Multivariate Data Methods for ASD Screening

Sofia Belém
Instituto Superior Técnico, Lisboa, Portugal

January 2021

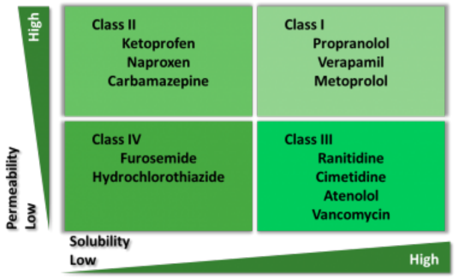
Abstract

Amorphous solid dispersions (ASDs) are a prominent formulation to overcome the challenge of low solubility in promising pharmacological compounds. Two important characteristics of ASDs are the drug loading in the formulation and the existence of the spring and parachute effect. Two successful PLS models were developed for the prediction of the drug loading: one taking into account drug and polymer, with accuracy of 45% in an external validation set, an accuracy of 71% for a set of commercialized ASDs and an accuracy of 81% for a set of internally developed ASDs (for threshold of 10%), and the other one taking into account uniquely the drug, with accuracy of 50% in an external validation set, an accuracy of 57% for a set of commercialized ASDs and an accuracy of 75% for a set of internally developed ASDs (for threshold of 10%). These models were shown to be more accurate than the Flory-Huggins theory. For the prediction of the spring and parachute effect, two models were developed: a random forest model and an artificial neural network model, with an accuracies for an external validation dataset of, respectively, 67% and 57%.

Keywords: Amorphous Solid Dispersions; Multivariate Data Analysis; Machine Learning; Partial Least Squares; Artificial Neural Network; Random Forest

1. Introduction

In any drug, for the active pharmaceutical ingredient (API) to be transported to its physiological target, it must be released and absorbed in the gastrointestinal (GI) tract, where it will enter the circulatory system. This implies that the bioavailability of a given drug is dependent on its ability to dissolve in the GI fluid (solubility) and to pass through the intestinal membrane (permeability) [1]. Based on this two concepts, a regulatory mechanism was created: the Biopharmaceutics Classification System (BCS) (Figure 1 [2]). Nowadays, ever more promising substances are crystalline molecules that are usually inserted in the classes II and IV of the BCS (low solubility).



Class II Ketoprofen Naproxen Carbamazepine	Class I Propranolol Verapamil Metoprolol
Class IV Furosemide Hydrochlorothiazide	Class III Ranitidine Cimetidine Atenolol Vancomycin

Figure 1: Class division of compounds by BCS [2].

Poor aqueous solubility results in a low dissolution rate, which is specially problematic for drugs with a restrict absorption window as they might dissolve after passing their absorptive sites [3]. Amorphous solid dispersions (ASDs) consist of a solid-solid blend of the API within a polymer excipient (the API molecules are uniformly dispersed within a polymer matrix); the mixture is vitrified so that the crystalline drug transforms into meta-stable amorphous glass [4]. Amorphous pharmaceutic products are characterized by its solid-state nature and lack of distinct intermolecular arrangement without crystalline structure and, consequently, with poor thermodynamic stability, providing enhanced solubility properties [5, 6].

Spray-dried amorphous solid dispersions (ASD SDs) present a number of advantages for low-solubility API delivery. They rapidly dissolve due to their high free energy, enhance the oral absorption of poorly soluble compounds by sustaining supersaturated concentrations of the drug in the GI fluid, and provide a physically stable drug form avoiding crystallization or phase separation [7, 8]. The dissolution behaviour of ASDs is often described by the ”spring and parachute” model (Figure 2). The ”spring” represents the initial phase where the drug is propelled into a solution as the polymer matrix dissolves, resulting in a supersaturated solution. In order to maintain the drug in the supersaturated state long enough for it to be absorbed, the polymer must also inhibit precipitation of the drug - the ”parachute”: precipitation

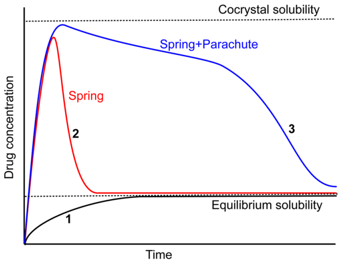


Figure 2: Schematic representation of the drug concentration–time profiles, illustrating the spring and parachute effect of supersaturating drug delivery systems [9].

inhibitors interact with the API molecules in solution, slowing down critical steps in the process of drug crystallization [9].

1.1. ASD SD's Key Performance Parameters

1.1.1 API properties

Key API properties include the melting temperature (T_m), glass transition temperature (T_g), partition coefficient ($logP$), logarithmic acid dissociation constant (pka), molecular weight, miscibility in polymers, hydrogen bond donors and acceptors, epithelial membrane permeability, solubility in aqueous media, solubility in spray solvents, and chemical stability [7, 8, 10, 11, 12, 13]. It has been demonstrated that physicochemical properties of the compounds that had fast crystal growth rates included lower molecular weights, high T_m values, lower T_g values, fewer rotatable bonds, lower melt entropy, lower melt viscosity and higher crystal densities [13, 14].

A very important aspect to be taken into account when manufacturing ASD SDs is the crystallization tendency of a given compound; two good parameters to evaluate this propensity are the T_m/T_g ratio and the $logP$. T_g represents the temperature at which amorphous materials transition from a hard, glassy state into a viscous state, while T_m represents the temperature at which a given substance changes from solid state to liquid. A high T_m implies a high crystallization tendency due to the high thermodynamic driving force; a low T_g poses a small kinetic barrier to molecular diffusion and, therefore, allows higher mobility, implying high crystallization tendency. Therefore, the higher the T_m/T_g ratio, the bigger the crystallization tendency. $LogP$, represents the ratio of concentrations of a compound in a mixture of two immiscible solvents at equilibrium. A high $logP$ value means the compound is highly hydrophobic, and therefore poorly soluble in water.

1.1.2 Polymer Choice

Polymers reduce the molecular mobility of the drug by forming intermolecular interactions between drug and polymer and reduce the chemical potential of the drug (minimizing the crystallization driving force), resulting in a stabilization of the ASD. An amorphous drug is usually most stable when drug and polymer are mixed homogeneously at molecular level. The strong interactions between an API and a polymer via ionic interactions, hydrogen bonding, halogen bonding, van der Waals forces, and hydrophobic interactions are expected to facilitate miscibility of the drug in the polymer and may increase physical stability. Reducing the drug mobility in the polymer matrix is also directly related to the parameter T_g: high T_g limits drug mobility and, thus, phase separation. In an SDD, the amorphous API is optimally homogeneously dispersed in the polymer matrix, so the dispersion exhibits a single T_g value, between the polymer T_g and the drug T_g [8].

1.1.3 API loading

The maximum drug loading in the ASD SD depends on the physical and chemical stability, dissolution performance, and powder properties as a function of drug loading. The maximum achievable loading is often limited for drugs with high T_m and low $logP$ values that have a strong tendency to crystallize from the amorphous state. Polymers in which the drug is more miscible or that offer a lower mobility environment can help stabilize the drug against crystallization or phase separation. In cases where the T_g is lower in the drug than in the polymer (most cases), increasing the drug load will also increase the tendency for the drug to crystallize. High drug loadings can also result in poor dissolution properties, especially for highly lipophilic drugs (with poor wettability in aqueous media) [8].

1.2. Prediction of ASD Stability Through Propensity for Phase Separation: Lattice Models and the Flory-Huggins Theory

The miscibility behaviour of these dispersions is typically described by the Flory-Huggins theory [15, 16], a lattice-based statistical mechanics model where the free energy of mixing is broken into an entropy part (that always favors mixing) and an enthalpy part (that can facilitate or prevent mixing, depending on the nature and intensity of the interaction between the components) [17]. The expression for the Gibbs energy of mixing is shown in Equation 1, where x_1 and x_2 are the molecular fractions of solvent and polymer (respectively), φ_1 and φ_2 are the volume fraction of solvent and polymer (respectively), χ is the solubility parameter and V_1

and V_2 are the molar volumes of the solvent and polymer (respectively) [18].

$$\frac{\Delta G}{RT} = x_1 \ln \varphi_1 + x_2 \ln \varphi_2 + \chi \varphi_1 \varphi_2 \left(x_1 + x_2 \frac{V_2}{V_1} \right) \tag{1}$$

The Hansen solubility parameters (HSP) [19, 20] are an attempt to extend solubility parameter theory to include polar and hydrogen-bonding interactions. The solubility parameter is divided into three partial solubility parameters: δD, descriptive of dispersion interactions, δP, relative to polar interactions, and δH, corresponding to hydrogen bonding interactions. The HSP can be used to predict how miscible two molecules are through the HSP distance (Ra, equation 2): the smaller the Ra, the more likely they are to be compatible [21].

$$\mathrm{Ra}^2 = 4\left(\delta \mathrm{D}_1 - 8\mathrm{D}_2\right)^2 + \left(\delta \mathrm{P}_1 - \delta \mathrm{P}_2\right)^2 + \left(\delta \mathrm{H}_1 - \delta \mathrm{H}_2\right)^2 \tag{2}$$

1.3. Motivation

The current ASD screening methodology employed requires a considerable amount of time and material resources. Presently, the initial *in-silico* analysis addresses only the computation of API properties and the assessment of the API/polymer system miscibility based on thermodynamic properties. Therefore, all supersaturation studies (for various API/polymer combinations and drug loads) are carried out *in vitro*, usually leading to a few promising conditions amongst many failed combinations. The goal of this project was to design an alternative workflow for the initial part of the ASD screening process that would save time and material resources. This workflow should allow scientists, when receiving a new API to formulate, to assess ASD stability and performance *a priori* through statistical and machine learning models that would predict two distinct outputs: the maximum API load and the existence (or absence thereof) of a spring and parachute behaviour.

2. Methodology

The variables were divided into "API descriptors", "Polymer descriptors" and "ASD variables / interaction parameters". In total, 136 observations referring to ASD SD formulations were harvested, comprised by combinations of 37 different APIs and 25 different polymers. The dataset includes two outputs, "API loading" and "Spring and parachute effect". All the observations used are referent to *in vitro* dissolution studies of ASD formulations – it was chosen not to include *in vivo* studies due to the high variability to those attached.

2.1. Partial Least Squares

Partial least squares (PLS) methods assume that the observed data is generated by a process driven by a small number of latent variables (not directly measured variables) – this is called indirect modeling [22]. A PLS model creates orthogonal score vectors (also called components) by maximising the covariance between different sets of variables. The predictor and predicted variables are each considered as a block of variables. The PLS model extracts the score vectors that will then serve as a new predictor representation, and regresses the response variables on these new predictors [23].

The most widely used method to evaluate the performance of a regression method is the r^2, a value between 0 and 1 (or 0% and 100%), that represents the goodness of the fit to the model, or the percentage of explained variance by the model. Another very important factor to take into consideration is the q^2 parameter, that represents the r^2 applied to the cross validation data (the term cross validation is explained furtherly). The q^2 parameter represents the predictive capability of the model, and at a certain degree of complexity will not improve any further and then degrade.

2.2. Artificial Neural Network

Artificial neural networks (ANN) are brain-inspired systems that consist of at least an input and an output layer of neurons (or nodes), and usually one or more hidden layers. The connections between the nodes are called weights, and each node has associated a "bias" term: the weights represent the strength of a particular node, and the bias term shifts the activation function up or down. The activation function serves the purpose of inserting non-linearity into the model: by calculating the weighted sum and further adding bias to it, it converts an input signal of a node to an output signal, which is used as input to the next layer. If no activation function is applied, the output would be merely linear, and while linear functions are easy to solve, they have very limited modeling power [24].

2.3. Random Forest

The random forest (RF) classifier, that consists of an assemble of decision trees, is one of the most used supervised learning models to approach classification problems. In a random forest, each decision tree makes an individual class prediction; in the end, the class that was predicted more often is the class predicted by the random forest [25]. The key for the good performance of these models is the low correlation between the individual trees; that way, the trees protect each other from their individual errors, as long as they don't constantly err in the same direction. The model uses two methods to ensure the behaviour of the independent trees is uncorrelated enough: bootstrap aggregation (or tree bagging) and feature randomness [26].

2.4. Overfitting and how to overcome it

When training a machine learning model, there are two phenomenons one has to look out for: bias and variance. Bias (or underfitting) is an algorithm's tendency to pick a model that is not structurally correct for the data, by making incorrect assumptions about the dataset. On the other hand, variance (or overfitting) arises from sensitivity to small fluctuations in the training set, because the model learned every quantitative detail of the training data, inevitably including random noise and missing the broader regularities in the data [27, 28, 29].

The primary step to avoid overfitting is to have a sufficient amount of data: if the training set is too small, even a simple model will adjust almost perfectly to it. Removing features is also very useful: having irrelevant features is not only expensive computationally and in the sense that it's necessary to harvest more data, but it also causes overfitting by introducing unnecessary noise and complexity into the model [29]. Another frequently used tool is early stopping: when an algorithm is being trained iteratively, it is possible to measure the performance of the algorithm at each iteration. Early stopping consists on stopping the training when the model reaches the point when, at a certain number of iterations, the error in the validation set starts increasing, as the model is starting to overfit the data [30]. Finally, an indispensable tool to avoid overfitting is validation, that consists on testing the model on data the model hasn't yet been exposed. Two of the most common validation techniques are hold-out validation (splitting the dataset into a "training set" and into a "validation set") and cross validation (splitting the dataset randomly into 'k' subgroups, or folds; one of the groups works as the validation set (hold-out fold), while the rest k-1 folds work as a training set) [31].

2.5. Model Development

2.5.1 Partial Least Squares

The PLS models were developed in the software SIMCA by Umetrics ®. Two separate PLS models were developed: one that takes into account API, polymer and interaction variables, and one that makes the predictions solely based on the API features. Both PLS models have as target output the prediction of the maximum API loading. For the PLS models, the dataset was scaled through mean normalization so that all the values would fall into a [-1, 1] interval.

The PLS model that takes into account API, polymer and interaction variables had as inputs 84 observations (comprised by combinations of 37 different APIs and 23 different polymers) and 30 features (or variables) descriptive of the observations. The model was then optimized: variables which mainly produced noise were excluded according to the VIP (Variable Importance in Projection) value, and outliers were excluded based on the prediction plot (predicted output by the model versus actual output) and the normal probability plot of residuals, referred to as n-plot. One of the assumptions for regression analysis is that the residuals (error terms, or the differences between the observed value of the dependent variable and the predicted value) are normally distributed, and this plot is a method of learning whether this is a valid assumption, and therefore to identify possible outliers. If the data follows a normal distribution with mean μ and variance σ^2, then a plot of the theoretical percentiles of the normal distribution versus the observed sample percentiles should be approximately linear. The validation was performed through cross-validation; to diminish overfitting and enhance the model's performance on new unseen data, each cross validation group consisted of the observations of a single API, so that when the q^2 valued was calculated, for each cross-validation step the model had never been exposed to that API before. Therefore, the value of k folds was the number of APIs in the model, so this valued changed during model optimization due to the removal of outliers.

The second PLS model developed is meant to predict the maximum API loading in a given ASD for a given API, without taking into account the polymer (its inputs are, therefore, the API characteristics). Prior to model optimization, the model had as inputs 37 observations (and, therefore, the same number of APIs) and 12 features (or variables) descriptive of the said APIs. The model was then optimized in the same manner as before: variables which mainly produced noise were excluded according to the VIP value, and outliers were excluded based on the prediction plot and the normal probability plot of residuals. Cross validation was performed; since in this model there are no two observations with the same API, the number of folds was simply defined as $k = 7$.

2.5.2 Random Forest

The RF classifier model was developed in the software MATLAB by MathWorks®, with the objective of predicting the output "spring and parachute" effect based on the variables descriptive of the API, of the polymer and of the ASD / interaction variables. This model should predict a class of 1 if the ASD is predicted to have good spring and parachute effect, and 0 otherwise. The model had as inputs 92 observations (comprised by combinations of 23 different APIs and 19 different polymers) and 29 features (or variables) descriptive of the observations. The random forest models do not require data scaling [32]. The validation was performed through cross-validation using 20 folds. The maximum number of splits was defined by default as 91, and the number of trees in the model was defined by default as 30.

2.5.3 Artificial Neural Network

The ANN model was developed in the software MATLAB by MathWorks®. For this model, the data was not normalized manually; instead, the function 'mapminmax' was applied. This model had the objective of predicting if the ASD would yield spring and parachute behaviour. The model had as inputs 92 observations (comprised by combinations of 23 different APIs and 19 different polymers) and 29 features (or variables) descriptive of the observations. In the ANN model, the dataset is divided in three: the training set, where the model is trained and the weights are updated; the validation set (hold-back validation), used to prevent overfitting through early-stopping, and the test set, that works as an independent test set without any role in the model training itself, but with the purpose of analyzing the performance of the model in new, unseen data. The division was 65% to training (60 samples), 15% to validation (14 samples) and 20% to testing (18 samples). The neural network architecture was defined as one input layer comprised of 29 nodes (the number of features); one output layer comprised of 1 node, that outputs either a value of 1 or a value of 0; and one hidden layer, with a number of nodes with approximately a medium value between the input and output layers (15 nodes). The backpropagation algorithm used was scaled conjugate gradient (SCG). For the hidden layer, the hyperbolic tangent sigmoid transfer function (returns values between -1 and 1) was used; for the output layer, the sigmoid transfer function was used (maps values between 0 and 1).

3. Results and Discussion

3.1. Prediction of maximum API loading

3.1.1 PLS model – all variables

After optimization by removal of unimportant variables and outliers (18 variables and 10 outliers), the r^2 and q^2 values obtained for this model were, respectively, 58% and 48%. The final model is taking as inputs 12 variables: 4 related to the polymer, 7 related to the API, and 1 interaction variable. The prediction plot for the final model is presented in Figure 3.

It is visible in the prediction plots (Figure 3) that, even when recurring to cross-validation (plot B), the data points do not fall far from the regression line.

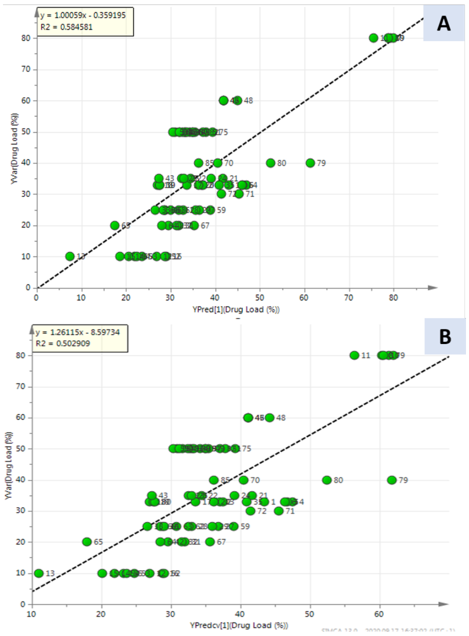


Figure 3: Prediction plots for the final model. A – actual values of the dependent variable versus the predicted values for this variable; B – actual values of the dependent variable versus the predicted values through cross validation for this variable.

In fact, even for the observations with the highest drug loading (observations 7-11, with a real drug loading of 80%), the model using cross-validation (meaning that the model hasn't been exposed to any observations with a drug loading as high) does an acceptable job at extrapolating, predicting values between 55% and 65%. Taking into account that, except those observations, there are only two observations with a drug loading of 60%, and all the other ones have smaller values, this extrapolation is indeed promising.

It is also worth noting that in both prediction plots, there are more data points falling below the trend line rather than on top; this means that it is more common for the model to predict a drug loading with a higher value comparing to its real value, than to predict a drug loading value that is inferior to the real drug loading. When harvesting the data, when possible the observations were taken from internal reports or literature papers where the authors tested several drug/polymer ratios in order to choose the highest possible API loading that conferred stability to the ASD. However, in many cases the authors did not perform these experiments; instead, often a ”typical” API:Polymer ratio was chosen (for example, 1:2 or 1:3). It is, then, logical to conclude that for many observations the drug loading that was used in the referred paper is not, indeed, the maximum API loading allowed for that API/polymer combination. Therefore, the fact that

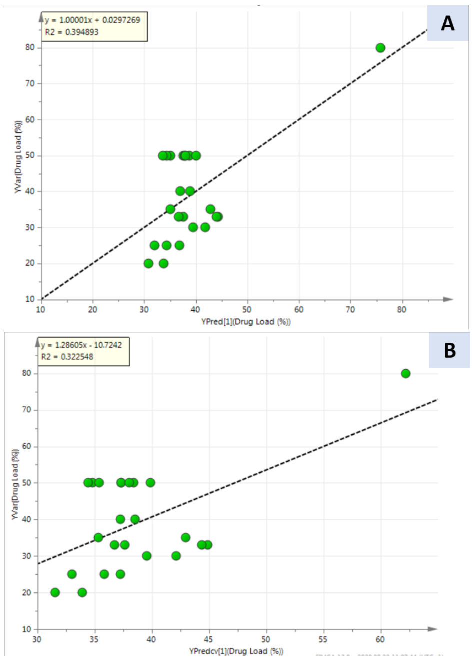


Figure 4: Prediction plots for the final API only model. A – actual values of the dependent variable versus the predicted values for this variable; B – actual values of the dependent variable versus the predicted values through cross validation for this variable.

the model is predicting more ”higher than real” values rather than ”lower than real” may mean that, at least in some cases, the value input as real may simply not be the maximum possible API loading.

3.1.2 PLS model – API variables only

It is also very interesting (and possibly even more useful) to be able to predict the maximum API loading for a given API prior to any decisions about the polymer. Therefore, a new model was developed: all the polymer variables were excluded, and for each API, only the observation with the highest drug loading parameter was kept. After optimization by removal of unimportant variables and outliers (8 variables and 8 outliers), the r^2 and q^2 values obtained for this model were, respectively, 39% and 31%. The final model is taking as inputs 6 variables. The prediction plot for the final model is presented in Figure 4.

It is visible in Figure 4 that, according to what was expected due to the lower number of predictors and observations, for the API only model the predictions are not as close to the trend line, compared to the model with all the features. However, even using cross validation (plot B), the maximum deviation between the real y value and the predicted y value is approximately 15-20%; this means that, even if the model doesn't predict exactly the maximum API loading for a new formulation, it certainly allows to greatly reduce the spectrum of API loadings to experimentally test. The model is also performing good extrapolations: the API with the highest y value has a drug loading of 80%, and the API with the second highest has a value of 50%; this means that, when using cross-validation (graphic B), the model predicts the value for the first observations based on a dataset that only goes as far as 50%. However, the model is predicting an API loading of over 60% for this API, which falls outside the range in which the model was trained. It is also worth noting that, similarly to the previous model, there are more APIs falling below the trend line rather than on top: the model is more commonly predicting a superior drug loading comparing to its real value rather than the contrary. As previously explained, many observations come from literature papers where the authors did not study the maximum API loading, but instead chose an average API loading, which means some observations may have as a ”real value” a drug loading that is, in fact, not the maximum possible for that given API.

Table 1: API loading predictions by the developed models for the output ”API loading” for an external validation dataset (observations harvested from literature). Yellow: observations that, for the PLS, have a difference between prediction and real value of over 10%.

API	**Polymer**	**API load (%)**	**Prediction PLS All Features**	**Prediction PLS API Only**
Rebamipide	PVP K30	**33**	44	36
Taranabant	HPMCAS L	**10**	27	23
Raloxifene	PVP K30	**20**	38	30
Itraconazole	HPMCP HP55	**33**	33	38
Sirolimus	Eudragit E	**33**	33	28
Sirolimus	HPMC	**10**	25	
Andrographolide	PVP K30	**33**	32	23
Piroxicam	PVP K25	**20**	45	38
Tadalafil	PVP/VA 64	**50**	38	31
Rivaroxaban	Eudragit 100L	**43**	36	29
Ciprofloxacin	HPMC E3	**50**	43	36

3.1.3 External validation

To further validate the results obtained, an external, completely independent dataset was harvested to be used as an external validation set ([33, 34, 35, 36, 37, 38, 39, 40, 41, 42]). The results obtained are presented on Table 1.

If a percentage of correct predictions is calculated taking into account a cutoff of 10% difference, values of 45% and 50% are obtained for the model with all features and the model with only the API features, respectively. These values are merely representative, since this validation set has only 11 observations (10 for the model with API variables only).

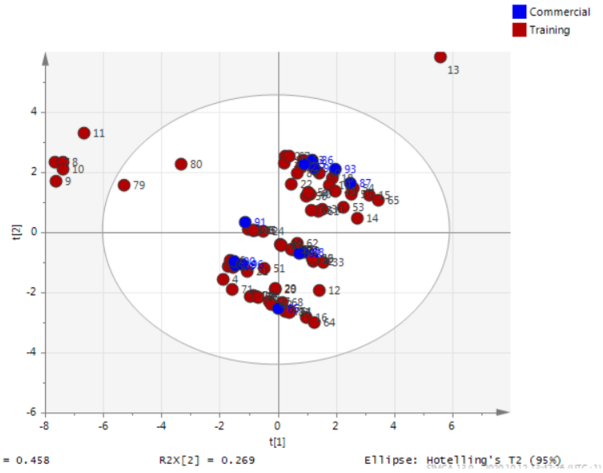


Figure 5: PCA map: distribution of the observations used in model training (red) and of the commercial ASDs (blue) across the first two principal components from the PCA analysis performed.

Moreover, the 10% value is an arbitrary cutoff; if this cutoff was altered to 15%, these accuracies go up to 73% and 70%, respectively. This cutoff of 15% would be perfectly acceptable: even if the model does not yield the exact, final API loading, it allows a great reduction of the interval of drug loadings to experiment.

3.1.4 Additional tests

Commercial ASDs

Some tests were performed in a set of real commercialised ASD formulations. If the model provided good predictions for APIs that were previously and successfully formulated as ASDs and commercialized as such, then the confidence in the model would increase greatly.

To begin with, the observations used to train the PLS model and the commercial observations were joined in a single dataset, and after removing the variables and observations removed in the original PLS model, a PCA with three principal components was performed (with an R^2 of 46% in the first component, 73% in the second component and 81% in the third component). The scores map obtained for the two first principal components is represented in Figure 5, where the observations used in the process of model training are represented in red and the commercial observations are represented in blue.

By analyzing Figure 5, one can see that the commercial ASDs perfectly adjusted to the PCA map built: there is not a single outlier in the commercial ASDs set, and they are homogeneously distributed amongst the training ASD observations. This gives extra confidence in the results to reluctant clients: nowadays, there is still a lot of reluctance in formulating APIs as amorphous solid dispersions due to the unpredictability of whether the formulation will be a success; however, if the new API falls into the confidence zone of this PCA map, it means that it is similar both to APIs used in the training of the model, and to APIs that were successfully formulated as ASDs – therefore, the new API is likely to be a good fit to be formulated as an ASD, and the predictions made by the model if the API is run by it are likely to be accurate.

Table 2: API loading predictions by the developed models for the output "API loading" for set of commercial ASDs. Yellow: observations that have a difference between prediction and real value of over 10%. Purple: observations with unknown real API loading values.

API	Polymer	Prediction PLS API + Polymer (%)	Prediction PLS API only8 (%)	Real Formulated API Load (%)
Elbasvir	HPMC	31	43	**25**
Evacetrapib	HPMC	24	36	**50**
Torcetrapib	HPMCAS L	33	36	unknown
Voxilaprevir	PVP/VA 64	45	45	**50**
Velpatasvir	PVP/VA 64	44	46	**50**
Telaprevir	HPMCAS L	42	44	**50**
Ivacaftor	HPMCAS H	31	37	**80**
Everolimus	HPMC	25	41	unknown
Etravirine	HPMC	31	41	unknown
Rosuvastatin	HPMC	33	43	unknown
Ledipasvir	PVP/VA 64	43	45	**50**

Followingly, the set of commercial ASDs was run through the models developed for the prediction of the API loading . The results obtained, as well as the real API load for these observations, are presented on Table 2. If a 10% threshold is taken into account, accuracies of 71% and 57% are obtained for the API+Polymer PLS and the API only PLS, respectively. If the threshold is changed to 15% instead of 10%, the accuracy of the PLS taking into account API and polymer is maintained, while the accuracy of the API only PLS also goes up to 71%. These accuracies are merely illustrative, since they are being calculated based on only 7 observations. However, for most of these observations, both PLS models do predict outputs very similar to the real API loading formulated, which is very promising.

Internal projects – comparison with Flory-Huggins theory

The Flory-Huggins (F-H) theory has been used to predict the maximum API loading for a given API/polymer combination. However, this methodology has some limitations, since it was originally created for a mixture of two polymers [43]. The PLS models developed would have some advantages over this approach: to begin with, the models were developed specifically to predict the success of amorphous solid dispersions, based on observations composed of APIs and polymers. Second of all, the F-H theory is assessing only the miscibility of the two

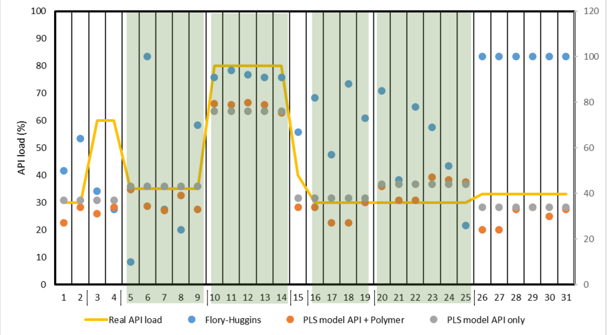


Figure 6: Graphical representation of the API load prediction for 31 observations from 8 different internal reports. Vertical lines in x-axis: division between different projects. Observations shaded in green: reports that had been used in the training of the PLS models that, therefore, may be yielding biased predictions for these models.

components, while the PLS models are taking into account more variables and, therefore, are probably evaluating more phenomenons than the F-H theory. Finally, unlike the F-H theory, the PLS model taking into account only the API features allows the prediction of the maximum API loading before any assessment of suitable polymers, which would allow the user to exclude beforehand APIs that would not yield a good ASD, independently of the polymer.

To evaluate if the models developed are actually advantageous over the F-H approach, it was necessary to check if the predictions of the referred models were better than (or at least as good as) the F-H predictions. For that purpose, 31 API/Polymer combinations from 8 different internal projects (and, therefore, 8 different APIs) were run through both PLS models to obtain an API loading prediction, and these values were compared with the predictions obtained by the F-H theory, present in said reports. The results obtained are shown in Figure 6.

The only observations where the predictions yielded by the F-H theory were closer to the real API loading were observation 3 and observation 7. For all the other observations, the PLS models were better at predicting the real API loading. For the PLS model containing API and polymer variables, only observations 3, 4, 20, 23, 24 and 25 have a $\Delta RealAPILoading/Prediction$ larger than 10% (which translates into an accuracy of 81%). For the four projects that weren't part of the training set, for the first one (observations 1 and 2) both predictions were extremely close to the real API loading (less than 5% difference), and for the last one (observations 26-31) two of the predictions (observations 29 and 30) were extremely close to the real API loading (less than 5% difference) and two of the predictions (28 and 31) were exactly equal to the real API loading. As for the PLS containing uniquely API features, only observations 3-4 (project 2) and 20-25 (project 7) have a $\Delta RealAPILoading/Prediction$ larger than 10% (meaning 2 APIs amongst 8 – accuracy of 75%). For the four projects that weren't part of the training set, for project 5 (observation 15) the value predicted is extremely close to the real API value (2% difference), being even closer than the prediction made by the PLS model taking into account all features; for the last project (observations 26-31), the prediction is also extremely close to the real value (1% difference). These results provide a very high level of confidence to the model: not only are the predictions made by the PLS models much more accurate than the previously used computational tool, but also they are extremely accurate for the observations tested. Moreover, since these observations are from internal reports, the workflow to obtain them is known, and therefore it is known that the API loading formulated was indeed thoroughly studied and extended to the maximum, and therefore, this validation is internally more valued than a validation obtained from using external observations.

3.2. Prediction of spring and parachute effect

3.2.1 Random forest model

The random forest classifier was explored in the prediction of the spring and parachute effect, through the assignment of the observations to a label ”0” (no spring and parachute effect) or ”1” (good spring and parachute effect). The overall accuracy for the model indicated by MATLAB was 78.3%. The confusion matrix referent to this model is presented in Figure 7.

For class ”1”, the performance seems to be very good – the model has a TPR of 84.2%. For the class ”0”, the TPR is lower, but still promising – 68.6%. This numbers may indicate that the model is more frequently predicting a positive output rather than a negative output – in practical terms, this means that it is more probable for the model to predict a bad ASD to be good, rather than to predict a good ASD to be bad. For the present purpose, it is actually an advantage for the classifier to more frequently predict false positives rather than false negatives – the prediction of false negatives may lead to the scientist not testing experimentally an API/polymer combination that may actually be successful.

The random forest model allows one to evaluate which variables are contributing more or less to the separation into classes through parallel coordinates plots. This plots show, for each variable (x

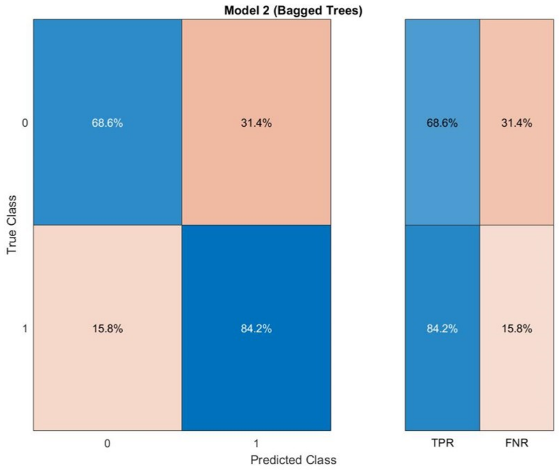


Figure 7: Confusion matrix referent to the random forest classifier model developed for the prediction of the spring and parachute effect. The diagonal (in blue) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class.

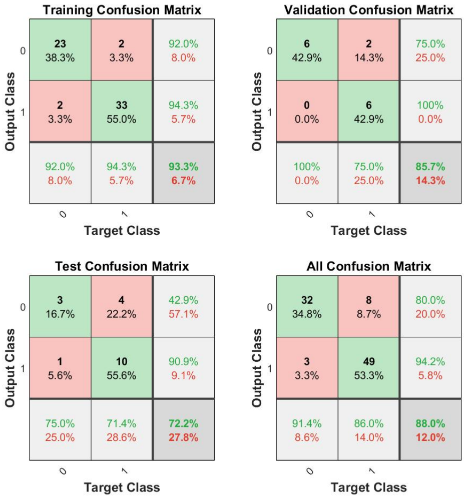


Figure 8: Confusion matrices referent to the several subsets of data used in the training of the ANN model for the prediction of the spring and parachute effect. The diagonal (in green) represents the TPR for each class, while the remaining cells (in red) represent the FNR for each class.

axis), the standard deviation (y axis) represented in different colours for different output classes: this way, if for a given variable the colours are distinctly separated, or at least somewhat differentiated, that variable is important for the prediction capacity of the model. A non-linear algorithm such as the random forest classifier requires a large enough dataset; therefore, despite the use of 20-fold cross validation, overfitting may be occurring. A possible resolution is to remove variables that are only adding noise to the model. After this optimization step, the accuracy was maintained at 78.3% and the respective confusion matrix obtained was exactly similar to the previous one. This, therefore, means that the removed variables were merely adding noise and, while removing them did not improve the performance of the model on the training data, it may have reduced the overfitting (this model has less eight variables than before). Therefore, the external validation performed furhterly was performed on both models (before and after optimization).

3.2.2 Artificial neural network model

The predicted output was the same as in the RF model for spring and parachute prediction. The overall accuracy for the model indicated by MATLAB was 88.0%. The confusion matrices and the receiver operating characteristic (ROC) curves referent to this model are presented in Figures 8 and 9, respectively. A ROC curve is a graphical representation of the TPR (y axis) versus FPR (x axis) relationship at different classification thresholds. The closer the graphic is to the top and left, the better the performance (and, in opposition, the closer to the diagonal the less accurate).

The most relevant value belongs to the test set (the model's performance on a completely unseen and independent dataset) – 72.2%. This value is very promising: for 10 observations predicted to be good, 7 will actually be. In practical terms, this can be a very helpful model, since its main objective would be to alleviate the experimental testing and all the costs that come with it in the preliminary ASD screening steps, and not to predict an actual final result.

In relation to the ROC curves, in the test set, the classifier appears to perform better in the left size of the graphic (it is better at identifying likely positives than at identifying likely negatives), since at the end of the plot, the curve crosses the diagonal. The optimal threshold for FPR seems to be around 0.25 – for a balanced dataset, this value should be around 0.5. For the present purpose it is actually an advantage for the classifier to more frequently predict false positives rather than false negatives, as it has been mentioned. However, if the FPR is too high, the model becomes superfluous.

3.2.3 External validation

To further validate the results obtained, an external, completely independent dataset was harvested to be used as an external validation set ([33, 34, 35, 36, 37, 38, 39, 40, 41, 42]). The re-

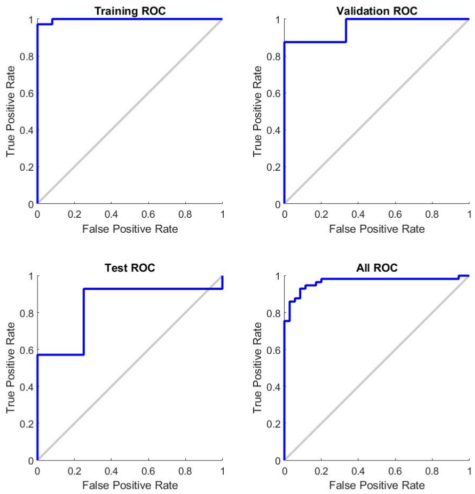


Figure 9: ROC curves referent to the several subsets of data used in the training of the ANN model for the prediction of the spring and parachute effect.

Table 3: Spring and parachute effect predictions by the developed models for this output for an external validation dataset (observations harvested from literature). Red: observations that are classified in the wrong class.

API	Polymer	Spring and Parachute Effect (real)	Prediction ANN	Prediction RF	Prediction RF Optimized
Rebamipide	PVP K30	**1**	1	1	1
Raloxifene	PVP K30	**1**	0	1	1
Sirolimus	Eudragit E	**1**	1	0	0
Sirolimus	HPMC	**0**	0	0	0
Tadalafil	PVP/VA 64	**1**	1	1	1
Rivaroxaban	Eudragit 100 L	**1**	0	1	1
Ciprofloxacin	HPMC E3	**0**	1	1	1
Sorafenib	PVP/VA 64	**1**	0	0	0
Sorafenib	PVP K30	**0**	0	0	0

sults obtained are presented on Table 3.

Both random forest classifiers have predicted exactly equal outputs for this dataset. Since they have the same accuracy and exactly the same confusion matrix, this isn't unexpected; the removed variables were likely to merely provide noise, but since random forest models aren't prone to overfitting due to tree bagging, removing the features likely did not change the way the model predicted the output at all. However, it is still advantageous to have an algorithm run on as little features as possible in order to diminish the time and resources spent on obtaining the necessary variables for each observation to be tested.

Comparing the ANN and the RF models, for this external validation set one can obtain accuracies of 56% and 67%, respectively. An accuracy of 56% is not significant for a model with binary outputs – it is very close to 50%, which would mean that the predictions are merely arbitrary. However, an accuracy of 67% is high enough to be considered non-arbitrary. Taking into account the fact that, naturally, this accuracy is representative due to the limited size of the external validation set used, this model seems quite promising for a preliminary screening application.The random forest model was chosen as the preferential one.

4. Conclusions

These models provide a new promising method to greatly accelerate the initial process of ASD screening by predicting *in silico* two very important aspects of an ASD: if it will behave according to the spring and parachute effect in order to maintain the supersaturated condition long enough to be absorbed, and what is the maximum possible ratio of API to polymer (API loading). The experimental testing can be highly reduced, since ASDs unlikely to succeed can be excluded *a priori*, and formulations that go on to the next step of the screening can be formulated with an API loading close to the real maximum loading from the beginning, requiring less adjustments and rectifications. The models presented allowed for the development of a new workflow to be used by formulation scientists in the *in silico* step of the ASD screening.

As further steps, adding more observations to the datasets and balance them to prevent data skew would be advisable. It would also be possible to develop new models based on these ones that explored the addition of other excipients, such as surfactants, and mixtures of polymers instead of simply ASDs with a single polymer. Developing models taking into account *in vivo* data would also be a possibility; while this would have the advantage of taking into account factors such as permeability and absorption, *in vivo* data has a much higher variability and, therefore, it could make the development of the models a much harder task.

It would also be possible to include this approach to ASD screening in an automatized, high throughput screening strategy. If this were to be implemented, the polymers predicted to be good could automatically move on to the next step of the screening and the resulting prototypes could be formulated in a high-throughput manner with API loadings similar to the one obtained by the PLS model for those API/polymer combinations (this model presented an accuracy of approximately 70-80% for a confidence interval of 10%, so the API loadings formulated could be in the 10% range, and a broader range would be explored if the result was unsatisfying). This approach would allow to use the models without need of human experimentation, highly accelerating the process of ASD screening, while maintaining or ameliorating the confidence in the *in-silico* results obtained.

References

[1] Daniel J Price, Felix Ditzinger, Niklas J Koehl, Sandra Jankovic, Georgia Tsakiridou, Anita Nair, René Holm, Martin Kuentz, Jennifer B Dressman, and Christoph Saal. Approaches to increase mechanistic understanding and aid in the selection of precipitation inhibitors for supersaturating formulations–a peer review. *Journal of Pharmacy and Pharmacology*, 71(4):483–509, 2019.

[2] Absorption Systems. Biopharmaceutical classification system. Accessed 20-Feb-2020.

[3] Sandrien Janssens and Guy Van den Mooter. Physical chemistry of solid dispersions. *Journal of Pharmacy and Pharmacology*, 61(12):1571–1586, 2009.

[4] Ralm G Ricarte, Nicholas J Van Zee, Ziang Li, Lindsay M Johnson, Timothy P Lodge, and Marc A Hillmyer. Recent advances in understanding the micro-and nanoscale phenomena of amorphous solid dispersions. *Molecular pharmaceutics*, 16(10):4089–4103, 2019.

[5] Stéphanie Greco, Jean-René Authelin, Caroline Leveder, and Audrey Segalini. A practical method to predict physical stability of amorphous solid dispersions. *Pharmaceutical research*, 29(10):2792–2805, 2012.

[6] Teófilo Vasconcelos, Sara Marques, José das Neves, and Bruno Sarmento. Amorphous solid dispersions: Rational selection of a manufacturing process. *Advanced drug delivery reviews*, 100:85–101, 2016.

[7] Dwayne T Friesen, Ravi Shanker, Marshall Crew, Daniel T Smithey, WJ Curatolo, and JAS Nightingale. Hydroxypropyl methylcellulose acetate succinate-based spray-dried dispersions: an overview. *Molecular pharmaceutics*, 5(6):1003–1019, 2008.

[8] B Vig and M Morgen. Formulation, process development, and scale-up: Spray-drying amorphous solid dispersions for insoluble drugs. pages 793–820, 2017.

[9] Shahram Emami, Mohammadreza Siahi-Shadbad, Khosro Adibkia, and Mohammad Barzegar-Jalali. Recent advances in improving oral drug bioavailability by cocrystals. *BioImpacts: BI*, 8(4):305, 2018.

[10] Jared A Baird, Bernard Van Eerdenbrugh, and Lynne S Taylor. A classification system to assess the crystallization tendency of organic molecules from undercooled melts. *Journal of pharmaceutical sciences*, 99(9):3787–3806, 2010.

[11] Amrit Paudel, Zelalem Ayenew Worku, Joke Meeus, Sandra Guns, and Guy Van den Mooter. Manufacturing of solid dispersions of poorly water soluble drugs by spray drying: formulation and process considerations. *International journal of pharmaceutics*, 453(1):253–284, 2013.

[12] Xiaolin Charlie Tang, Michael J Pikal, and Lynne S Taylor. The effect of temperature on hydrogen bonding in crystalline and amorphous phases in dihydropyrine calcium channel blockers. *Pharmaceutical research*, 19(4):484–490, 2002.

[13] Niraj S Trasi, Jared A Baird, Umesh S Kestur, and Lynne S Taylor. Factors influencing crystal growth rates from undercooled liquids of pharmaceutical compounds. *The Journal of Physical Chemistry B*, 118(33):9974–9982, 2014.

[14] Bernard Van Eerdenbrugh, Jared A Baird, and Lynne S Taylor. Crystallization tendency of active pharmaceutical ingredients following rapid solvent evaporation—classification and comparison with crystallization tendency from under cooled melts. *Journal of pharmaceutical sciences*, 99(9):3826–3838, 2010.

[15] Paul J Flory. Thermodynamics of high polymer solutions. *The Journal of Chemical Physics*, 9(8):660–660, 1941.

[16] Maurice L Huggins. Thermodynamic properties of solutions of long-chain compounds. *Annals of the New York Academy of Sciences*, 43(1):1–32, 1942.

[17] Feng Qian, Jun Huang, and Munir A Hussain. Drug–polymer solubility and miscibility: stability consideration and practical challenges in amorphous solid dispersion development. *Journal of pharmaceutical sciences*, 99(7):2941–2947, 2010.

[18] Pavel Gurikov, Igor Lebedev, Andrey Kolnoochenko, and Natalia Menshutina. Prediction of the solubility in supercritical carbon dioxide: a hybrid thermodynamic/qspr approach. In *Computer Aided Chemical Engineering*, volume 38, pages 1587–1592. Elsevier, 2016.

[19] CM Hansen. Three dimensional solubility parameter and solvent diffusion coefficient. importance in surface coating formulation. *Doctoral Dissertation*, 1967.

[20] Charles M Hansen. The universality of the solubility parameter. *Industrial & engineering chemistry product research and development*, 8(1):2–11, 1969.

[21] The official site of Hansen Solubility Parameters and HSPiP software. The famous factor of 4 - dr hansen's view. Accessed 15-Sept-2020.

[22] Randall D Tobias et al. An introduction to partial least squares regression. In *Proceedings of the twentieth annual SAS users group international conference*, volume 20. SAS Institute Inc Cary, 1995.

[23] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*, pages 34–51. Springer, 2005.

[24] Machine Learning From Scratch. Neural networks: Feedforward and backpropagation explained optimization. Accessed 19-Apr-2020.

[25] Antanas Verikas, Evaldas Vaiciukynas, Adas Gelzinis, James Parker, and M Charlotte Olsson. Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness. *Sensors*, 16(4):592, 2016.

[26] Tony Yiu. Understanding random forest: How the algorithm works and why it is so effective. Accessed 20-Apr-2020.

[27] Aleksey Bilogur. Bias variance tradeoff. Accessed 27-Apr-2020.

[28] Masters in Data Science. What is the difference between bias and variance? Accessed 27-Apr-2020.

[29] Erica Briscoe and Jacob Feldman. Conceptual complexity and the bias/variance tradeoff. *Cognition*, 118(1):2–16, 2011.

[30] Elite Data Science. Overfitting in machine learning: What it is and how to prevent it. Accessed 28-Apr-2020.

[31] Eijaz Allibhai. Hold-out vs. cross-validation in machine learning. Accessed 27-Apr-2020.

[32] Siddharth Misra and Hao Li. Noninvasive fracture characterization based on the classification of sonic wave travel times. *Machine Learning for Subsurface Characterization*, page 243, 2019.

[33] Xinnuo Xiong, Kailin Xu, Shanshan Li, Peixiao Tang, Ying Xiao, and Hui Li. Solid-state amorphization of rebamipide and investigation on solubility and stability of the amorphous form. *Drug Development and Industrial Pharmacy*, 43(2):283–292, 2017.

[34] S Sotthivirat, C McKelvey, J Moser, B Rege, W Xu, and D Zhang. Development of amorphous solid dispersion formulations of a poorly water-soluble drug, mk-0364. *International journal of pharmaceutics*, 452(1-2):73–81, 2013.

[35] Tuan Hiep Tran, Bijay Kumar Poudel, Nirmal Marasini, Jong Soo Woo, Han-Gon Choi, Chul Soon Yong, and Jong Oh Kim. Development of raloxifene-solid dispersion with improved oral bioavailability via spray-drying technique. *Archives of pharmacal research*, 36(1):86–93, 2013.

[36] David Engers, Jing Teng, Jonathan Jimenez-Novoa, Philip Gent, Stuart Hossack, Cheryl Campbell, John Thomson, Igor Ivanisevic, Alison Templeton, Stephen Byrn, et al. A solid-state approach to enable early development compounds: Selection and animsal bioavailability studies of an itraconazole amorphous solid dispersion. *Journal of pharmaceutical sciences*, 99(9):3901–3922, 2010.

[37] Youngseok Cho, Eun-Sol Ha, In-Hwan Baek, Min-Soo Kim, Cheong-Weon Cho, and Sung-Joo Hwang. Enhanced supersaturation and oral absorption of sirolimus using an amorphous solid dispersion based on eudragit® e. *Molecules*, 20(6):9496–9509, 2015.

[38] C Bothiraja, Mukesh B Shinde, S Rajalakshmi, and Atmaram P Pawar. Evaluation of molecular pharmaceutical and in-vivo properties of spray-dried isolated andrographolide—pvp. *Journal of Pharmacy and Pharmacology*, 61(11):1465–1472, 2009.

[39] KE Wu, Jing Li, Wayne Wang, and Denita A Winstead. Formation and characterization of solid dispersions of piroxicam and polyvinylpyrrolidone using spray drying and precipitation with compressed antisolvent. *Journal of pharmaceutical sciences*, 98(7):2422–2431, 2009.

[40] K Wlodarski, L Tajber, and W Sawicki. Physicochemical properties of direct compression tablets with spray dried and ball milled solid dispersions of tadalafil in pvp-va. *European Journal of Pharmaceutics and Biopharmaceutics*, 109:14–23, 2016.

[41] Hong Yu and Kunn Hadinoto. Mitigating the adverse effect of spray drying on the supersaturation generation capability of amorphous nanopharmaceutical powders. *Powder Technology*, 277:97–104, 2015.

[42] Chengyu Liu, Zhen Chen, Yuejie Chen, Jia Lu, Yuan Li, Shujing Wang, Guoliang Wu, and Feng Qian. Improving oral bioavailability of sorafenib by optimizing the "spring" and "parachute" based on molecular interaction mechanisms. *Molecular pharmaceutics*, 13(2):599–608, 2016.

[43] Shakhawath Hossain, Aleksei Kabedev, Albin Parrow, Christel AS Bergström, and Per Larsson. Molecular simulation as a computational pharmaceutics tool to predict drug solubility, solubilization processes and partitioning. *European Journal of Pharmaceutics and Biopharmaceutics*, 137:46–55, 2019.