Pharm Res
https://doi.org/10.1007/s11095-021-03065-1

RESEARCH PAPER

Scipion PKPD: an Open-Source Platform for Biopharmaceutics, Pharmacokinetics and Pharmacodynamics Data Analysis

C. O. S. Sorzano[1,2] • Y. Fonseca-Reyna[1] • M. A. Pérez de la Cruz-Moreno[2,3]

Received: 9 March 2021 / Accepted: 21 May 2021

ABSTRACT

Purpose Biopharmaceutics examines the interrelationship of the drug's physical/chemical properties, the dosage form (drug product) in which the drug is given, and the administration route on the rate and extent of sys- temic drug absorption. Pharmacokinetics is the study of the movement of drugs in the body. It uses mathematical models to evaluate the movement of absorption, distribution, metabolism, and excretion (ADME) within an organism. Finally, Pharmacodynamics is the analysis of how these drugs af- fect that organism. Pharmacokinetics data normally comes in samples over time of the drug concentration either in plasma or in some specific tissue. Similarly, pharmacodynamics data comes normally in samples over time of some quantity of interest (biophysical quantity like temperature, blood pres- sure, etc.). The data is submitted to a non-parametric analysis, in which a description of the observed data is reported (e.g., the Area Under the Curve), or to a parametric analysis by fitting a model (normally based on differential equations) so that prediction about future events can be made. This paper aims to introduce Scipion PKPD, an open-source platform for data analysis of this kind in the three domains (Biopharmaceutics, Pharmacokinetics, and Pharmacodynamics). The platform implements the most popular models and is open to new ones. The platform provides almost 100 different high-level operations that we call protocols.

Methods We have developed a Python module integrated into the work- flow engine Scipion. The plugin implements the numerical analysis and meta- data handling tools to address multiple problems (see Suppl. Material for a detailed list of the tasks solved).

Results We illustrate the use of this package with an integrative exam- ple that involves all these areas.

Conclusions We show that the package successfully addresses these kinds of analyses. Scipion PKPD is freely available at https://github. com/cossorzano/scipion-pkpd.

KEY WORDS biopharmaceutics · data analysis · pharmacodynamics · pharmacokinetics

INTRODUCTION

The combination of drug release from its dosage form, followed by dissolution and permeability, explains how drug molecules become available for their internalization (absorption) in an organism (1). Once the drug is inside the body, it is distributed through the blood system to the different organs and tissues (some molecules preferably stay in the blood system with a small fraction of them going to their target; while some others quickly accumulate in some tissue, for instance, adipocytes, and then released from them at a different rate). Finally, the drug is metabolized into a different compound (the liver is very efficient in this task, although this process may occur in other organs) or excreted (through urine, bile, sweat, saliva, breast milk, or as a gas in the lungs). As a result of all these processes, the drug's concentration in different fluids (intestinal fluid, blood plasma, urine, etc.) or tissue (fat, muscular, etc.) changes over time (2). All these processes are generally known as LADMET (liberation; adsorption; distribution; metabolism; excretion; and toxicity).

Pharmaceutical companies are highly interested in characterizing all these processes. They have to accurately determine the time evolution of the concentration of their products and relate these time profiles to the desired (therapeutic) and

✉ C. O. S. Sorzano
coss@cnb.csic.es

[1] National Center of Biotechnology (CSIC), c/Darwin, 3. Campus Universidad Autónoma de Madrid, 28049 Cantoblanco, Madrid, Spain

[2] Kinestat Pharma, Madrid, Spain

[3] Chemo Group, Guadalajara, Spain

Published online: 23 June 2021

undesired (toxic) drug effects (3). For this characterization, samples of drug concentration are taken at several time points of the same experimental unit (an experimental unit can be a dissolution vase, a research animal, a person, or whatever other unit that makes sense in the analytical context). Several experimental units' measurements are then gathered to describe the expected variability within a population of similar units. The analysis of these samples can be: 1) purely descriptive, referred to as non-parametric or non-compartmental (3), with parameters such as the C_{max} and t_{max} (maximum observed concentration and time at which this concentration was observed), AUC (Area Under the (concentration) Curve), MRT (Mean Residence Time of a drug molecule), etc.; or 2) descriptive and predictive, also referred to as parametric or compartmental (3), in which a model is fitted to the observations so that the response under varying conditions can be predicted from the parameters estimated from each experimental unit. The most common models are based on differential equations that describe the time evolution of the drug concentration at the different tissues. Models based on differential equations can be generic (first or second order, linear time-invariant systems) or based on the physiology of the organism under study (in this case, the parameters of the differential equations are related to blood flows, connectivity between the different tissues, drug affinity of each one of the tissues, etc.) (4).

Several platforms allow the data analysis of this kind of samples. Among the most established, we can find SAS (a generic data analysis platform with specific routines developed to handle this kind of data and models), Phoenix from Certara, L.P., and Nonmem from Icon, Plc. (5) Less established software programs address specific kinds of analyses: PKBugs (Bayesian analysis of pharmacokinetic profiles) (6), Matlab Simbiology Toolbox (7–9), R libraries, each focusing on a specific problem (ncappc, PK, clinPK, ...a full list can be obtained at https://cran.r-project.org/web/views/Pharmacokinetics.html), S-Plus (10, 11), Excel (12, 13), etc. The interested reader is referred to the following reviews on software for these domains (14, 15). We can highlight four features of all these solutions:

1. Programming skills: most solutions based on SAS, Nonmem, Matlab, or R require programming skills from the user, which allow him/her to be very flexible about the analysis performed but also imposes an important entrance barrier.
2. Limited scope: on the other extreme, those solutions with a simple user interface are very much focused on the solution of a particular problem, and they cannot be applied to any other problem.
3. Closed solutions: apart from the R packages and SAS routines, all other solutions are closed in the sense that their source code is private and they do not allow verifying the correctness of the code, identification of potential malfunctions, the inadequacy of the assumptions to our experimental conditions, or modification of the behavior of the code to address slightly different situations.
4. License price: except for the R packages and some particular software pieces, most of the solutions above involve expensive license costs ranging from 2000$ to 10,000$ for a single user, and some of them have to be paid yearly.

This article introduces Scipion PKPD, a free, open-source platform based on the workflow engine Scipion (16). This engine has been used by thou- sands of image processing projects in cryo-electron microscopy (http:// scipion.i2pc.es/report_protocols/scipionUsage) and has hundreds of users worldwide. The workflow engine allows the integration of independent software packages in the form of plugins. The user interface is friendly and based on a workflow graph that grows as new data analysis steps (called protocols within Scipion) are performed (see Fig. 1). Each protocol has a standardized form (the standardization lowers the user learning curve, see Fig. 2), receives several inputs, and produces some outputs. Input and out- put objects are semantically understood by the workflow engine so that the user does not need to access files directly. At each point, he/she is only of- fered those objects that match the operation he/she is planning to do. There are general visualization tools for each type of object recognized by the sys- tem and specific visualization tools for analyzing the output of any particular protocol (see Fig. 3). Each protocol is executed in its own directory, where all the information related to that protocol is kept. Protocols cannot write in the directories of other protocols. Additionally, Scipion provides a layer of reproducibility (except for steps with random initializations), traceability, and integration in High-Performance Computers (HPC). Each package can be developed in any programming language, but the integration must be done in Python. Scipion PKPD is purely developed in Python, and consequently, Scipion PKPD can be deployed in any operating system capable of running Python 3 with virtual environments (virtualenv) and git, although its most natural environment is Linux. Scipion has been successfully deployed in computing clouds (private, like Amazon, or public, like the European Fed- erated Cloud). The release cycles of the workflow engine (Scipion) and the plugins (Scipion PKPD) are totally independent. Both can be updated at any moment without the need for a new release of the other.

PKPD calculations are particularly well suited to a workflow calculation. For instance, we would start by importing the data for analysis (Step 1). The output of Step 1 is then the input of a change of unit protocol (Step 2). The output of Step 2 is then submitted to a non-compartmental analysis in which we measure the Area-Under-the-Curve or the Mean-Residence-Time (Step 3). At the same time, the output of

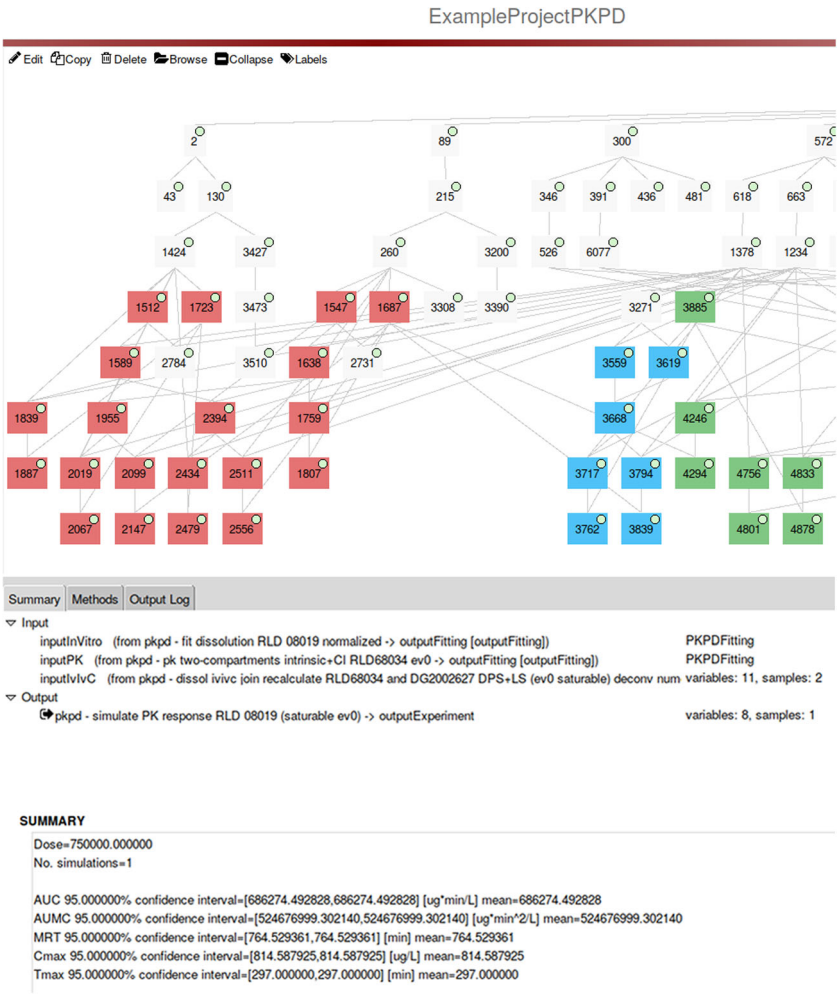

Fig. 1 Example of the appearance of a Scipion project. The upper panel shows the data analysis graph at the moment (the color of each box can be defined by the user according to tags that can be created along the analysis). The lower panel shows the inputs, outputs and a summary of the selected protocol.

Step 2 can be submitted to two different protocols, one to estimate the elimination rate (Step 4) and another one to estimate the absorption rate (Step 5). The estimates of these two rates can be the input of a compartmental analysis that explains the observed samples (Step 6). This sequence of steps can be arbitrarily extended and be as complex as needed.

At the moment, Scipion PKPD has about 100 different protocols, each one implementing a high-level task like performing a non-compartmental analysis of a profile corresponding to an extravascular bolus, compartmental analysis, or IVIVC (*In vitro-in vivo* correlation). Other support functions in- clude importing and exporting data (most importantly from CSV and Excel files), splitting and joining datasets, adding or removing samples or vari- ables, etc. These protocols are divided into logical subgroups (experiments, statistics, biopharmaceutics, non-compartmental and compartmental phar- macokinetics, populations, physiologically based pharmacokinetics, pharma- codynamics, joint pharmacokinetics and pharmacodynamics analysis, and dose escalation). Each protocol has unit tests guaranteeing its correct func- tion. It is recommended to run these unit tests after installation to validate the system's correct functioning (for instance, the installation tests of some of the commercial programs above are sold separately and are rather costly). Additionally, as an open-source tool, anyone can join the project at GitHub (https://github.com/cossorzano/scipion-pkpd) and add the protocols of his/her interest.

GENERAL INTRODUCTION TO SCIPION

Scipion is a workflow engine written in Python that runs as a desktop application, typically on a Linux-like operating system. A workflow is a sequence of concatenated high-level operations, called protocols, in which the outputs of one or more protocols are the inputs of the next protocol. Scipion was designed with several goals in mind:

- Adding a traceability and reproducibility layer to complex calculations. For doing so, all parameters are recorded, any protocol or workflow can be exactly recalculated in the same conditions as it was originally executed; this is an

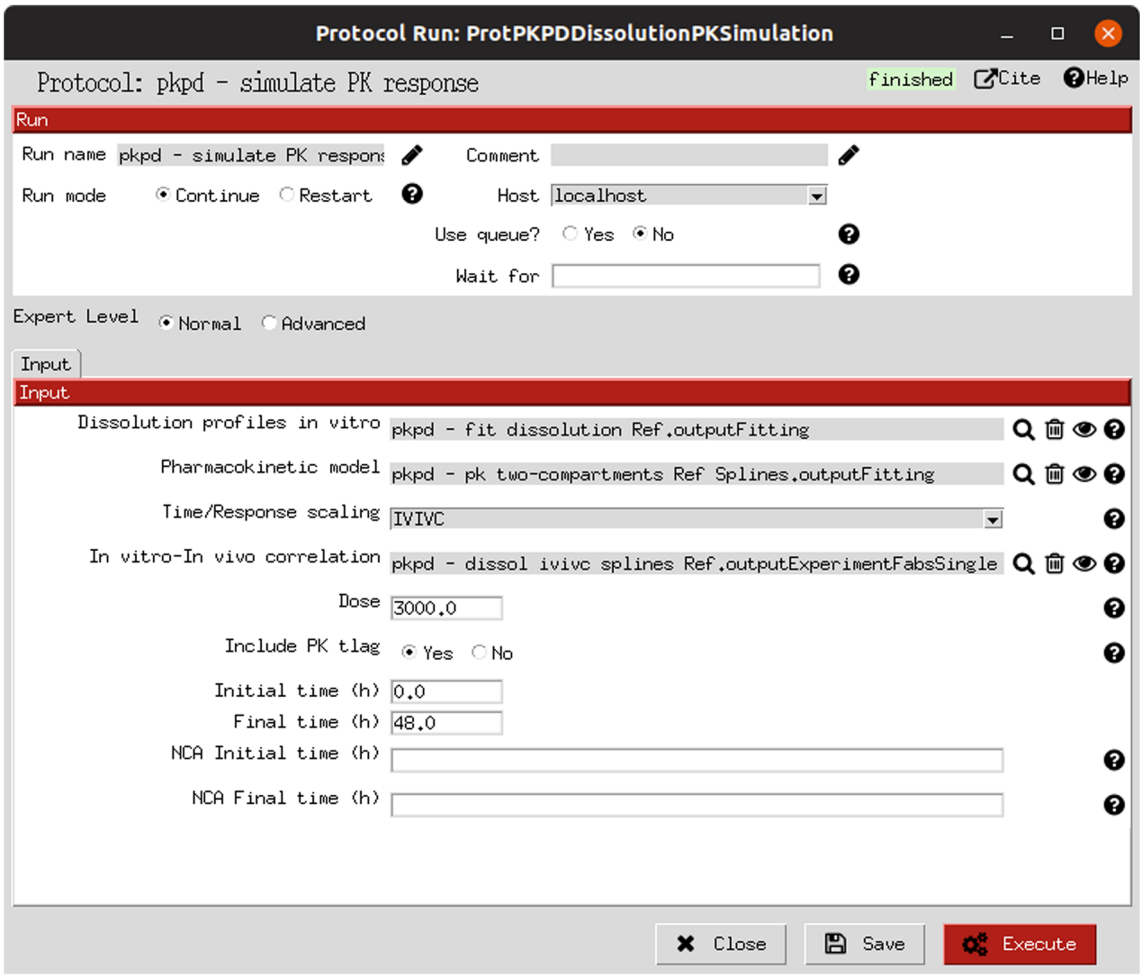

Fig. 2 Example of form, in this case for the simulation of the pharmacokinetic response of a person after receiving a dose with a release specified by an *in vitro* dissolution profile and elimination based on previous PK parameters as typically used for IVIVC analysis. The *in vitro-in vivo* correlation is used to perform this simulation. Each field in the form has an object type, and Scipion only allows choosing objects of that type. The user does not need to deal with specific files; what is more, Scipion PKPD checks the integrity of the files by adding an MD5 code that verifies that the file has not been modified since its creation (this is absolutely needed for traceability and good laboratory practices in pharmaceutical companies).

important contribution to the Good Laboratory Practices required in pharmaceutical companies.

- Allowing interoperability between software packages. For doing so, Sci- pion defines an internal representation of each data type, mostly the metadata associated with it. Protocols typically start with a convertInput step that transforms the metadata from Scipion's representation to the internal representation needed by the software underlying the pro- tocol to be executed. The last step of the protocol is typically a createOutput step that transforms back from the internal represen- tation of the package to Scipion's internal representation. Scipion's data types model is rather flexible and allows storing in the metadata fields that are unknown to Scipion but have a specific meaning within a particular package. In this way, software packages can interoperate using the common fields known to Scipion. Still, a particular software package can fully exploit its own potential if several protocols of the same package are concatenated.

An important consequence of interoperability is that the barrier to use several software programs to perform a difficult data analysis task is significantly lowered for the final user. Without these automatic con- versions, the final user would have to write scripts that allow the con- version of a package's metadata onto the metadata of another package, handling all the data conventions (e.g., units) in between.

- Reducing the programming burden to developers and providing a sim- ple mechanism to access common tasks like defining a form to input the parameters, validation of the input, offering wizards to choose parame- ters and viewers to explore the protocols' results, or launching jobs on a High-Performance Computing environment.

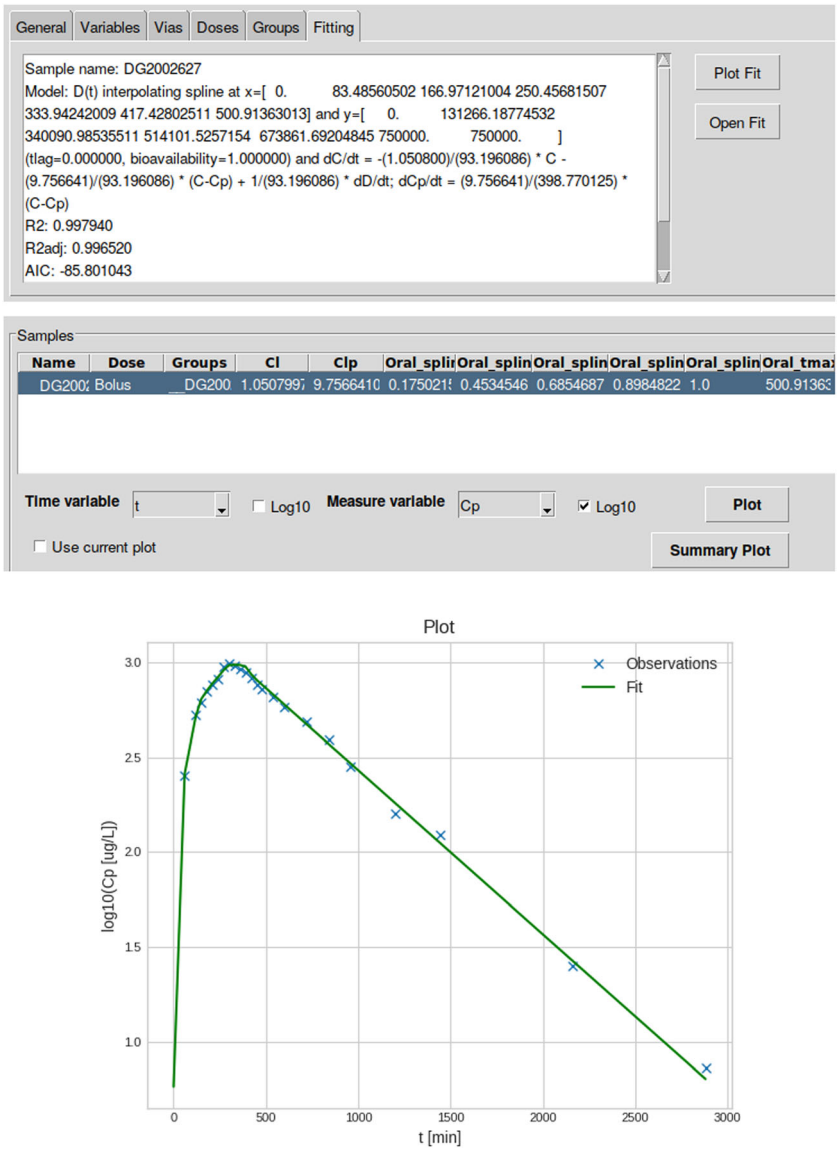

Fig. 3 Each data data type and protocol can implement its own visual analysis tool. In this figure we show the one associated to fitting steps. The fitting parameters and fitting quality measures are displayed in a window, and the fitted curve can be plotted and interactively manipulated to highlight specific parts of the fitting.

Scipion offers a Graphical User Interface based on Tkinter, and the com- munication between protocols is performed at the level of Scipion objects. Communication through files in the filesystem is discouraged because this requires a higher level of knowledge from the user, and it is much more error- prone. Ultimately, Scipion's objects point to files in the filesystem, but the specific details of filenames and file formats are transparent to the final user, reducing the possibilities of errors.

A protocol is composed of several steps that can be executed sequentially or in parallel. Each step can perform a calculation either in Python or in any other language through an appropriate binding library (it already has an extensive binding to C++ libraries) or system calls (command lines calling specific programs in the system shell). All command lines and all the output returned by the programs or routines called are recorded in a protocol log file that can be easily accessed through the graphical user interface. The protocol often serves as an intermediate layer between the final user and the command lines by automatically generating these command lines depending on the user's input. This automatic process prevents frequent errors of the manual construction of command lines.

The workflow engine takes care of the steps that can be computed at a particular time, evaluates the computational resources available (CPU, GPU, queues, etc.), and launches the steps. Several protocols can be scheduled and executed together in a streaming mode, in which data arrives sequentially, and the corresponding steps are executed as new data appears.

Although most users prefer the interaction with Scipion through the graphical interface, Scipion also allows programmatic access. Actually, the tests described in the Results Section are written using this second option. In this way, data analysis can be automated if needed.

Protocols using the same software package are normally bundled into a single plugin. Plugins can be installed through

Pypi, so they are actually Python modules (more than 60 plugins exist for Scipion, https://pypi.org/ search/?q=scipion, most of them in the domain of cryo-electron microscopy image processing, protein atomic modeling, and virtual drug screening). The reader interested in Scipion is referred to Scipion's documentation web page (http://scipion.i2pc.es) and to (16). Scipion has a mechanism to discover new versions of its plugins to inform the user when an updated version of the different installed plugins is available. See the instructions to install Scipion PKPD at https://github.com/cossorzano/scipion-pkpd/wiki/ Install-Scipion3-PKPD-in-Ubuntu.

Developers interested in the appearance of Scipion PKPD protocol may have a look at the Wagner-Nelson deconvolution implemented in https:// github.com/cossorzano/scipion-pkpd/blob/master/pkpd/protocols/protocol_pkpd_dissolution_wagner_nelson.py and Scipion's general documentation for developers (https://scipion-em.github.io/docs/docs/developer/developers.html).

PKPD PROTOCOLS

In this section, we describe the main domains addressed by the Scipion PKPD plugin. In the supplementary material, we provide a full list of all protocols available, their domain, and a short help describing their function.

Experiments

A central object in Scipion PKPD is the concept of Experiment. An experiment is a collection of acquired data, not necessarily time-dependent, related to a set of experimental units (dissolution vessels, animals, persons, etc.). Each experimental unit is referred to as a sample. The main piece of information of an experiment are the collected measurements of variables X_1, X_2, ... For time-dependent collections, one of the variables must be time. The metadata associated with the samples is also contained in the Experiment. In particular, we must define each of the variables, its units, the experimental units, dosage, and route of administration (if it applies), ... Each sample can have labels associated with his/her gender, weight, height, or whichever other pieces of information relevant for the analysis. We can also define groups of samples. For instance, we may have a volunteer for which we have drug concentration samples in blood plasma and urine at different time points. The person acts as a group, and the two sets of observations belong to this group.

The plugin includes protocols to merge experiments, filter samples or measurements that fulfill some conditions or do not fulfill some conditions, create labels, create new variables by doing any mathematical operation on the information available, and change the units of any of the variables, etc. These operations can be thought of as administrative operations that prepare the experiment for posterior analysis.

Experiments are persisted as text files with a file format easy to interpret by humans (see the Suppl. Material for two examples of these files). In this way, any possible mistake can be readily identified by the user. To preserve the files' integrity, each file is accompanied by an MD5 hash code that summarizes the file content and the timestamp of the file so that the system can detect if there has been any external manipulation of the file. This is absolutely needed to guarantee Good Laboratory Practices required by the pharmaceutical industry.

Statistics

There are algorithms to compute an experiment's average sample and compute a statistical summary (confidence intervals, percentiles, mean, me- dian, standard deviation, etc.). Hypothesis tests on the mean of a variable can also be compared between two groups or two experiments. There are also hypothesis tests on the equality of distribution of a variable in two groups or experiments (Kolmogorov-Smirnov test). The Mahalanobis distance be- tween two groups of measurements can also be computed. Finally, there is a protocol that allows regressing any variable on any other.

In Vitro Dissolution Test

A dissolution test is a very broad assay used in the pharmaceutical in- dustry with a role in quality control and the prediction of absorption for drugs belonging to a specific BCS (Biopharmaceutics Classification System) class. For instance, it can be the driving force controlling the entrance of the drug into the body. This is the first process that the drug experiences in its path to a therapeutic target, and it controls the entry of the drug into the body. Pharmaceutical companies carefully design this release to accomplish therapeutic time profiles. There are many experiments of drug dissolution *in vitro* that try to mimic the *in vivo* dissolution by stirring the drug tablet in a controlled environment. Scipion PKPD offers several models to fit the observed samples: zero, first or fractional order dissolution; Weibull and dou- ble Weibull dissolution model; Higuchi, Korsmeyer-Peppas, Hixson-Crowell, Hopfenberg, Hill, and Makoid-Banakar dissolution models; and B-splines dis- solution profiles with up to 10 knots. This plethora of models offer a wide variety of physically-based and very flexible models to address all kinds of experiments.

Non-compartmental and Compartmental Pharmacokinetics (PK)

Pharmacokinetics studies how drug concentration evolves in a particular tissue or, more typically, in blood. This analysis can

be based on a differ- ential equation model (compartmental analysis because the body is divided into compartments in which the drug is distributed) or on more descrip- tive analyses (non-compartmental) based on the description of the observed profile.

Most regulatory guides are focused on non-compartmental analyses. These analyses focus on the maximum concentration observed and the time of this observation, the calculation of drug exposure as the Area-Under-the-Curve (AUC), Mean Residence Time of the drug, ... normally after administering a single drug bolus either intravenously or orally. These parameters can be de- rived directly from the observations or a smoothed version of the observations by fitting them with decaying exponential functions. Scipion PKPD provides protocols that allow performing these fits, and all these measurements either from the smoothed curves or the direct observations. Additionally, this kind of analysis allows an estimation of the absorption and elimination rate. If two samples, one intravenous and one oral, are available, then a protocol allows estimating the bioavailability (that is, the percentage or fraction of drug that scapes the first-pass metabolism).

Compartmental pharmacokinetics considers the observed measurements as time samples of a curve that is the solution of a differential equation mod- eling the distribution and elimination (metabolization or excretion) of the drug. Scipion PKPD offers protocols for estimating the differential equation parameters for one (central) or two (central and peripheral) compartments. These are, by far, the two most commonly used models for most compartmen- tal analyses. Scipion PKPD provides these two models with many variants (drug concentration in blood, urine, or both), varying (also known as intrin- sic and related to the drug's metabolism), or autoinduced clearance. One of the protocols allows simultaneous fitting of the parameters to sets of samples from intravenous and oral doses. Scipion PKPD Pharmacokinetics analy- sis follows a signal processing approach in which the dosage regimen can be arbitrary and composed by bolus at any times intermixed with infusions (17, 18). This implies a huge generalization with respect to the standard approaches followed in the field that only consider a single bolus, a single infusion, or repeated bolus at regular intervals. The main difference is that Scipion PKPD numerically solves the differential equation using the dosage regimen as input. Simultaneously, the standard approaches take the analyt- ical solution of the differential equation for specific dosing regimes. There is obviously a time penalization for numerically solving the differential equation that is translated into higher execution times. This is further increased by the global optimization algorithm used in Scipion PKPD (Differential Evolution (19)). The standard approach to pharmacokinetics parameter identification uses some sort of local optimizers. This is very limiting as the user needs to know a suitable initial point for the optimization, resulting in local min- ima and frustrated users who cannot find a suitable way to find the model's parameters. This is particularly difficult when multiple time profiles have to be analyzed since the same starting point may not be valid for all the profiles, some of them requiring a significant amount of manual intervention. The main advantage of compartmental pharmacokinetics is that once the differential equations' parameters are found, they simulate the individual's response to any arbitrary dosing plan. Scipion PKPD also offers a protocol to perform these simulations.

In Vivo Drug Absorption

We may infer the fraction absorbed of the drug *in vivo* if a compart- mental pharmacokinetic analysis is performed. We can distinguish between the absorption, distribution, and elimination parts of the equation in these models. We can then integrate the absorption into a time profile that esti- mates the fraction absorbed within the individual. This process is referred to as deconvolution. Scipion PKPD provides protocols for performing this de- convolution for monocompartment (Wagner-Nelson) and two compartments (Loo-Riegelman) models. These deconvolutions are model-based. Additionally, two protocols allow numerically estimating these dissolution profiles employing the impulse response of the pharmacokinetic model (either in real space or Fourier space).

In Vitro-In Vivo Correlation (IVIVC)

This type of analysis tries to determine the time scaling between *in vitro* dissolutions and fraction absorbed *in vivo*. In this way, an individual's drug concentration can be predicted from the *in vitro* dissolution profile resulting in significant savings in experiments with animals or humans during the de- sign of the drug's dissolution profile. Scipion PKPD provides protocols for computing Levy plots (a tool widely used in this kind of analysis) and com- puting the time scaling function in various ways (with many different time dependency functions). One of the main differences between Scipion PKPD and other software available is that Scipion PKPD allows using B-splines for the time transformation, resulting in very flexible time dependencies. Some protocols allow estimating the quality of the IVIVC and simulating individ- uals' time response to any dose regime.

Populations

Scipion PKPD provides support to the management of the PK parame- ters of populations. Given a population of individuals, we can collect all the different individuals' PK parameters into a single object. Additionally, we can also determine the variability of the PK parameter estimates by boot- strapping. This would give us alternative PK parameters of the same time profile, resulting in a "pseudo-population" of PK

parameters. Having pop- ulations of PK parameters allows simulating the response of a population of people, rather than a single individual, to a particular dosing regime. The use of populations also enables defining allometric transformations that allow transferring PK parameters from a few species onto another species for instance, we can predict the PK parameters of humans based on the PK parameters of mice, rats, and dogs). There are also protocols to manage pop- ulations (merging and splitting populations, filtering individuals considered outliers, or not fulfilling some property).

Physiologically Based Pharmacokinetics (PBPK)

At the moment, Scipion PKPD provides only one protocol related to physiologically based PK analysis. This protocol simulates the liver's drug concentration, taking into account the liver blood flow, liver volume, and intrinsic clearance.

Pharmacodynamics (PD)

Pharmacodynamics studies the effect of this drug on a variable of interest (for instance, blood pressure or body temperature) as a function of time. This analysis amounts basically to a regression of the variable of interest on the drug concentration measurements. Scipion PKPD provides protocols to perform this regression with several models (polynomials, log-linear, several sigmoid and logistic functions, Richards, Gompertz, Morgan-Mercer-Flodin, Weibull, and Hill models). A protocol allows estimating the variability of the fitting parameters by bootstrapping and another one that permits simulating different PD responses based on possible drug concentrations.

Joint Pharmacokinetics and Pharmacodynamics Analysis (PKPD)

These protocols simultaneously estimate the PK and PD parameters for one or two compartmental PK models and specific PD models. For these es- timations, we need measures of the drug concentration and the effect variable of interest.

Dose Escalation

In Phase I clinical trials, the goal is to determine the dose that causes light toxic effects on healthy individuals. The trial starts by administering very low doses that are gradually increased until the probability of observing toxicity (light side effects) reaches a predefined level. A protocol follows the most widely used 3 + 3 rule and suggests the researchers the stopping dose.

TECHNICAL DESCRIPTION

The source code of Scipion PKPD is freely available at github (https://github.com/cossorzano/scipion-pkpd). Every commit is analyzed by SonarCloud (https://sonarcloud.io/dashboard?id=cossorzano_scipion-pkpd) so that the developer has immediate feedback on the quality of his/her code.

As one of the main features of the plugin for developers, Scipion PKPD has a class hierarchy that makes the addition of new biopharmaceutics, PK, and PD models very easy (https://github.com/cossorzano/scipion-pkpd/ tree/master/pkpd/models). The developer must concentrate only on de- scribing the model's parameters, units, and mathematical behavior. Then, the new model has to be added as a choice in the corresponding model fitting protocol, and automatically it would be available through the Scipion menu.

Although not compulsory, it is highly recommended that all protocols in a Scipion plugin have a unit test. Scipion PKPD has followed this recommen- dation, and there are unit tests for all its protocols. Actually, rather than isolated protocols, we have followed an approach to testing full workflows, each solving a PKPD problem. We verify that the workflow is fully executed (there is no syntax error) and that the estimated parameters are effectively the expected ones (there is no calculation error). These unit tests are a precious tool to guarantee the correctness of a particular installation. They are also regularly executed for the development version in a machine running buildbot (http://scipion-test.cnb.csic.es:9980/#/builders) so that any programming error can be easily identified.

EXAMPLE

We now give an example of the definition of a data analysis workflow involving biopharmaceutics and pharmacokinetics. The example can be re- produced running.

scipion test pkpd.tests.test_workflow_levyplot3

The example shows an *in vitro-in vivo* correlation. The dissolution data of 12 vessels and the drug concentration in the blood plasma of 12 individuals are imported. Note that these two datasets are independent, and there is no correspondence between vessels and individuals. The data values for the in-vitro and in-vivo experiments are given in the Suppl. Material. We show them using the internal representation of the experiments in Scipion. We refer to this file format as the Scipion PKPD file format. It can be appreciated for its simplicity and its human-friendly appearance.

We fitted a Weibull model to the dissolution experiments (Biopharmaceu- tics) and a first-order absorption, one-compartment model to the drug con- centration in blood profiles (Pharmacokinetics). The Weibull model ($P_{max}(1-$

exp.$(-\lambda t^b)$), where P_{max} is the maximum dissolved percentage, λ and b are model parameters) was fitted using a non-linear regression approach in which the parameters of the function (P_{max}, λ, and b) are sought using a Differen- tial Evolution optimizer (a global, stochastic optimizer) followed by a local optimizer based on gradient descent. The one-compartment model is de- fined by the differential equation $dC/dt = -Cl C/V + 1/V\, dD/dt$, where C is the concentration on the central compartment, Cl the clearance from the central compartment, V its apparent distribution volume, and D the external dose input. It is assumed that the dose is absorbed following a first-order process in which the amount remaining follows the eq. $A(t) = A_{max}(1 - \exp.(-K_a t))$ where A_{max} is the amount available in the bolus and K_a is the absorption rate. Note that the input along time to the central compartment is the opposite of the remaining amount, $dD/dt = -dA/dt$. This description in terms of differential equations follows the system's ap- proach and their numerical discretization described in (18). The parameters of the model are Cl and V and they are sought using the same combination of global, stochastic and local optimization described for the Weibull. Examples of such fittings are shown in Fig. 4. We then deconvolved the *in vivo* models to estimate the fraction absorbed *in vivo* (see Fig. 4 bottom). The system's identification approach proposed in (18) allows a clear separation between the input parameters K_a and the distribution and elimination parameters, Cl and V. Then, the deconvolution is performed by simply integrating the input dose, from 0 to t. The Levy plot is defined as the time-time transfor- mation that transforms the *in vitro* dissolution into the *in vivo* dissolution. It is typically performed using the average of the *in vitro* dissolution profiles and the average of the *in vivo* dissolution profiles. However, working with the averages loses the richness of the data underneath. Actually, any pair of vase-individual could have served to give an estimate of the Levy plot. Scipion PKPD allows doing that (Population analysis). Consequently, we may analyze the average Levy plot and the uncertainty around it (see Fig. 4 bottom). The whole workflow can be seen in Fig. 5.

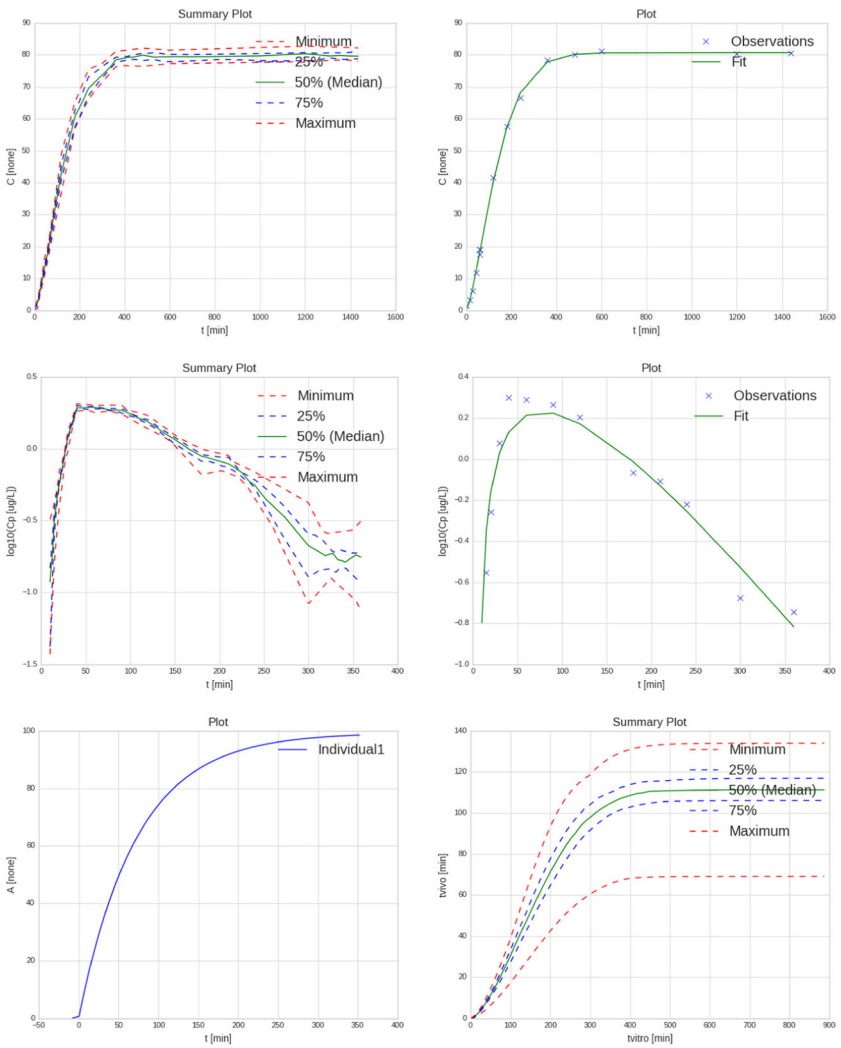

Fig. 4 Top row: Summary of the dissolution data for the 12 vessels (left) and an example of a fit (right). Middle row: Summary of the drug concentration profile in blood for the 12 individuals (left) and an example of a fit (right). Bottom row: Example of one of the estimated dissolution profiles for the individuals (left) and a summary of the population of Levy plots (right).t (right). Middle row: Summary of the drug concentration profile in blood for the 12 individuals (left) and an example of a fit (right). Bottom row: Example of one of the estimated dissolution profiles for the individuals (left) and a summary of the population of Levy plots (right).

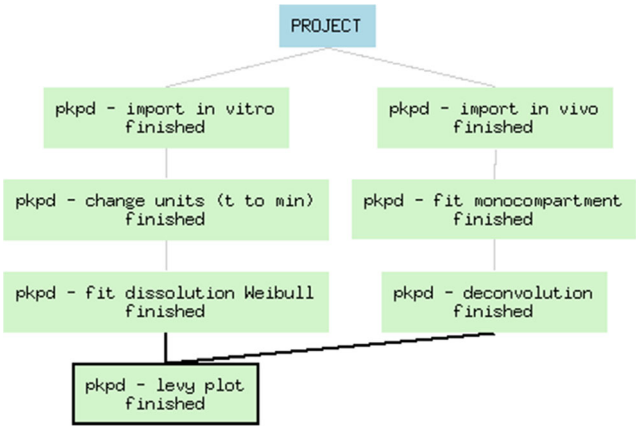

Fig. 5 Scipion workflow for the data analysis described in the Example section.

We have regularly used this software since 2015 in our Kinestat con- sultancy company (https://www.kinestatpharma.com), and the software is currently rather stable in its calculations. The correctness of these calcu- lations has been verified by over 40 numerical tests of the software (which are also publicly available at https://github.com/cossorzano/scipion-pkpd/ tree/master/pkpd/tests). These tests cover more than 20 examples from Gabrielsson's book (3), and the tests make sure that the calculations per- formed by Scipion PKPD give the same values for equivalent parameters as the results reported by Gabrielsson.

CONCLUSIONS

This paper has introduced Scipion PKPD, an open-source, free platform for biopharmaceutics, pharmacokinetics, and pharmacodynamics data analy- sis. It is based on a friendly graphical user interface so that no programming skills are needed as a user. It adds a reproducibility and traceability layer to data analysis in this domain and can be deployed in various systems, including HPC clusters and clouds. It currently contains the most widely used operations needed in these domains, and its open-source nature allows adding any other functionality required.

REFERENCES

1. Macheras P, Iliadis A. Modeling in biopharmaceutics, pharmacokinetics and pharmacodynamics: homogeneous and heterogeneous approaches, volume 30: Springer; 2016.
2. erez-Urizar J, Granados-Soto V, Flores-Murrieta FJ, Castan˜eda-Hern'andez G. Pharmacokinetic-pharmacodynamic modeling: why?, Arch Med Res 2000;31:539–545.
3. Gabrielsson J, Weiner D. Pharmacokinetic and pharmacodynamic data analysis: concepts and applications. 4th ed: Swedish Pharmaceutical Press; 2007.
4. Peters SA. Physiologically-based pharmacokinetic (PBPK) modeling and simulations: principles, methods, and applications in the pharmaceutical industry: John Wiley & Sons; 2012.
5. Duffull SB, Kirkpatrick CM, Green B, Holford NH. Analysis of population pharmacokinetic data using nonmem and winbugs. J Biopharm Stat. 2004;15:53–73.
6. Lunn DJ, Best N, Thomas A, Wakefield J, Spiegelhalter D. Bayesian analysis of population pk/pd models: general concepts and software. J Pharmacokinet Pharmacodyn. 2002;29:271–307.
7. Gieschke R, Serafin D. Development of innovative drugs via modeling with MATLAB: Springer; 2013.
8. Hosseini I, Gajjala A, Yadav DB, Sukumaran S, Ramanujan S, Paxson R, *et al.* gpkpdsim: a simbiology*QR* -based gui application for pkpd modeling in drug development. J Pharmacokinet Pharmacodyn. 2018;45:259–75.
9. Park J-S, Kim J-R. Non-compartmental data analysis using simbiology and matlab. Transl Clin Pharmacol. 2019;27:89–91.
10. Jonsson EN, Karlsson MO. Xpose—an s-plus based population phar- macokinetic/pharmacodynamic model building aid for nonmem. Comput Methods Programs Biomed. 1998;58:51–64.
11. Retout S, Mentre F. Optimization of individual and population designs using splus. J Pharmacokinet Pharmacodyn. 2003;30:417–43.
12. Dansirikul C, Choi M, Duffull SB. Estimation of pharmacokinetic pa- rameters from non-compartmental variables using microsoft excel. Comput Biol Med. 2005;35:389–403.
13. Zhang Y, Huo M, Zhou J, Xie S. Pksolver: an add-in program for pharmacokinetic and pharmacodynamic data analysis in microsoft excel. Comput Methods Prog Biomed. 2010;99:306–14.
14. Aarons L. Software for population pharmacokinetics and pharmacody- namics. Clin Pharmacokinet. 1999;36:255–64.
15. Bauer RJ, Guzy S, Ng C. A survey of population analysis meth- ods and software for complex pharmacokinetic and pharmacodynamic models with examples. AAPS J. 2007;9:E60–83.
16. de la Rosa-Trevín JM, Quintana A, Del Cano L, Zaldívar A, Foche I, Gutíerrez J, *et al.* Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. J Struct Biol. 2016;195:93–9.
17. Sorzano COS, erez-De-La-Cruz Moreno MA, Burguet-Castell J, Montejo C, Aguilar Ros A. Cost-constrained optimal sampling for system identification in pharmacokinetics applications with population priors and nuisance parameters. J Pharm Sci. 2015;104:2103–9.
18. Sorzano COS, erez-de-la Cruz Moreno MA, Martín FR, Montejo C, Aguilar A. A signal processing approach to pharmacokinetic data analysis. Pharm Re. 2021;38:625–35.
19. Price KV, Storn RM, Lampinen JA. Differential evolution: a practical approach to global optimization. Berlin, Germany: Springer; 2005.