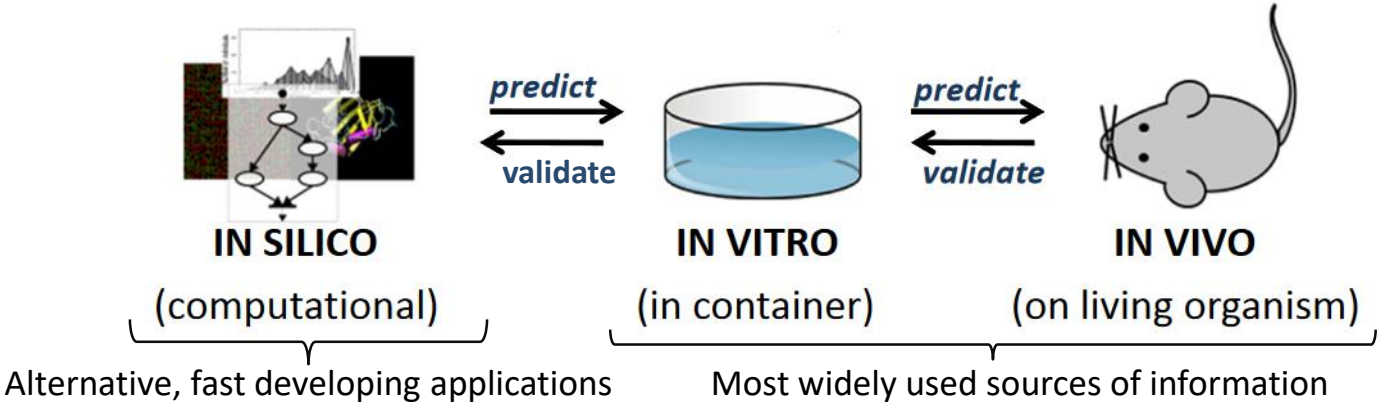
In-silico Toxicology (BB3R)

- All chemical substances need to be tested in terms of their toxicological, environmental or pharmaceutical properties before their use
  - Identify harmful effects on humans, animals, plants and the environment
  - Regulatory agencies require *in-vivo* testing for several toxic endpoints
  - ~ 2.9 million laboratory animals (Germany, 2010) and rising

➢ Establishment of alternative methods, reduction of animal testing



➢ Predict likely toxic compounds (QSAR, ADMETox, Machine learning)
➢ Reduce animal testing: a-priori exclusion from animal testing

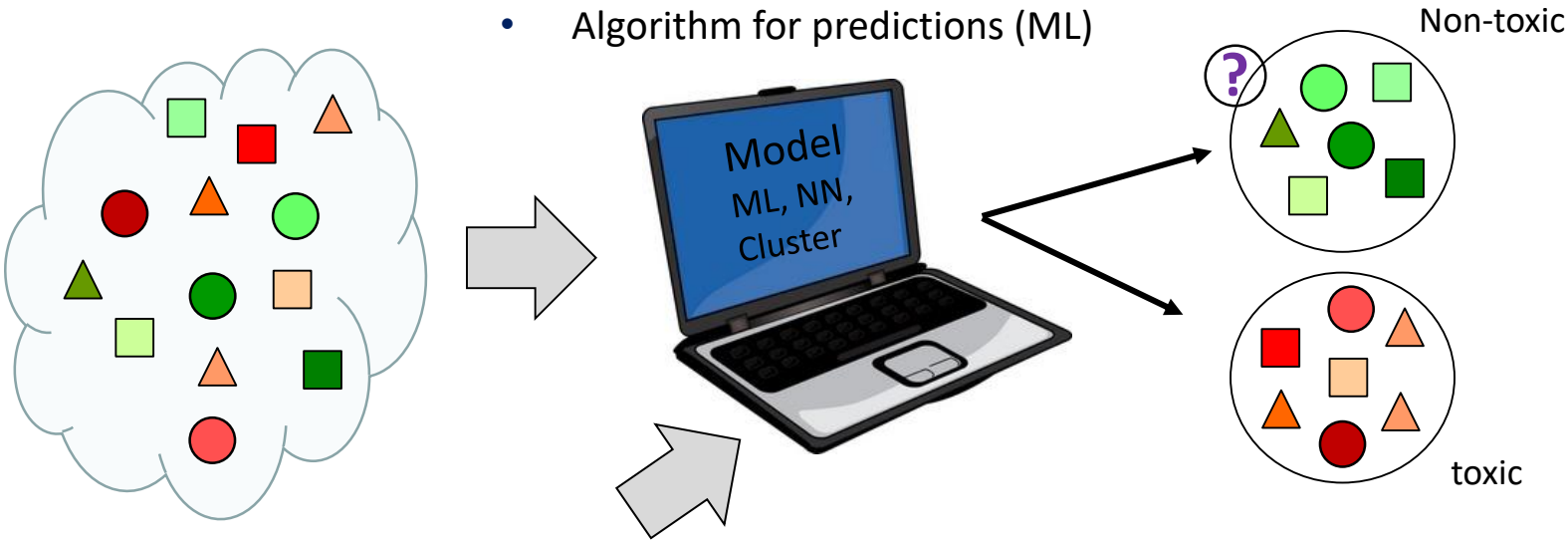
Project H: Ligand-based Toxicity Prediction

Assumption: Structurally similar molecules have similar properties and, thus, similar biological activity

Library of known toxic and non-toxic compounds against a specific toxic endpoint

Requirements:

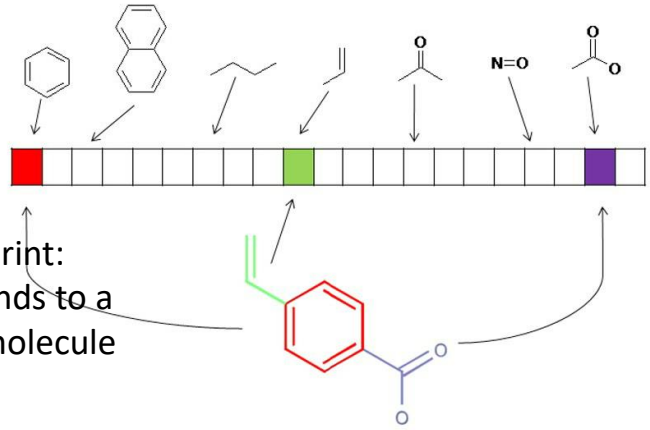
- Machine representation of the molecules
- Quantitative measure of similarity between the molecular descriptors
- Algorithm for predictions (ML)



- Learn the criteria that are important to distinguish between different outcomes

Prediction of Compound Toxicity - Workflow

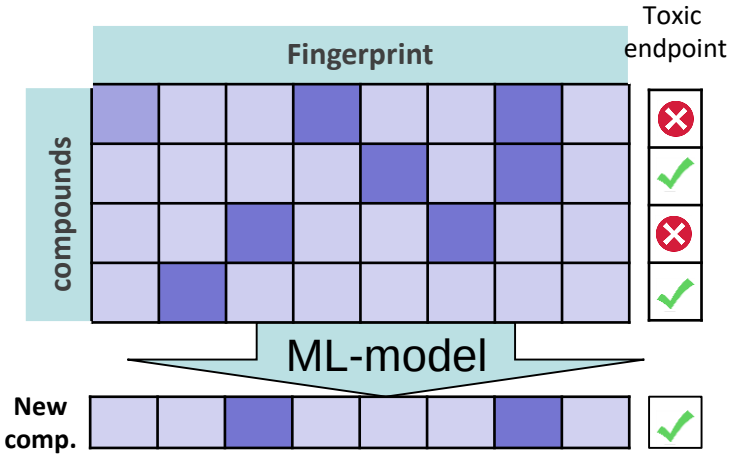
1. Data set
   - 10,000 compounds x 12 toxic endpoints (Tox21 challenge)
   - Fingerprint representation



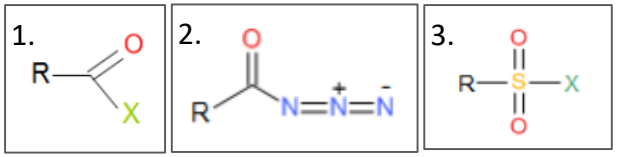
Molecular fingerprint:
Each bit corresponds to a fragment of the molecule

2. Methods
   - Machine learning (ML), i.e., support vector machine, random forest, neuronal nets, deep learning
   - Split test and training data
   - Model evaluation

Data matrix



3. Toxicity prediction on new compounds (and extraction of novel toxicophores)



Skin sensitization: Acid halides, acid azides, sulphonyl halides

Programming Skills (I) – Python/RDKit
IPython Notebooks and Compound Handling

Basic handling of molecules (i.e. reading of molecules, calculation of similarity)

More information can be here: http://www.rdkit.org/docs/index.html and http://www.rdkit.org/docs/api/index.html

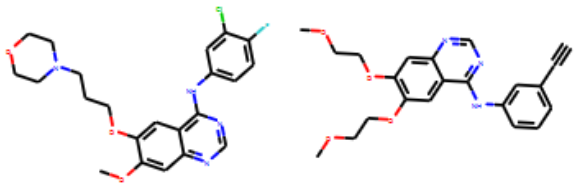
In [1]:
```
# The majority of the basic molecular functionality is found in module rdkit.Chem
from rdkit import Chem
from rdkit.Chem import AllChem
```

In [8]:
```
# Individual molecules can be constructed using a variety of approaches
# FDA approved EGFR inhibitors: Gefitinib, Erlotinib
m1 = Chem.MolFromSmiles('COc1cc2ncnc(Nc3ccc(F)c(Cl)c3)c2cc1OCCCN1CCOCC1')
m2 = Chem.MolFromSmiles('C#Cc1cccc(Nc2ncnc3cc(OCCOC)c(OCCOC)cc23)c1')
```

Drawing molecules

In [9]:
```
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem import Draw
Draw.MolsToGridImage([m1, m2])
```

Out[9]:



Fingerprinting and Molecular Similarity

In [12]:
```
from rdkit.Chem import MACCSkeys, AllChem
from rdkit import DataStructs
# MACCS
MACCS_fp1 = MACCSkeys.GenMACCSKeys(m1)
MACCS_fp2 = MACCSkeys.GenMACCSKeys(m2)
MACCS_fp1.ToBitString()
```

Out[12]:
```
'00000000000000000000000000000000000000010001000000000000001000000010000001001010010000011100000101011011010101010
10101000010111001111100111101100011111110111101111111111110'
```

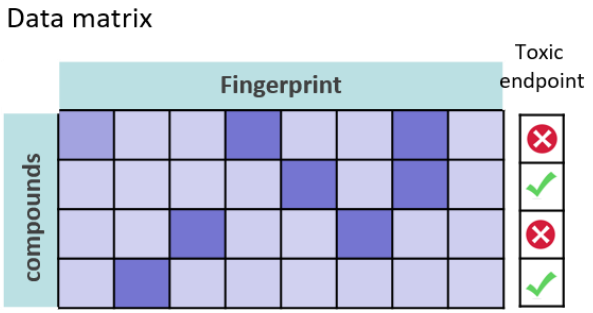
In [16]:
```
# Tanimoto similarity
DataStructs.TanimotoSimilarity(MACCS_fp1,MACCS_fp2)
```

Out[16]: 0.6923076923076923

Programming Skills (II) – Python/sklearn
Data Preprocessing and Machine Learning

Prepare data set

- Collect compounds (Tox21 data set)
- Convert compounds to fingerprint (MACCS or other)
- Collect classification per compound (toxic/non-toxic)



Machine learning, e.g. Random forest

Split data in train and test set

In [24]:
```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(fps, y, test_size=0.20)
```

Train the model

See http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html for an explanation of the parameter

In [22]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics

forest = RandomForestClassifier(n_jobs=-1, n_estimators=100, max_features=100)
forest.fit(X_train, y_train) # Build a forest of trees from the training set
```

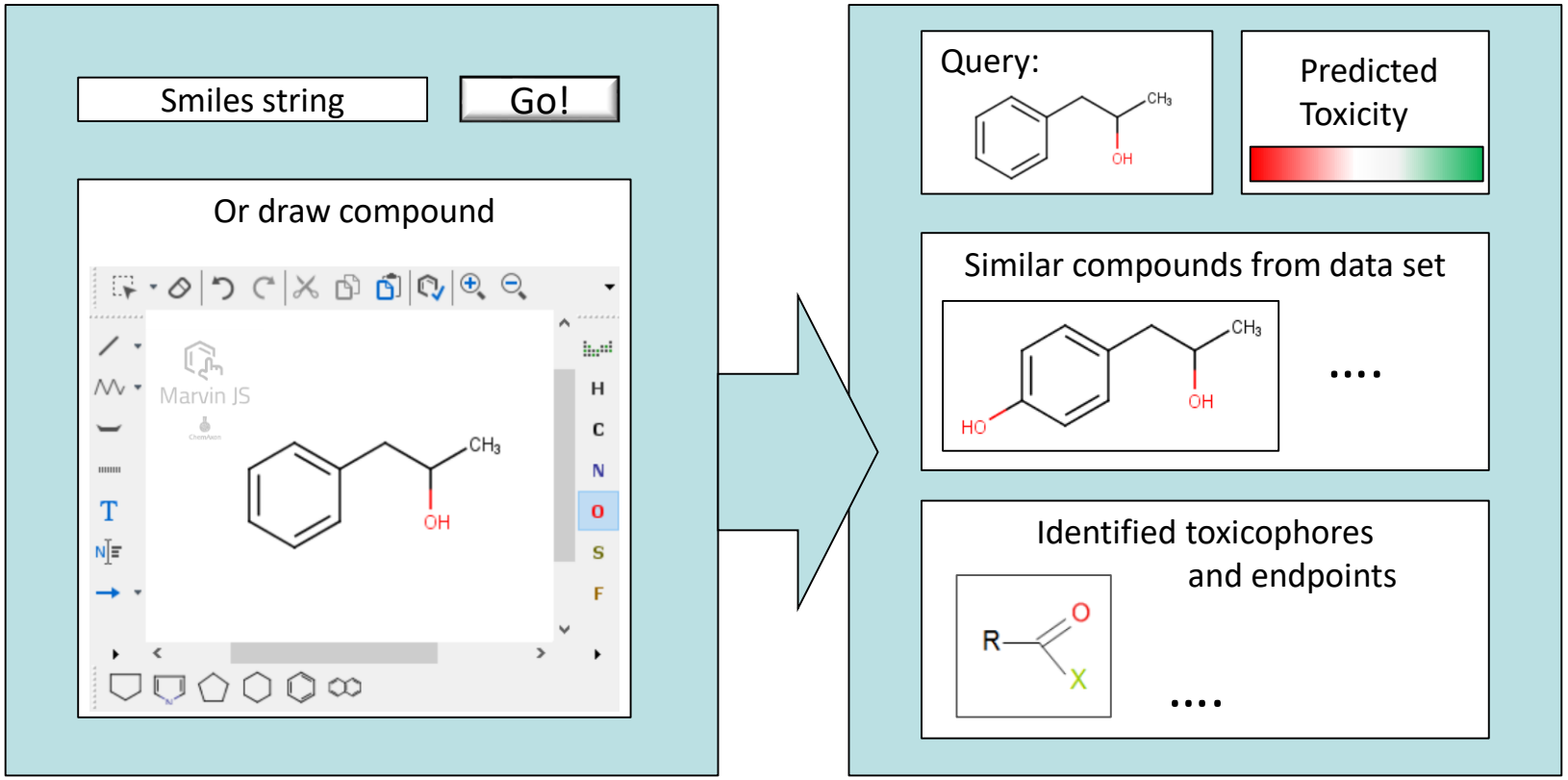
Test performance of the model

In [23]:
```
y_pred = forest.predict(X_test) # Predict class for X
accuracy = metrics.accuracy_score(y_test, y_pred)
roc_auc = metrics.roc_auc_score(y_test, y_pred)
print "Accuracy: %.2f; AUC: %.2f" % (accuracy, roc_auc)
```

```
Accuracy: 0.90; AUC: 0.92
```

Website Set-up/Design

- Query page
  - Input structure of interest
  - Smiles/mol format
- Result page
  - Predicted toxicity
  - Most similar compounds
  - Identified toxicophores



Requirements/Learning

- Insides into structural bioinformatics/cheminformatics
- Programming
  - Mainly Python language
  - Using libraries such as RDKit, sklearn, pandas, …
  - Some Web design
- Soft skills
  - Project management
  - Presentation
  - Scientific writing
- Work place
  - Campus Charité Mitte

Contact:
andrea.volkamer@charite.de