 Clearance (*ordinate*) and volume of distribution (*abscissa*) of selected acidic (*black circle*) and basic (*colored names*), as well as protein (*black triangle*), drugs vary widely. Diagonal lines on the fully logarithmic plot show the combinations of clearance and volume values with the same half-lives (hours). Note that drugs with very low clearance and very large volumes (*lower right-hand quadrant*) are uncommon; their half-lives are often too long for these drugs to be used practically in drug therapy. Note also that large protein drugs have volumes of distribution close to plasma volume (3 L) and that basic compounds tend to have larger volumes of distribution than acids. Digoxin and dutasteride are neutral compounds, while amphotericin B is both a weak acid and a weak base.

Mould DR, Upton RN. **Basic concepts in population modeling, simulation, and model-based drug development.** CPT Pharmacometrics Syst Pharmacol. 2012 Sep 26;1(9):e6. doi: 10.1038/psp.2012.4. PMID: 23835886; PMCID: PMC3606044.

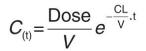
Implementation Example

from matplotlib import pyplot as plt import numpy as np t = np.linspace(start=0, stop=10*24, num=200) # [hr] C = Dose/V * np.exp(-CL/V * t) # [mg/l]f, ax = plt.subplots(nrows=1, ncols=1, figsize=(5, 5), dpi=300) ax.plot(t/24.0, C, label="warfarin", color="black", linewidth=2.0) ax.set_xlabel("time [day]") ax.set_ylabel("warfarin [mg/l]") ax.set_ylim(bottom=0) plt.show()

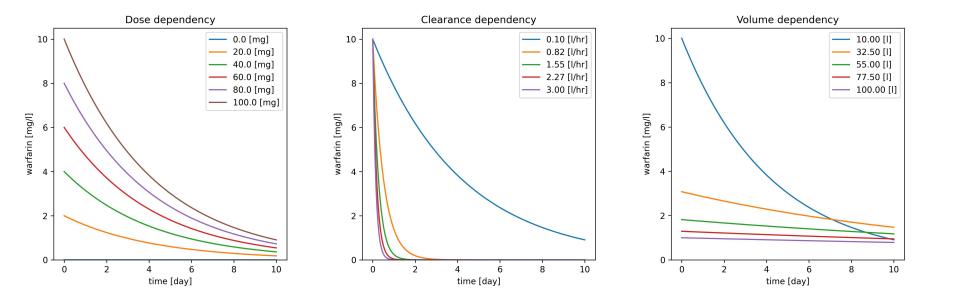
$$C_{(t)} = \frac{\text{Dose}}{V} e^{-\frac{\text{CL}}{V}.t}$$
(1)

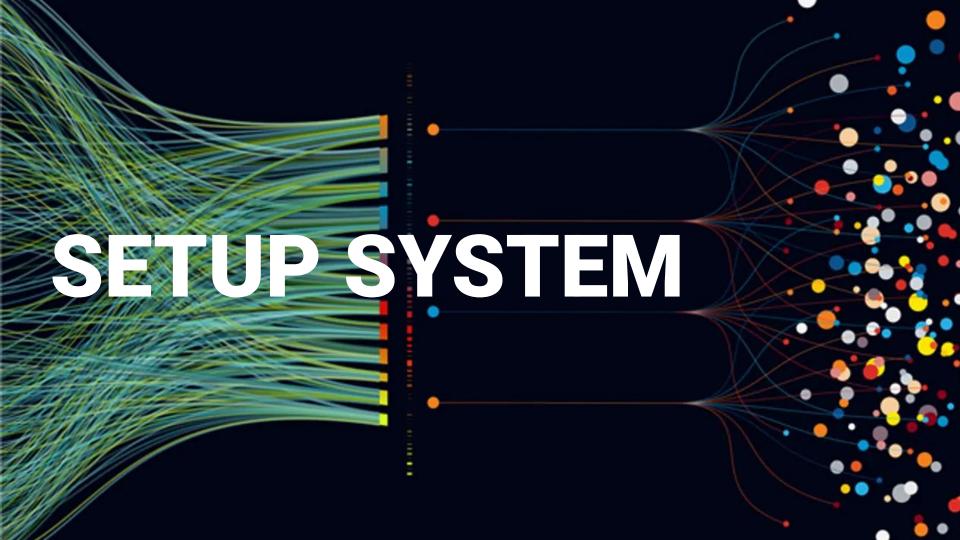


Parameter scans



(1)



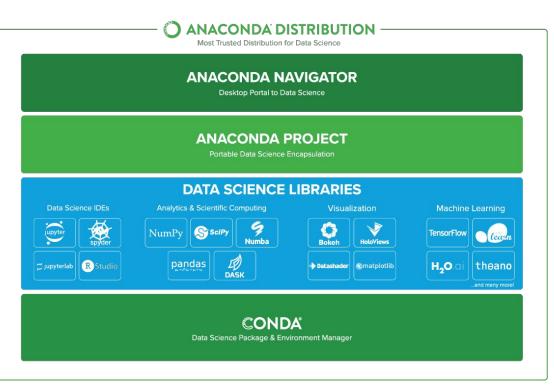


Install Anaconda

Install Anaconda Distribution https://docs.anaconda.com/free/anaconda/#ins tallation

Anaconda® Distribution is a free Python/R data science distribution that contains:

- <u>conda</u> a package and environment manager for your command line interface
- <u>Anaconda Navigator</u> a desktop application built on conda, with options to launch other development applications from your managed environments
- <u>250 automatically-installed packages</u> that work well together out of the box
- access to the <u>Anaconda Public</u> <u>Repository</u>, with 8000 open-source data science and machine learning packages



Setup conda environment

- create conda environment mb19
 - open terminal
 - create environment
 conda create -n mb19
 - install packages
 pip install numpy scipy
 matplotlib pandas

Activating/Deactivating environments

• To see a list of environments: conda env list

vperezg@login1:/home/vperezg>conda env list # conda environments: /prod/apps/conda/3 base /prod/apps/conda/3/envs/bio-computation bio-computation machine-learning /prod/apps/conda/3/envs/machine-learning machine-learning-gpu /prod/apps/conda/3/envs/machine-learning-gpu prosado /prod/apps/conda/3/envs/prosado aiime2-2019.7 /prod/apps/conda/3/envs/giime2-2019.7 quantum-chem /prod/apps/conda/3/envs/quantum-chem

- To load an env: conda activate <env_name> vperezg@login1:/home/vperezg>conda activate bio-computation (bio-computation) vperezg@login1:/home/vperezg>]
- To unload: conda deactivate (bio-computation) vperezg@login1:/home/vperezg>conda deactivate vperezg@login1:/home/vperezg>

Install Spyder

available via the anaconda navigator

- alternative use Jupyterlab
- alternative use
 Jupyter
 notebooks

-يو /Users/juanitagomez/Documents/SpyderDocs/Files /Users/juanitagomez/Local/Dev-Spyder/spyder/spyder/plugins/plots/plugin.py ٠ . B 💡 plugin.py × plugin.py - plots* × plot_example.py × plugin.py - ipythonconsole Name A Type Size Value V C Plots foo aet name /Users/Documents/spyder/spyder/tests/ test_dont_use.py filename get_description # Licensed under the terms of the MIT License # (see spyder/__init__.py for details) get_icon True register my_set set Plots Plugin. unregister 6.46567886443 switch_to_plugin current_widget from gtpy.OtCore import Signal tuple ('abcd', 745, 2.23, 'efgh', 70.2) add shellwidget # Local imports from spyder.api.plugins import Plugins, SpyderDockablePlugin from spyder.api.translations import get_translation from spyder.plugins.plots.widgets.main_widget import PlotsWidget m remove_shellwidget thisdict dict {'brand':'Ford', 'model':'Mustang', 'year':1964} set_shellwidget plot_example.py % Unnamed cell 1 Array of int64 (2,) [1 2] F generate polar plot _ = get_translation('spyder') timedelta 1 2 days, 0:00:00 generate_dem_plot - 🗾 main class Plots(SpyderDockablePlugin): olugin.pv Help Variable Explorer Files Plots plugin. IPythonConsole NAME = 'plots' REQUIRES = [Plugins.IPythonConsole] TABIFY = [Plugins.VariableExplorer, Plugins.Help] WIGET_LLASS = PlotsWidget init__ 8 . P update font _apply_gui_plugin_sett CONF_SECTION = NAME apply_mpl_plugin_set. CONF_FILE = False DISABLE ACTIONS WHEN HIDDEN = False apply_advanced_plug apply pdb plugin set apply_plugin_settings_ def get_name(self): return _('Plots') apply_plugin_settings toggle_view def get description(self): return _('Display, explore and save console generated plots.') get_plugin_title get_plugin_icon def get_icon(self): return self.create icon('hist') get_focus_widget Closing_plugin def register(self): refresh plugin ipyconsole = self.get_plugin(Plugins.IPythonConsole) get_plugin_actions register_plugin get_clients get_focus_client aet current client Plots IPython console History - 🔟 get_current_shellwidge 9 LSP Python: ready @ conda: spyder_dev (Python 3.7.10) ¥ master Line 1, Col 1 UTF-8 LF RW Mem 57%

Spyder The Scientific Python Development Environment

Jupyterlab instance

https://itbportal.biologie.hu-berlin.de/koenigcourse/hub/login