

Artificial intelligence strategies for improving nanomedicine **formulation development**

Thomas L. Moore¹

Valentina Di Francesco¹ | Daniela Boso² | Bernhard Schrefler^{2,3} | Paolo Decuzzi¹

¹ Laboratory of Nanotechnology for Precision Medicine, Istituto Italiano di Tecnologia

² Department of Civil, Environmental and Architectural Engineering, University of Padova

³ Institute for Advanced Studies, Technical University of Munich

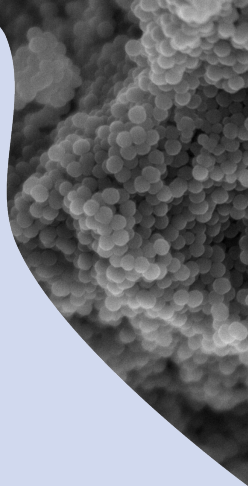
Tech Session 1: AI and Big Data in Drug Delivery

July 13, 2022

CRS 2022 Annual Meeting & Expo

July 11 – 15, 2022 | Montreal Congress Center, Montreal Canada

Advanced Delivery Science



Italy Locks Down Much of the Country's North Over the Coronavirus

The restrictions affect Milan and the regions that serve as Italy's economic engine, and are the most sweeping measures outside China.



The Duomo Square in Milan on Sunday, after the Italian government imposed a lockdown on much of the country's north because of the coronavirus. *Flavio Lo Scalzo/Reuters*

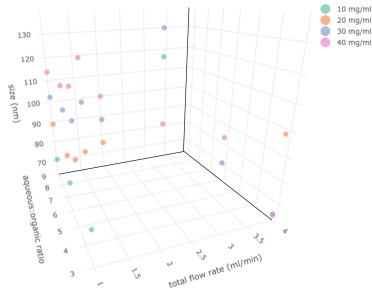
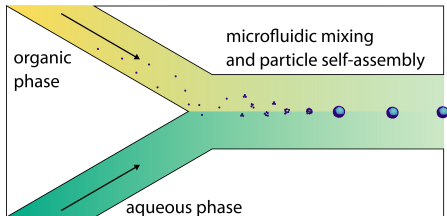


By Jason Horowitz

Published March 7, 2020 Updated March 11, 2020

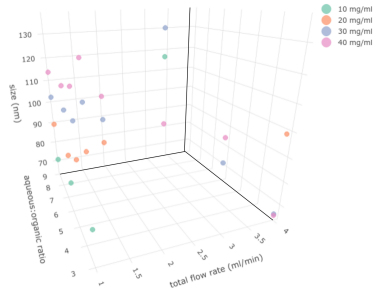
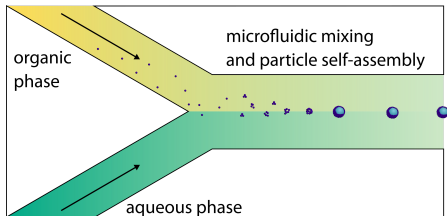


Microfluidic self-assembly of liposomes



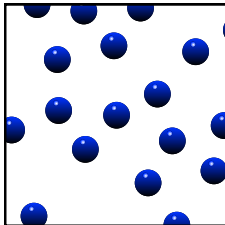
Dr. Valentina Di Francesco
Northeastern University

Microfluidic self-assembly of liposomes

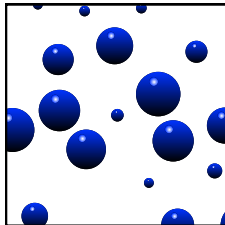


Dr. Valentina Di Francesco
Northeastern University

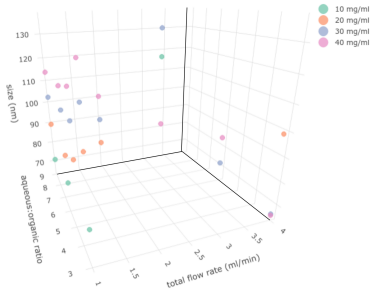
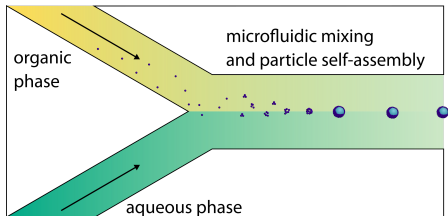
Monodisperse



Polydisperse

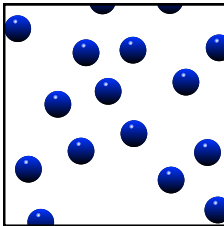


Microfluidic self-assembly of liposomes

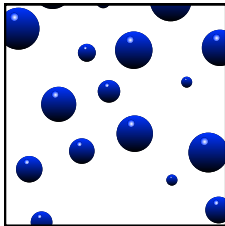


Dr. Valentina Di Francesco
Northeastern University

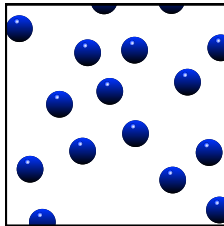
Monodisperse



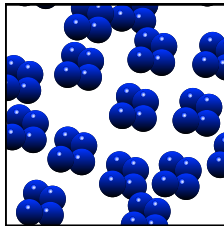
Polydisperse



Stable

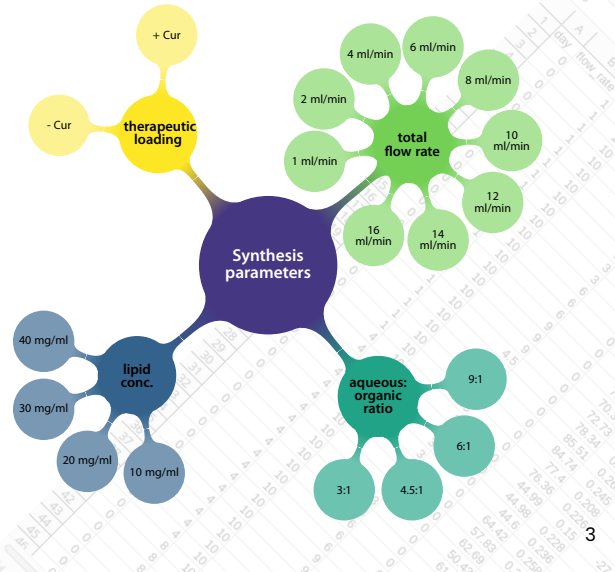


Aggregated



Total number of measurements: 3518

Total measurements after
trimming to $n \leq 6$: 1311

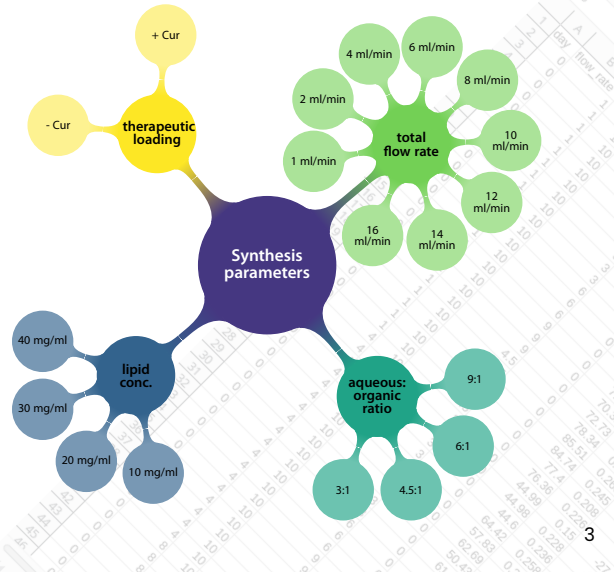


Total number of measurements: 3518

Total measurements after
 trimming to $n \leq 6$: 1311

**Number of unique [Day 0]
 formulations: 218**

Total number of [Day 0]
 measurements: 779



Total number of measurements: 3518

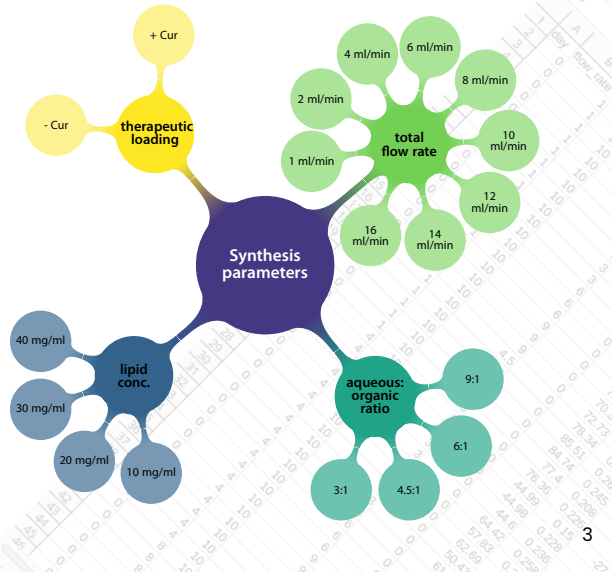
Total measurements after
 trimming to $n \leq 6$: 1311

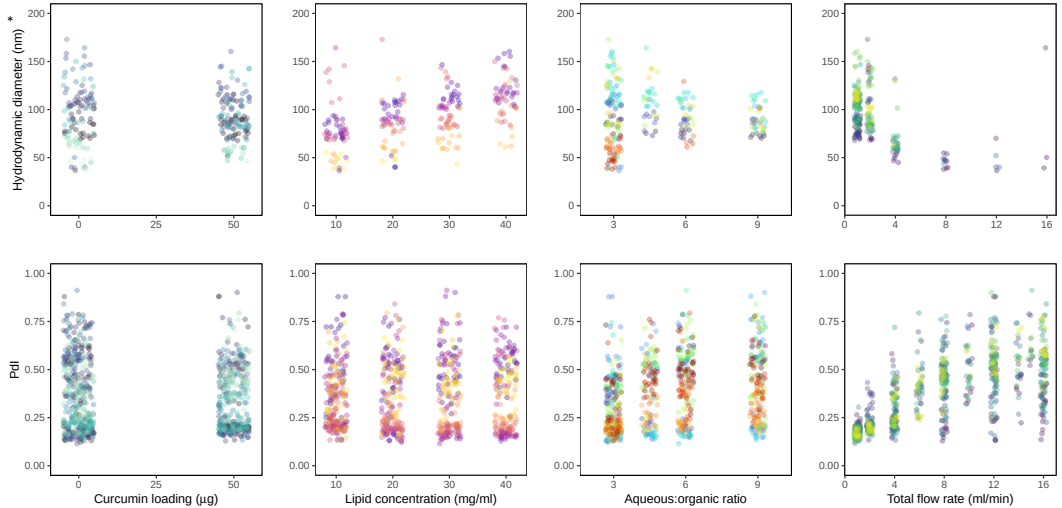
**Number of unique [Day 0]
 formulations: 218**

Total number of [Day 0]
 measurements: 779

**Number of unique monodisperse
 [Day 0] formulations: 72**

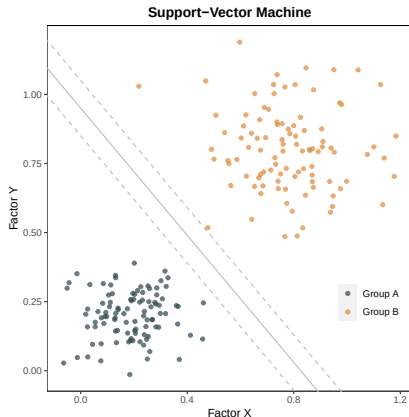
Total number of monodisperse
 [Day 0] measurements: 236





* Plotting only formulations with $PdI \leq 0.220$.

Supervised machine learning with scikit-learn¹



¹Pedregosa F, et al. *J. Mach. Learn. Res.*, **12**: 2825–2830 (2011)



```
# Dispersivity - classification training -----
array_classification = df.drop(['diam_nm', 'pdi', 'zeta_pot', 'perc_ee', 'ID'], axis = 1)
array_classification = array_classification.values

# Get an array of the important features (flow rate, aq/org ratio, concentration)
X_class_dispersivity = array_classification[:,0:4]
# Get the dispersivity values
y_class_dispersivity = array_classification[:,4]

# Randomly split the data into training data and validation data
X_train_dispersivity, X_val_dispersivity, Y_train_dispersivity, Y_val_dispersivity = train_test_split(X_class_dispersivity, y_class_dispersivity, test_size=0.4)

# Print length of training/validation data -----
print('# =====')
print('No. samples in classification (dispersivity) training data:')
print(len(X_train_dispersivity))
print('No. samples in classification (dispersivity) validation data:')
print(len(X_val_dispersivity))
print(' ')

# Create a list of classification algorithms
models_class = []
models_class.append(('LOR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models_class.append(('LDA', LinearDiscriminantAnalysis()))
models_class.append(('KNN', KNeighborsClassifier()))
models_class.append(('CART', DecisionTreeClassifier()))
models_class.append(('GBM', GaussianNB()))
models_class.append(('SVM', SVC(gamma='auto')))

# k-fold cross validation to predict classification algorithm accuracy
print('Predicted accuracy of the dispersivity classification models (k-fold cross validation):')
print('k splits = 6')
class_results = []
class_names = []
cv_class_mean = []
cv_class_sd = []
for name, model in models_class:
    kfold = StratifiedKFold(n_splits=10, random_state=42, shuffle=True)
    cv_results = cross_val_score(model, X_train_dispersivity, Y_train_dispersivity, cv=kfold, scoring='accuracy')
    class_results.append(cv_results)
    class_names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean()*100, cv_results.std()*100))
    cv_class_mean.append(cv_results.mean()*100)
    cv_class_sd.append(cv_results.std()*100)

print(' ') # print a blank line

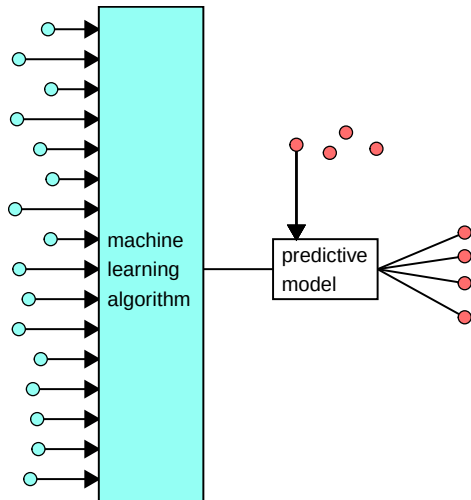
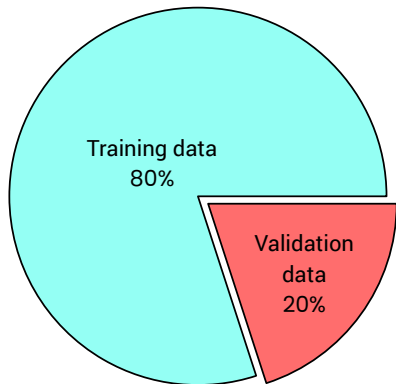
df_classpred = pd.DataFrame({'classification model': class_names, 'predicted accuracy (%)': cv_class_mean, 'stddev': cv_class_sd})

# Get the measured classification algorithm accuracy -----
print('Actual classification algorithm accuracy:')
pred_class_results = []
pred_class_names = []
pred_class_accuracy = []

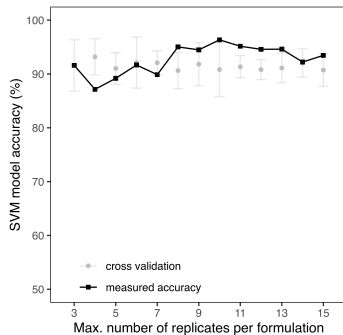
for name, model in models_class:
    model.fit(X_train_dispersivity, Y_train_dispersivity)
    pred_class_results.append(model.predict(X_val_dispersivity))
    pred_class_names.append(name)

for prediction in pred_class_results:
    pred_class_accuracy.append(accuracy_score(Y_val_dispersivity, prediction)*100)

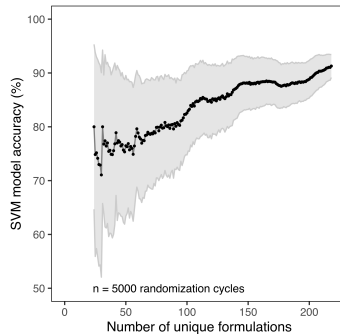
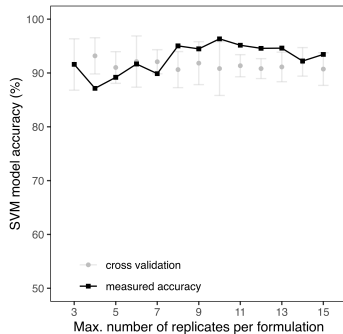
df_class_performance = pd.DataFrame({'model': pred_class_names, 'accuracy': pred_class_accuracy})
print(df_class_performance)
print('# =====')
print(' ') # print a blank line
```



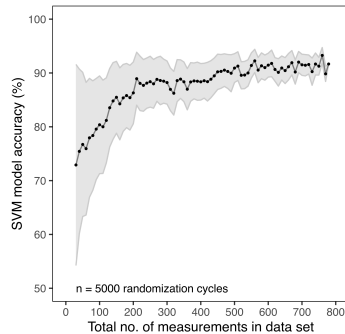
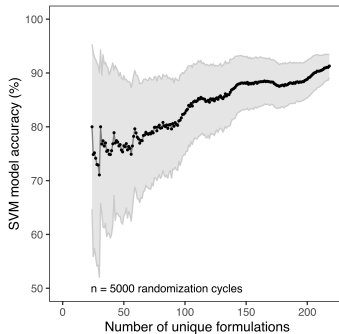
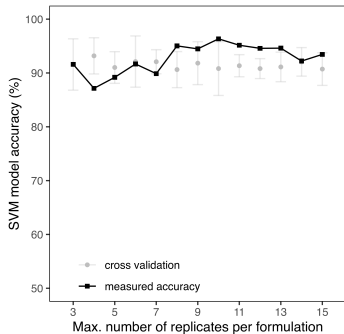
Factors affecting support-vector machines model accuracy



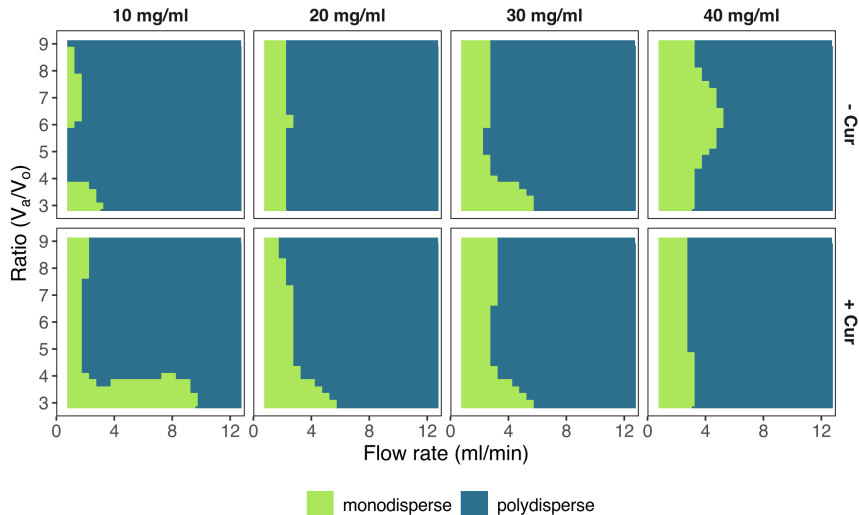
Factors affecting support-vector machines model accuracy



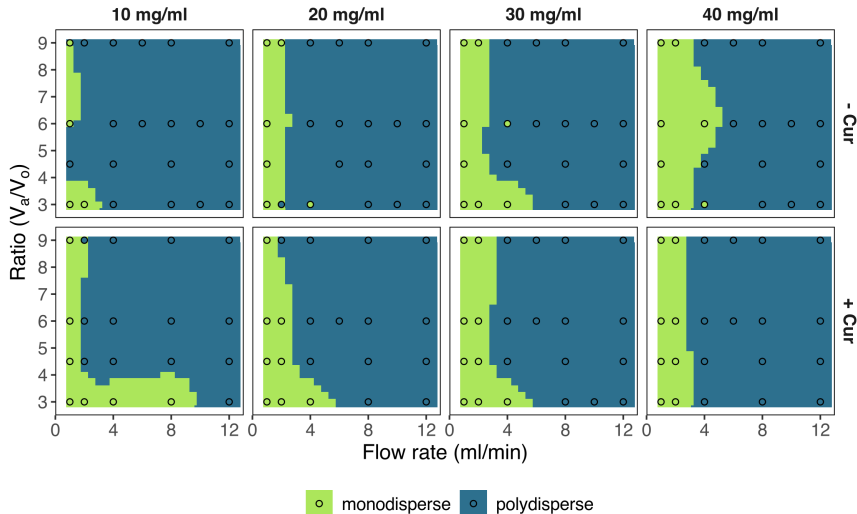
Factors affecting support-vector machines model accuracy



Predicted liposome dispersity

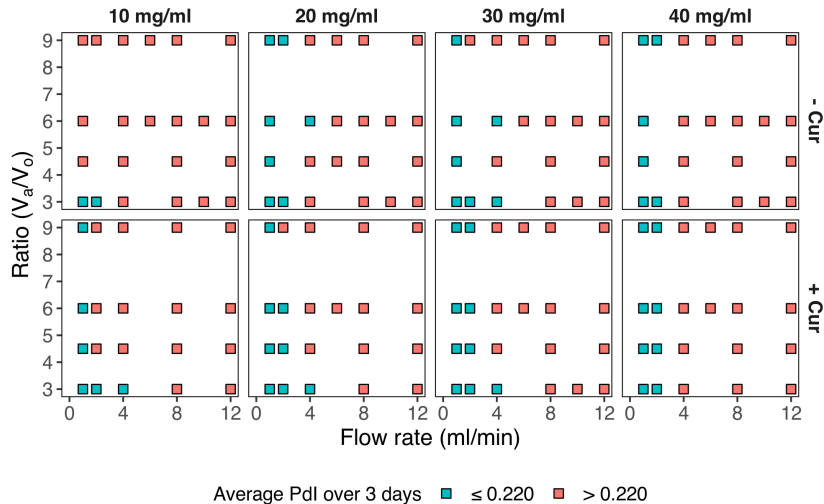


Predicted liposome dispersity

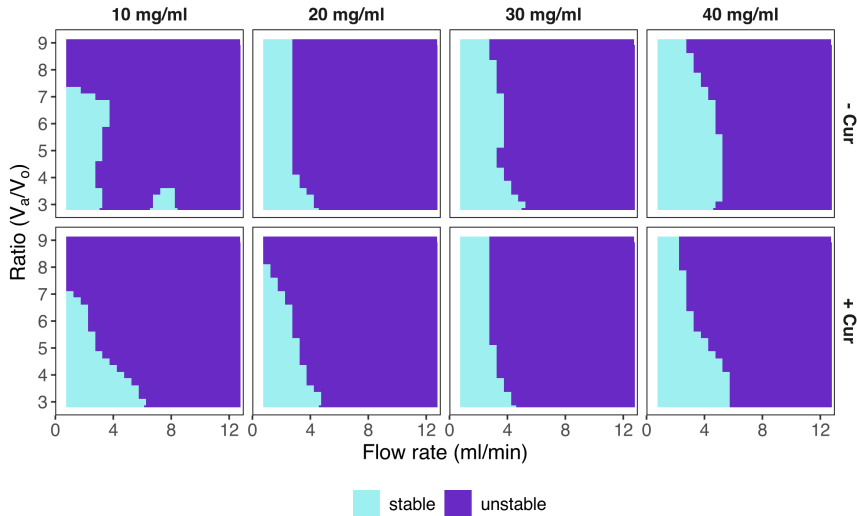


cross-validation accuracy: $92 \pm 5\%$, measured accuracy: 92%

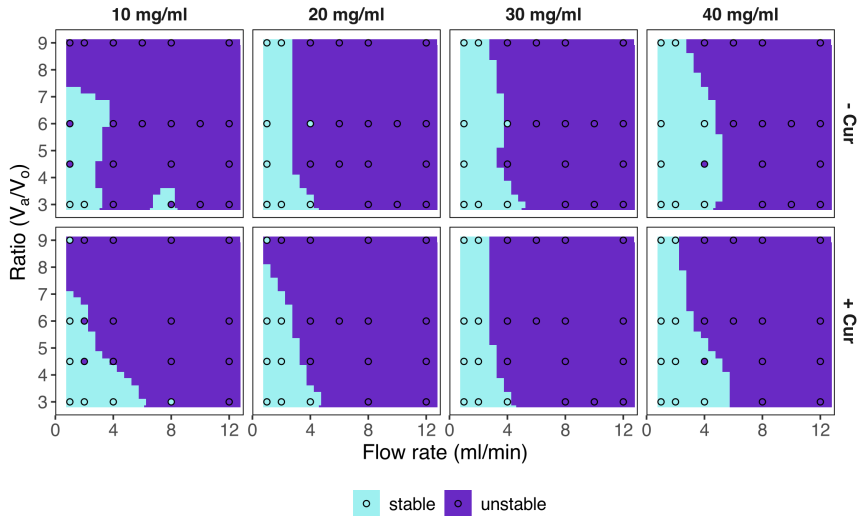
Liposome Pdl over time



Predicting liposome stability

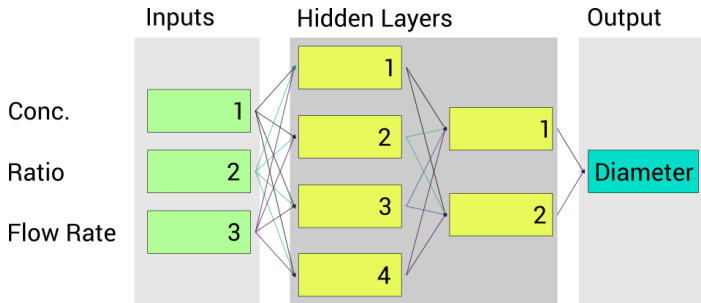


Predicting liposome stability



cross-validation accuracy: $82 \pm 5\%$, measured accuracy: 82%

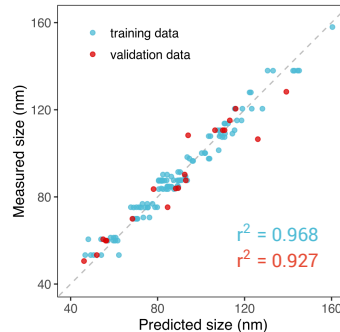
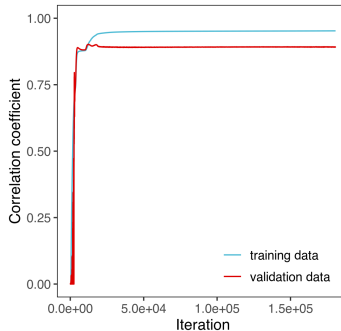
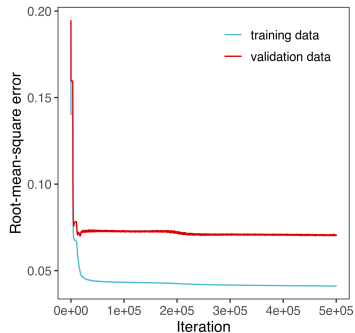
Artificial neural network (ANN) to predict diameter



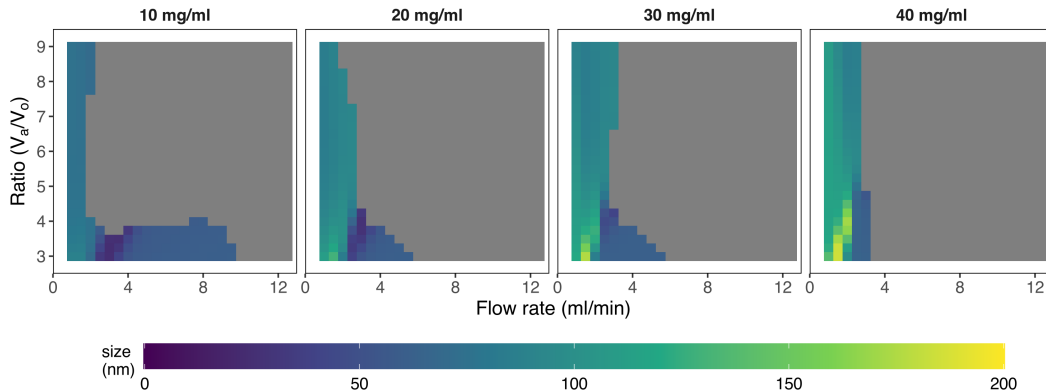
Prof. Daniela Boso
University of Padova

ANN are a field of **machine learning** that employ learning networks which “mimic” the behavior of the human brain to recognize patterns and solve problems.

Iterative ANN learning

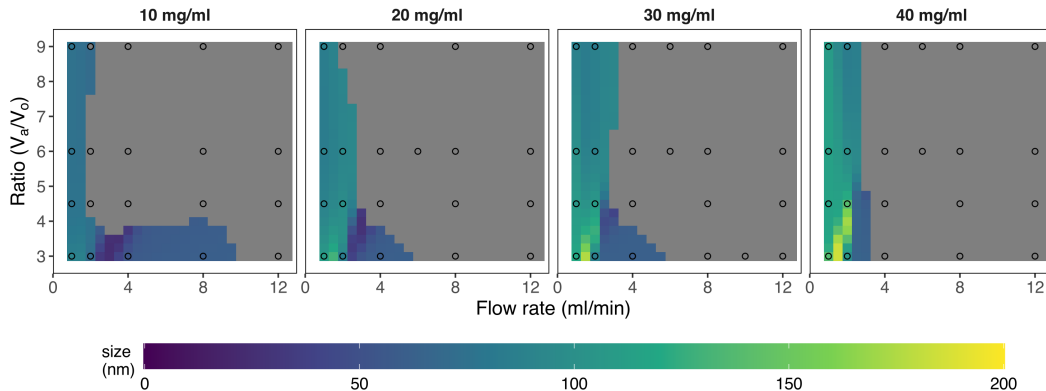


ANN diameter prediction heatmaps[†]



[†] For curcumin-loaded liposomes only

ANN diameter prediction heatmaps[†]



[†] For curcumin-loaded liposomes only

Shiny Liposome

Use the inputs below to select desired liposome properties. Click 'Submit' when you are finished to get appropriate formulations.

Press 'Reset' to start a new prediction.

Desired diameter (nm)



Submit

Reset

Total Flow Rate (ml/min)	Organic Phase Concentration (mg/ml)	Aqueous:Organic Volume Ratio	Predicted Diameter (nm)	Absolute Offset (nm)
1.50	10	5.50	75.00	0.00
1.00	10	7.00	75.16	0.16
1.00	10	6.75	75.16	0.16
1.00	10	6.50	75.17	0.17
1.00	10	6.25	75.20	0.20
2.00	10	3.50	75.22	0.22
1.00	10	6.00	75.25	0.25
1.50	10	5.75	74.74	0.26
1.00	10	5.75	75.33	0.33
1.50	10	5.25	75.34	0.34

Thank You

Istituto Italiano di Tecnologia

Prof. Paolo Decuzzi

Dr. Valentina Di Francesco

Università degli Studi di Padova

Prof. Daniela Boso

Prof. Bernhard Schrefler

CRS Italy Local Chapter **The MINDED programme**



@tmoore_BioNano



thomas.moore@iit.it

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement no. 754490 (MINDED) and the Fondazione Istituto Italiano di Tecnologia.

