

The API of Chemics

February 24, 2016

1 Introduction

This document describes the API of Chemics by exemplifying with python code. The usage of the API is illustrated by creating the required input objects, calling the URL and displaying the returned object. All complex input objects, as well as all returned objects are of JSON format.

1.1 D360endpoints

This method obtains a dictionary with the names of all endpoints exposed to D360, together with information about the corresponding unit and the program used to predict the value of the endpoint. This object also controls the folder structure in D360. The output of this method is in the Appendix because of its length.

```
import urllib
import requests

5 MYSERVER = "192.168.100.27:8081"
  #MYSERVER = "chemics.medivir.com:8085"

def D360endpoints():

10     url = 'http://'+MYSERVER+'/D360endpoints'
      print url
      response = requests.get(url)
      print response.text

15 if __name__ == "__main__":

    D360endpoints()
```

Listing 1: Calling the method returning information about all available endpoints.

1.2 listAllEndpoints

The AllEndpoints method uses a special mode of execution in Chemics and calculates multiple endpoints simultaneously whereas all other endpoints are requested individually. This method defines the endpoints included in the AllEndpoints endpoint.

```
import urllib
import requests

5 MYSERVER = "192.168.100.27:8081"
  #MYSERVER = "chemics.medivir.com:8085"

def D360endpoints():

10     url = 'http://'+MYSERVER+'/listAllEndpoints'
      print url
      response = requests.get(url)
      print response.text

15 if __name__ == "__main__":

      D360endpoints()
```

Listing 2: List all endpoints predicted with the AllEndpoints method.

```
http://192.168.100.27:8081/listAllEndpoints
{
  "endpoints": [
    "logP",
    5 "logD",
    "MDCK",
    "Peff",
    "RuleOf3",
    "RuleOf5",
10 "Sp",
    "Acidic_pKa",
    "Acidic_pKa_74prox",
    "pKa_mostAcidic",
    "Basic_pKa",
15 "Basic_pKa_74prox",
    "pKa_mostBasic",
    "Mixed_pKa",
    "Mixed_pKa_74prox"
  ]
20 }
```

Listing 3: Output of the listAllEndpoints method.

1.3 Prediction

This method is used to obtain a prediction for a single molecule providing the smiles as an input. Please note that url encoded smiles are assumed for both input and output objects.

```

import urllib
import requests

5 MYSERVER = "192.168.100.27:8081"
  #MYSERVER = "chemics.medivir.com:8085"

def testSingle(smiles, ID, endpoint):
10     smiles = urllib.quote(smiles)
      url = 'http://'+MYSERVER+'/prediction/'+endpoint+'/'+ID+'/'
          '+smiles+'/DummyProject/DummySeries'
      print url
      response = requests.get(url)
      print "Response from single molecule execution"
15     print response.text

if __name__ == "__main__":
20     smiles = "N(C(=S)NCCc1cccc1)c1ncc(cc1)C"
      ID = "Ibuprofen"
      endpoint = "logP"
      testSingle(smiles, ID, endpoint)

```

Listing 4: Calling the prediction method.

```

http://192.168.100.27:8081/prediction/logP/Ibuprofen/N%28C
%28%3DS%29NCCc1cccc1%29c1ncc%28cc1%29C/DummyProject/
DummySeries
Response from single molecule execution
{
5   "ID": "Ibuprofen",
      "confidence": "inAD",
      "descEndpoint": "logP_AP7.1",
      "descStatus": "Finished",
      "endpoint": "logP",
      "prediction": "1.90021973067163",
      "project": "DummyProject",
10     "series": "DummySeries",
      "smiles": "N%28C%28%3DS%29NCCc1cccc1%29c1ncc%28cc1%29C",
      "status": "Finished",
      "unit": ""
15 }

```

Listing 5: Output of the prediction method.

1.4 PredictionMV

PredictionMV predicts a single molecule given an MV number. A prediction can be returned provided that the MV number is associated with a smiles in the corporate database.

```
import urllib
```

```

import requests

5 MYSERVER = "192.168.100.27:8081"
  #MYSERVER = "chemics.medivir.com:8085"

def testSingle(ID, endpoint):
10     url = 'http://'+MYSERVER+'/predictionMV/'+endpoint+'/'+ID+
        '/DummyProject/DummySeries'
    print url
    response = requests.get(url)
    print "Response from single molecule execution"
    print response.text
15
if __name__ == "__main__":
    ID = "MV002863"
    endpoint = "TPSA"
20    testSingle(ID, endpoint)

```

Listing 6: Calling the predictionMV method.

```

http://192.168.100.27:8081/predictionMV/TPSA/MV002863/
  DummyProject/DummySeries
Response from single molecule execution
{
5   "ID": "MV002863",
    "confidence": "NaN",
    "descEndpoint": "TPSA_RDK12.12.1",
    "descStatus": "Finished",
    "endpoint": "TPSA",
    "prediction": "122.74",
10   "project": "DummyProject",
    "series": "DummySeries",
    "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5Cc1sccc1
              %29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5D%28C%28C1
              %29OC%28%3D0%29c1cccc1%29C0C%28%3D0%29c1cccc1",
    "status": "Finished",
15   "unit": "Angstrom^2"
}

```

Listing 7: Output of the predictionMV method.

1.5 BatchPredictions

BatchPredictions is used to obtain predictions for a batch of molecules providing the smiles. Please note that the input JSON object is constructed from a python list of dictionaries. One of the provided smiles is intentionally wrong to show the output of a failed prediction.

```

import urllib
import requests
import json

```

```

5 MYSERVER = "192.168.100.27:8081"
  #MYSERVER = "chemics.medivir.com:8085"

def testBatchPrediction(smilesList, endpoint):

10     url = 'http://'+MYSERVER+'/batchPredictions/'+endpoint
       print url

       myData = json.dumps(smilesList)
       headers = {'Content-type': 'application/json', 'Accept': '
               text/plain'}
15     response = requests.post(url, data=myData, headers =
           headers)

       print "Response from batch execution"
       print response.text

20 if __name__ == "__main__":

       smilesList = [{ "ID" : "Simeprevir", "smiles": "ACC1=C(C=
               CC2=C1N=C(C=C2OC3CC4C(C3)C(=O)N(CCCCC=CC5CC5(NC4=O)C(=
               O)NS(=O)(=O)C6CC6)C)C7=NC(=CS7)C(C)C)OC", "project" :
               "dummyProject", "series": "dummySeries"},
               {"ID" : "L-alanine", "smiles": urllib.quote(
               "N[C@@H](C)C(=O)O"), "project" : "
               dummyProject", "series": "dummySeries"},
               {"ID" : "Ibuprofen", "smiles": "CC(C)CC1=CC=
               C(C=C1)C(C)C(=O)O", "project" : "
               dummyProject", "series": "dummySeries"}]
25     endpoint = "Basic_pKa"
       testBatchPrediction(smilesList, endpoint)

```

Listing 8: Calling the batchPrediction method.

```

http://192.168.100.27:8081/batchPredictions/Basic_pKa
Response from batch execution
{
5   "batchResults": [
       {
           "ID": "Simeprevir",
           "confidence": null,
           "descEndpoint": "Basic_pKa_AP7.1",
           "descStatus": "ADMET Predictor unable to predict. Please
               check the smiles.",
10          "endpoint": "Basic_pKa",
           "prediction": null,
           "project": "dummyProject",
           "series": "dummySeries",
           "smiles": "ACC1%3DC%28C%3DCC2%3DC1N%3DC%28C%3DC2OC3CC4C
               %28C%29C%28%3D0%29N%28CCCC%3DCC5CC5%28NC4%3D0%29C
               %28%3D0%29NS%28%3D0%29%28%3D0%29C6CC6%29C%29C7%3DNC
               %28%3DCS7%29C%28C%29C%29OC",
15          "status": "Error",
           "unit": ""
       },
       {

```

```

20     "ID": "L-alanine",
    "confidence": "NaN",
    "descEndpoint": "Basic_pKa_AP7.1",
    "descStatus": "Finished",
    "endpoint": "Basic_pKa",
    "prediction": " 9.47",
25     "project": "dummyProject",
    "series": "dummySeries",
    "smiles": "N%5BC%40%40H%5D%28C%29C%28%3D0%290",
    "status": "Finished",
    "unit": ""
30 },
    {
    "ID": "Ibuprofen",
    "confidence": "NaN",
    "descEndpoint": "Basic_pKa_AP7.1",
35     "descStatus": "Finished",
    "endpoint": "Basic_pKa",
    "prediction": "NaN",
    "project": "dummyProject",
    "series": "dummySeries",
40     "smiles": "CC%28C%29CC1%3DCC%3DC%28C%3DC1%29C%28C%29C
        %28%3D0%290",
    "status": "Finished",
    "unit": ""
    }
45 ]
}

```

Listing 9: Output of the batchPrediction method.

1.6 BatchPredictionsMV

BatchPredictionsMV is used to obtain predictions for a batch of molecules providing the MV number. Please note that the input JSON object is constructed from a python list of dictionaries. The endpoint used in this example, AllAPendpoints, returns predictions from all ADMET Predictor endpoints that are exposed through Chemics. Because of the computational time, individual ADMET Predictor endpoints cannot be requested for more than 10 compounds. If predictions for a larger batch of molecules is required, the AllAPendpoints method should be used. This method returns predictions for all endpoints as displayed in Listing 9 and it is a fast route to ADMET Predictor predictions, which will prediction 3000 molecules in approximately 6 minutes. However, please note that the AllAPendpoints does not accept more than 1000 molecules.

```

import urllib
import requests
import json

5 MYSERVER = "192.168.100.27:8081"
#MYSERVER = "chemics.medivir.com:8085"

def testBatchPredictionMV(IDList, endpoint):

```

```

10     url = 'http://' + MYSERVER + '/batchPredictionsMV/' + endpoint
        print url

        myData = json.dumps(IDList)
        headers = {'Content-type': 'application/json', 'Accept': '
                    text/plain'}
15     response = requests.post(url, data=myData, headers =
                                headers)

        print "Response from batch execution"
        print response.text

20 if __name__ == "__main__":

    IDList = [{ "ID" : "MV080290", "project" : "dummyProject",
                "series": "dummySeries"},
               {"ID" : "MV002863", "project" : "dummyProject
                ", "series": "dummySeries"}]

    endpoint = "AllAPendpoints"
25     testBatchPredictionMV(IDList, endpoint)

```

Listing 10: Calling the batchPredictionMV method.

```

http://192.168.100.27:8081/batchPredictionsMV/AllAPendpoints
Response from batch execution
{
  "batchResults": [
5     {
        "ID": "MV080290",
        "confidence": "inAD",
        "descEndpoint": "RuleOf3_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
10     "endpoint": "RuleOf3",
        "prediction": 2,
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                    %29%28%3D0%29%3D0%29C",
15     "status": "Finished",
        "unit": ""
    },
    {
20     "ID": "MV080290",
        "confidence": "inAD",
        "descEndpoint": "Sp_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "Sp",
        "prediction": "40.9829024537796",
25     "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                    %29%28%3D0%29%3D0%29C",
        "status": "Finished",
        "unit": "mg/mL"
30     },
    {

```

```

35  "ID": "MV080290",
    "confidence": "inAD",
    "descEndpoint": "RuleOf5_AP7.1",
    "descStatus": "ADMET Predictor calculation finished",
    "endpoint": "RuleOf5",
    "prediction": "0",
    "project": "dummyProject",
    "series": "dummySeries",
40  "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
          %29%28%3D0%29%3D0%29C",
    "status": "Finished",
    "unit": ""
  },
  {
45  "ID": "MV080290",
    "confidence": null,
    "descEndpoint": "Acidic_pKa_AP7.1",
    "descStatus": "ADMET Predictor calculation finished",
    "endpoint": "Acidic_pKa",
    "prediction": " 10.87; 5.37",
50  "project": "dummyProject",
    "series": "dummySeries",
    "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
          %29%28%3D0%29%3D0%29C",
    "status": "Finished",
55  "unit": ""
  },
  {
    "ID": "MV080290",
    "confidence": null,
60  "descEndpoint": "Basic_pKa_AP7.1",
    "descStatus": "ADMET Predictor calculation finished",
    "endpoint": "Basic_pKa",
    "prediction": "None",
    "project": "dummyProject",
65  "series": "dummySeries",
    "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
          %29%28%3D0%29%3D0%29C",
    "status": "Finished",
    "unit": ""
  },
70  {
    "ID": "MV080290",
    "confidence": null,
    "descEndpoint": "Basic_pKa_74prox_AP7.1",
    "descStatus": "ADMET Predictor calculation finished",
75  "endpoint": "Basic_pKa_74prox",
    "prediction": "NaN",
    "project": "dummyProject",
    "series": "dummySeries",
    "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
          %29%28%3D0%29%3D0%29C",
80  "status": "Finished",
    "unit": ""
  },
  {
    "ID": "MV080290",

```



```

85     "confidence": "inAD",
      "descEndpoint": "logP_AP7.1",
      "descStatus": "ADMET Predictor calculation finished",
      "endpoint": "logP",
      "prediction": "2.27305027985408",
90     "project": "dummyProject",
      "series": "dummySeries",
      "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                %29%28%3D0%29%3D0%29C",
      "status": "Finished",
      "unit": ""
95   },
    {
      "ID": "MV080290",
      "confidence": "inAD",
      "descEndpoint": "MDCK_AP7.1",
100     "descStatus": "ADMET Predictor calculation finished",
      "endpoint": "MDCK",
      "prediction": "35.8521490840245",
      "project": "dummyProject",
      "series": "dummySeries",
105     "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                %29%28%3D0%29%3D0%29C",
      "status": "Finished",
      "unit": "cm/s*10-7"
    },
    {
110     "ID": "MV080290",
      "confidence": "inAD",
      "descEndpoint": "Peff_AP7.1",
      "descStatus": "ADMET Predictor calculation finished",
      "endpoint": "Peff",
115     "prediction": "1.02531842052889",
      "project": "dummyProject",
      "series": "dummySeries",
      "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                %29%28%3D0%29%3D0%29C",
      "status": "Finished",
120     "unit": "cm/s*10-4"
    },
    {
      "ID": "MV080290",
      "confidence": null,
125     "descEndpoint": "Acidic_pKa_74prox_AP7.1",
      "descStatus": "ADMET Predictor calculation finished",
      "endpoint": "Acidic_pKa_74prox",
      "prediction": "5.37",
      "project": "dummyProject",
      "series": "dummySeries",
130     "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                %29%28%3D0%29%3D0%29C",
      "status": "Finished",
      "unit": ""
    },
135   {
      "ID": "MV080290",
      "confidence": null,

```

```

140     "descEndpoint": "pKa_mostBasic_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "pKa_mostBasic",
        "prediction": null,
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
145         %29%28%3D0%29%3D0%29C",
        "status": "Finished",
        "unit": ""
    },
    {
150         "ID": "MV080290",
        "confidence": null,
        "descEndpoint": "Mixed_pKa_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "Mixed_pKa",
        "prediction": "None",
155         "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
        %29%28%3D0%29%3D0%29C",
        "status": "Finished",
        "unit": ""
160     },
    {
        "ID": "MV080290",
        "confidence": "inAD",
        "descEndpoint": "logD_AP7.1",
165         "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "logD",
        "prediction": "0.967635864779056",
        "project": "dummyProject",
        "series": "dummySeries",
170         "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
        %29%28%3D0%29%3D0%29C",
        "status": "Finished",
        "unit": ""
    },
    {
175         "ID": "MV080290",
        "confidence": null,
        "descEndpoint": "Mixed_pKa_74prox_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "Mixed_pKa_74prox",
180         "prediction": "NaN",
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
        %29%28%3D0%29%3D0%29C",
        "status": "Finished",
185         "unit": ""
    },
    {
        "ID": "MV080290",
        "confidence": null,
190         "descEndpoint": "pKa_mostAcidic_AP7.1",

```

```

195     "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "pKa_mostAcidic",
        "prediction": 10.87,
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "c1c%28ccc%28c1%29S%28NC%28NCCCC%29%3D0
                %29%28%3D0%29%3D0%29C",
        "status": "Finished",
        "unit": ""
    },
200     {
        "ID": "MV002863",
        "confidence": "inAD",
        "descEndpoint": "RuleOf3_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "RuleOf3",
205     "prediction": 3,
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
                Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
                D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
                c1cccc1",
        "status": "Finished",
210     "unit": ""
    },
    },
    {
        "ID": "MV002863",
        "confidence": "inAD",
        "descEndpoint": "Sp_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "Sp",
        "prediction": "0.00105835422262213",
220     "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
                Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
                D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
                c1cccc1",
        "status": "Finished",
        "unit": "mg/mL"
225     },
    },
    {
        "ID": "MV002863",
        "confidence": "inAD",
        "descEndpoint": "RuleOf5_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "RuleOf5",
        "prediction": "1",
        "project": "dummyProject",
        "series": "dummySeries",
235     "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
                Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
                D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
                c1cccc1",
        "status": "Finished",
        "unit": ""
    }

```

```

},
{
240   "ID": "MV002863",
      "confidence": null,
      "descEndpoint": "Acidic_pKa_AP7.1",
      "descStatus": "ADMET Predictor calculation finished",
      "endpoint": "Acidic_pKa",
245   "prediction": "None",
      "project": "dummyProject",
      "series": "dummySeries",
      "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
      "status": "Finished",
250   "unit": ""
},
{
  "ID": "MV002863",
  "confidence": null,
255   "descEndpoint": "Basic_pKa_AP7.1",
  "descStatus": "ADMET Predictor calculation finished",
  "endpoint": "Basic_pKa",
  "prediction": " 3.44; -0.34",
  "project": "dummyProject",
260   "series": "dummySeries",
  "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
  "status": "Finished",
  "unit": ""
},
265 {
  "ID": "MV002863",
  "confidence": null,
  "descEndpoint": "Basic_pKa_74prox_AP7.1",
  "descStatus": "ADMET Predictor calculation finished",
270   "endpoint": "Basic_pKa_74prox",
  "prediction": "3.44",
  "project": "dummyProject",
  "series": "dummySeries",
  "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
275   "status": "Finished",
  "unit": ""
},
{
  "ID": "MV002863",
280   "confidence": "inAD",
  "descEndpoint": "logP_AP7.1",
  "descStatus": "ADMET Predictor calculation finished",
  "endpoint": "logP",
  "prediction": "2.98384231012252",
285   "project": "dummyProject",

```

```

"series": "dummySeries",
"smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
"status": "Finished",
"unit": "",
},
{
  "ID": "MV002863",
  "confidence": "inAD",
  "descEndpoint": "MDCK_AP7.1",
  "descStatus": "ADMET Predictor calculation finished",
  "endpoint": "MDCK",
  "prediction": "72.6992890858162",
  "project": "dummyProject",
  "series": "dummySeries",
  "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
  "status": "Finished",
  "unit": "cm/s*10^7"
},
{
  "ID": "MV002863",
  "confidence": "inAD",
  "descEndpoint": "Peff_AP7.1",
  "descStatus": "ADMET Predictor calculation finished",
  "endpoint": "Peff",
  "prediction": "1.4344589969277",
  "project": "dummyProject",
  "series": "dummySeries",
  "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
  "status": "Finished",
  "unit": "cm/s*10^4"
},
{
  "ID": "MV002863",
  "confidence": null,
  "descEndpoint": "Acidic_pKa_74prox_AP7.1",
  "descStatus": "ADMET Predictor calculation finished",
  "endpoint": "Acidic_pKa_74prox",
  "prediction": "NaN",
  "project": "dummyProject",
  "series": "dummySeries",
  "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
D%28C%28C1%29OC%28%3D0%29c1cccc1%29COC%28%3D0%29
c1cccc1",
  "status": "Finished",
  "unit": ""
},
{

```

```

335     "ID": "MV002863",
        "confidence": null,
        "descEndpoint": "pKa_mostBasic_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "pKa_mostBasic",
        "prediction": -0.34,
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
            Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
            D%28C%28C1%290C%28%3D0%29c1cccc1%29COC%28%3D0%29
            c1cccc1",
340     "status": "Finished",
        "unit": ""
    },
    {
345     "ID": "MV002863",
        "confidence": null,
        "descEndpoint": "Mixed_pKa_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "Mixed_pKa",
        "prediction": "None",
350     "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
            Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
            D%28C%28C1%290C%28%3D0%29c1cccc1%29COC%28%3D0%29
            c1cccc1",
355     "status": "Finished",
        "unit": ""
    },
    {
360     "ID": "MV002863",
        "confidence": "inAD",
        "descEndpoint": "logD_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "logD",
        "prediction": "2.9837960983954",
        "project": "dummyProject",
        "series": "dummySeries",
365     "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
            Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
            D%28C%28C1%290C%28%3D0%29c1cccc1%29COC%28%3D0%29
            c1cccc1",
        "status": "Finished",
        "unit": ""
    },
370     "ID": "MV002863",
        "confidence": null,
        "descEndpoint": "Mixed_pKa_74prox_AP7.1",
        "descStatus": "ADMET Predictor calculation finished",
        "endpoint": "Mixed_pKa_74prox",
375     "prediction": "NaN",
        "project": "dummyProject",
        "series": "dummySeries",
        "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5

```

```

380         Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
           D%28C%28C1%290C%28%3D0%29c1cccc1%29C0C%28%3D0%29
           c1cccc1",
           "status": "Finished",
           "unit": ""
385     },
     {
       "ID": "MV002863",
       "confidence": null,
       "descEndpoint": "pKa_mostAcidic_AP7.1",
       "descStatus": "ADMET Predictor calculation finished",
       "endpoint": "pKa_mostAcidic",
       "prediction": null,
       "project": "dummyProject",
       "series": "dummySeries",
       "smiles": "N%5C1%28C%28/N%3DC%28%5CC%28%3DC1%29%5
390         Cc1sccc1%29/N%29%3D0%29%5BC%40%40H%5D10%5BC%40%40H%5
           D%28C%28C1%290C%28%3D0%29c1cccc1%29C0C%28%3D0%29
           c1cccc1",
       "status": "Finished",
       "unit": ""
395     }
   ]
}

```

Listing 11: Output of the batchPredictionMV method.

1.7 Asynchronous execution

All four different methods of obtaining Chemics predictions, `predictions`, `predictionMV`, `batchPredictions` and `batchPredictionsMV`, have corresponding asynchronous methods, as defined by the list below.

- `startPrediction`
- `startPredictionMV`
- `startBatchPredictions`
- `startBatchPredictionsMV`
- `getStatus`
- `getPrediction`
- `getBatchPredictions`
- `jobCancellation`

The start methods return a job identifier, which can be used to check the status of the job with the `getStatus` method. The list below shows the six types of return codes from the `getStatus` method and one of them is variable (`SPECIFIC ERROR MESSAGE`) to proved the user with specific information about errors.

- Queued
- Running
- Completed: All molecules predicted successfully
- Incomplete: Some molecules could not be predicted. Please see the 'Calculation status' column. In case of errors, please report to Helpdesk providing the information in this box (Copy Summary To Clipboard).
- TASK FAILED: SPECIFIC ERROR MESSAGE + In case of errors, please report to Helpdesk providing the information in this box (Copy Summary To Clipboard).
- No job with this ID

Once the job is completed, as indicated by the "Completed" string being part of the return code, the results can be retrieved with the `getPrediction` or `getBatchPredictions` method. The input and output JSON objects are the same as for the synchronous methods. To avoid repetition, solely the execution of the asynchronous `batchPredictionsMV` method is illustrated below.

```

import string
import urllib
import requests
import json
5 import time

MYSERVER = "192.168.100.27:8081"
#MYSERVER = "chemics.medivir.com:8085"

10 def startBatchJob(MVDict, endpoint):
    url = 'http://'+MYSERVER+'/startBatchPredictionsMV/'+
        endpoint
    myData = json.dumps(MVDict)
    headers = {'Content-type': 'application/json', 'Accept': '
        text/plain'}
    response = requests.post(url, data=myData, headers =
        headers)
15     return json.loads(response.text)

def getBatchJobResult(jobID):
    url = 'http://'+MYSERVER+'/getBatchPredictions/'+jobID
    response = requests.get(url)
20     return json.loads(response.text)

def getStatus(jobID):
    url = 'http://'+MYSERVER+'/getStatus/'+jobID
    response = requests.get(url)
25     return json.loads(response.text)

def testBatchAsync(endpoint):

```



```

30 TIMEOUT = 30
   MVDict = [{ "ID" : "MV084958", "project" : "dummyProject",
               "series": "dummySeries"},
               {"ID" : "MV087263", "project" : "dummyProject",
               "series": "dummySeries"},
               {"ID" : "MV087333", "project" : "dummyProject",
               "series": "dummySeries"}]

   result = startBatchJob(MVDict, endpoint)
35 jobID = str(result["jobID"])
   runTime = 0
   while runTime < TIMEOUT:
       status = getStatus(jobID)
       print "Status ", status["jobStatus"]
40       if string.find(status["jobStatus"], "Completed") !=
          -1:
           runTime = TIMEOUT + 1
           result = getBatchJobResult(jobID)
       else:
           print "Job not finished. Waiting 5 s"
45           time.sleep(5)
           runTime = runTime + 5

       print json.dumps(result)

50 if __name__ == "__main__":
    #endpoint = "AllAPendpoints"
    endpoint = "logP"
    testBatchAsync(endpoint)

```

Listing 12: Calling the asynchronous batch method to obtain predictions from MV numbers.

```

Status Running
Job not finished. Waiting 5 s
Status Completed: All molecules predicted successfully
{"batchResults": [{"status": "Finished", "smiles": "c1cccc2c1C
%28%5CN%3DC%28/N2CCC%28C%29C%29%5CCN1C%28%3D0%29N%28
c2c1cncc2%29C1CCN%28S%28%3D0%29%28%3D0%29C%29CC1%29%3D0",
"confidence": "outAD", "endpoint": "logP", "descEndpoint":
"logP_AP7.1", "series": "dummySeries", "descStatus": "
Finished", "prediction": "1.87963071654801", "project": "
dummyProject", "ID": "MV084958", "unit": ""}, {"status": "
Finished", "smiles": "c1%28ccc2c%28c1%29cnc%28c2CCCC0%29
CN1c2c%28N%28C1%3D0%29C1CCN%28CC1%29C%28%3D0%290C%29ccnc2
%29CN", "confidence": "inAD", "endpoint": "logP", "
descEndpoint": "logP_AP7.1", "series": "dummySeries", "
descStatus": "Finished", "prediction": "2.51790987284744",
"project": "dummyProject", "ID": "MV087263", "unit": ""},
{"status": "Finished", "smiles": "n1n2c%28N%28c3c%28cccc3
%29C2%3D0%29Cc2nc%28c3cccc3n2%290%5BC%40%40H%5D2CCN%28C2
%29C%28%3D0%29C%29cc1C", "confidence": "inAD", "endpoint":
"logP", "descEndpoint": "logP_AP7.1", "series": "
dummySeries", "descStatus": "Finished", "prediction": "
2.0526111855064", "project": "dummyProject", "ID": "
MV087333", "unit": ""}]}

```

Listing 13: Output of the asynchronous execution.

1.8 Asynchronous execution with job cancellation

If a large job was accidentally submitted it can be cancelled with the `jobCancellation` method. The code in Listing 12 submits an asynchronous job requesting predictions for all ADMET Predictor endpoints for 1000 compounds, but cancels the job after approximately 40 seconds. Please note that the `startBatchPredictionsMV` method returns also a `APjobID` that is required as an input to `jobCancellation` to properly cancel an ADMET Predictor job.

```
import string
import urllib
import requests
import json
5 import time

MYSERVER = "192.168.100.27:8081"
#MYSERVER = "chemics.medivir.com:8085"

10 def startBatchJob(MVDict, endpoint):
    url = 'http://'+MYSERVER+'/startBatchPredictionsMV/'+
        endpoint
    myData = json.dumps(MVDict)
    headers = {'Content-type': 'application/json', 'Accept': '
        text/plain'}
    response = requests.post(url, data=myData, headers =
        headers)
15     return json.loads(response.text)

def getBatchJobResult(jobID):
    url = 'http://'+MYSERVER+'/getBatchPredictions/'+jobID
    response = requests.get(url)
20     return json.loads(response.text)

def getStatus(jobID):
    url = 'http://'+MYSERVER+'/getStatus/'+jobID
    response = requests.get(url)
25     return json.loads(response.text)

def cancelJob(jobID, APjobID = None):
    url = 'http://'+MYSERVER+'/jobCancellation/'+jobID+"/"+
        APjobID
30     print url
    response = requests.get(url)
    return json.loads(response.text)

35 def getMVlist():
    fileName = "AllMVnumbersSMILES_1200_nofrag.txt"
    fid = open(fileName)
    MVlist = []
```

```

40     for line in fid:
        lineList = string.split(line)
        smiles = lineList[0]
        ID = string.strip(lineList[1])
        MVlist.append(ID)
        fid.close()
45     return MVlist

def testBatchAsyncCancel(endpoint):

50     TIMEOUT = 600
        CANCELTIMEOUT = 10

        # Get molecules
        MVlist = getMVlist()
        smilesDict = []
55     for elem in MVlist:
        smilesDict.append({"ID" :elem, "project" : "
            DummyProject", "series" : "DummySeries"})
        print "Number of submitted molecules ", len(smilesDict)

60     # Start the job
        result = startBatchJob(smilesDict, endpoint)
        jobID = result["jobID"]
        APjobID = result["APjobID"]
        print "Result from startBatch ", jobID, APjobID

65     # Check status
        runTime = 0
        while runTime < TIMEOUT:
            status = getStatus(jobID)
70            print "status", status
            if string.find(status["jobStatus"], "Completed") !=
                -1:
                runTime = TIMEOUT + 1
                result = getBatchJobResult(jobID)
            elif status["jobStatus"] != "No job with this ID":
75                print "Job not finished. Waiting 5 s"
                time.sleep(5)
                runTime = runTime + 5
                if runTime > CANCELTIMEOUT:
                    print "Cancelling job"
                    cancelResp = cancelJob(jobID, APjobID)
                    print cancelResp
                    status = getStatus(jobID)
                    print "status ", status
                    runTime = TIMEOUT + 1
80                else:
                    runTime = TIMEOUT + 1

            print json.dumps(result)

90 if __name__ == "__main__":
        endpoint = "AllAPendpoints"
        #endpoint = "logP"

```

```
testBatchAsyncCancel(endpoint)
```

Listing 14: Calling the asynchronous batch method and cancelling the job after 40 seconds.

```

Number of submitted molecules 1200
Result from startBatch 6c2b5962-9242-4ac8-8597-613d6d6ad360
  jobID1456310536
status {u'jobStatus': u'Running'}
Job not finished. Waiting 5 s
5 status {u'jobStatus': u'Running'}
Job not finished. Waiting 5 s
status {u'jobStatus': u'Running'}
Job not finished. Waiting 5 s
Cancelling job
10 http://192.168.100.27:8081/jobCancellation/6c2b5962-9242-4ac8
  -8597-613d6d6ad360/jobID1456310536
{u'jobStatus': u'Job cancelled'}
status {u'jobStatus': u'No job with this ID'}
{"APjobID": "jobID1456310536", "jobID": "6c2b5962-9242-4ac8
  -8597-613d6d6ad360"}

```

Listing 15: Output of the asynchronous execution with job cancellation.

2 Appendix

```

http://192.168.100.27:8081/D360endpoints
{
  "endpoints": {
    "displayname": "Chemics Service",
    "name": "Chemics Service",
    5 "outputProperties": null,
    "subFolders": [
      {
        "displayname": "General",
        "name": "General",
        10 "outputProperties": null,
        "subFolders": [
          {
            "display": true,
            "displayname": "SMILES_RDK12.12.1",
            "name": "SMILES",
            "outputProperties": [
              {
                "dataType": "string",
                "descriptorId": "prediction",
                "displayname": "SMILES_RDK12.12.1",
                "selectedByDefault": true,
                "unit": null
              },
              {
                25 "dataType": "string",
                "descriptorId": "descStatus",

```

```

        "displayname": "Calculation status SMILES",
        "selectedByDefault": true,
        "unit": null
    },
    ],
    "subFolders": null,
    "type": "task",
    "version": "RDKit12.12.1"
},
{
    "display": true,
    "displayname": "MolWt_RDK12.12.1",
    "name": "MolWt",
    "outputProperties": [
        {
            "dataType": "float",
            "descriptorId": "prediction",
            "displayname": "MolWt_RDK12.12.1 (g/mol)",
            "selectedByDefault": true,
            "unit": "g/mol"
        },
        {
            "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status MolWt",
            "selectedByDefault": true,
            "unit": null
        }
    ],
    "subFolders": null,
    "type": "task",
    "version": "RDKit12.12.1"
},
{
    "type": "folder",
    "version": null
},
{
    "displayname": "Counts",
    "name": "Counts",
    "outputProperties": null,
    "subFolders": [
        {
            "display": true,
            "displayname": "HeavyAtomCount_RDK12.12.1",
            "name": "HeavyAtomCount",
            "outputProperties": [
                {
                    "dataType": "int",
                    "descriptorId": "prediction",
                    "displayname": "HeavyAtomCount_RDK12.12.1",
                    "selectedByDefault": true,
                    "unit": null
                }
            ],
            {
                "dataType": "string",
                "descriptorId": "descStatus",

```

```

85         "displayname": "Calculation status
           HeavyAtomCount",
           "selectedByDefault": true,
           "unit": null
       }
   ],
90   "subFolders": null,
   "type": "task",
   "version": "RDKit2.12.1"
},
{
95   "display": true,
   "displayname": "BondCount_RDKit2.12.1",
   "name": "BondCount",
   "outputProperties": [
100     {
       "dataType": "int",
       "descriptorId": "prediction",
       "displayname": "BondCount_RDKit2.12.1",
       "selectedByDefault": true,
       "unit": null
105     },
     {
       "dataType": "string",
       "descriptorId": "descStatus",
       "displayname": "Calculation status BondCount",
110       "selectedByDefault": true,
       "unit": null
     }
   ],
   "subFolders": null,
115   "type": "task",
   "version": "RDKit2.12.1"
},
{
120   "display": true,
   "displayname": "FluorineCount_RDKit2.12.1",
   "name": "FluorineCount",
   "outputProperties": [
125     {
       "dataType": "int",
       "descriptorId": "prediction",
       "displayname": "FluorineCount_RDKit2.12.1",
       "selectedByDefault": true,
       "unit": null
130     },
     {
       "dataType": "string",
       "descriptorId": "descStatus",
       "displayname": "Calculation status
           FluorineCount",
       "selectedByDefault": true,
135       "unit": null
     }
   ],
   "subFolders": null,
   "type": "task",

```

```

140     "version": "RDKit12.12.1"
    },
    {
145       "display": true,
       "displayname": "HalogenCount_RDKit12.12.1",
       "name": "HalogenCount",
       "outputProperties": [
150         {
           "dataType": "int",
           "descriptorId": "prediction",
           "displayname": "HalogenCount_RDKit12.12.1",
           "selectedByDefault": true,
           "unit": null
155         },
         {
           "dataType": "string",
           "descriptorId": "descStatus",
           "displayname": "Calculation status
             HalogenCount",
           "selectedByDefault": true,
           "unit": null
160         }
       ],
       "subFolders": null,
       "type": "task",
       "version": "RDKit12.12.1"
165     },
    {
       "display": true,
       "displayname": "CarbonCount_RDKit12.12.1",
       "name": "CarbonCount",
       "outputProperties": [
170         {
           "dataType": "int",
           "descriptorId": "prediction",
           "displayname": "CarbonCount_RDKit12.12.1",
           "selectedByDefault": true,
           "unit": null
175         },
         {
           "dataType": "string",
           "descriptorId": "descStatus",
           "displayname": "Calculation status CarbonCount
             ",
           "selectedByDefault": true,
           "unit": null
180         }
       ],
       "subFolders": null,
       "type": "task",
       "version": "RDKit12.12.1"
185     },
    {
190       "display": true,
       "displayname": "PhosphorusCount_RDKit12.12.1",
       "name": "PhosphorusCount",
       "outputProperties": [

```

```

195         {
            "dataType": "int",
            "descriptorId": "prediction",
            "displayname": "PhosphorusCount_RDK12.12.1",
            "selectedByDefault": true,
200         "unit": null
        },
        {
            "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status
205             PhosphorusCount",
            "selectedByDefault": true,
            "unit": null
        }
    ],
    "subFolders": null,
210    "type": "task",
    "version": "RD12.12.1"
},
{
215    "display": true,
    "displayname": "ChlorineCount_RDK12.12.1",
    "name": "ChlorineCount",
    "outputProperties": [
        {
220            "dataType": "int",
            "descriptorId": "prediction",
            "displayname": "ChlorineCount_RDK12.12.1",
            "selectedByDefault": true,
            "unit": null
225        },
        {
            "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status
                ChlorineCount",
230            "selectedByDefault": true,
            "unit": null
        }
    ],
    "subFolders": null,
235    "type": "task",
    "version": "RD12.12.1"
},
{
240    "display": true,
    "displayname": "OxygenCount_RDK12.12.1",
    "name": "OxygenCount",
    "outputProperties": [
        {
245            "dataType": "int",
            "descriptorId": "prediction",
            "displayname": "OxygenCount_RDK12.12.1",
            "selectedByDefault": true,
            "unit": null
        }
    ],
    "subFolders": null,
    "type": "task",
    "version": "RD12.12.1"
},
{
    "display": true,
    "displayname": "PhosphorusCount_RDK12.12.1",
    "name": "PhosphorusCount",
    "outputProperties": [
        {
            "dataType": "int",
            "descriptorId": "prediction",
            "displayname": "PhosphorusCount_RDK12.12.1",
            "selectedByDefault": true,
            "unit": null
        }
    ],
    "subFolders": null,
    "type": "task",
    "version": "RD12.12.1"
}
]
}

```



```

250         {
            "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status OxygenCount",
            "selectedByDefault": true,
255         "unit": null
        }
    ],
    "subFolders": null,
    "type": "task",
260     "version": "RDKit12.12.1"
},
{
    "display": true,
    "displayname": "TPSA_RDK12.12.1",
265     "name": "TPSA",
    "outputProperties": [
        {
            "dataType": "int",
            "descriptorId": "prediction",
270             "displayname": "TPSA_RDK12.12.1",
            "selectedByDefault": true,
            "unit": null
        },
        {
275             "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status TPSA",
            "selectedByDefault": true,
            "unit": null
280         }
    ],
    "subFolders": null,
    "type": "task",
    "version": "RDKit12.12.1"
285 },
{
    "display": true,
    "displayname": "HAcceptorsCount_RDK12.12.1",
    "name": "HAcceptorsCount",
290     "outputProperties": [
        {
            "dataType": "int",
            "descriptorId": "prediction",
            "displayname": "HAcceptorsCount_RDK12.12.1",
295             "selectedByDefault": true,
            "unit": null
        },
        {
            "dataType": "string",
300             "descriptorId": "descStatus",
            "displayname": "Calculation status
                HAcceptorsCount",
            "selectedByDefault": true,
            "unit": null
        }
    ]
}

```

```

305     ],
    "subFolders": null,
    "type": "task",
    "version": "RDKit12.12.1"
  },
310  {
    "display": true,
    "displayname": "HDonorsCount_RDKit12.12.1",
    "name": "HDonorsCount",
    "outputProperties": [
315      {
        "dataType": "int",
        "descriptorId": "prediction",
        "displayname": "HDonorsCount_RDKit12.12.1",
        "selectedByDefault": true,
320        "unit": null
      },
      {
        "dataType": "string",
        "descriptorId": "descStatus",
        "displayname": "Calculation status
325        HDonorsCount",
        "selectedByDefault": true,
        "unit": null
      }
    ],
    "subFolders": null,
330    "type": "task",
    "version": "RDKit12.12.1"
  },
  {
335    "display": true,
    "displayname": "RingCount_RDKit12.12.1",
    "name": "RingCount",
    "outputProperties": [
340      {
        "dataType": "int",
        "descriptorId": "prediction",
        "displayname": "RingCount_RDKit12.12.1",
        "selectedByDefault": true,
345        "unit": null
      },
      {
        "dataType": "string",
        "descriptorId": "descStatus",
        "displayname": "Calculation status RingCount",
350        "selectedByDefault": true,
        "unit": null
      }
    ],
    "subFolders": null,
355    "type": "task",
    "version": "RDKit12.12.1"
  },
  {
    "display": true,
360    "displayname": "RotatableBondsCount_RDKit12.12.1",

```

```

365     "name": "RotatableBondsCount",
        "outputProperties": [
            {
                "dataType": "int",
                "descriptorId": "prediction",
                "displayname": "RotatableBondsCount_RDK12.12.1",
                "selectedByDefault": true,
                "unit": null
            },
            {
370                "dataType": "string",
                "descriptorId": "descStatus",
                "displayname": "Calculation status
                    RotatableBondsCount",
                "selectedByDefault": true,
375                "unit": null
            }
        ],
        "subFolders": null,
        "type": "task",
380        "version": "RDKit12.12.1"
    },
    {
        "display": true,
        "displayname": "NitrogenCount_RDK12.12.1",
385        "name": "NitrogenCount",
        "outputProperties": [
            {
                "dataType": "int",
                "descriptorId": "prediction",
                "displayname": "NitrogenCount_RDK12.12.1",
                "selectedByDefault": true,
                "unit": null
            },
            {
395                "dataType": "string",
                "descriptorId": "descStatus",
                "displayname": "Calculation status
                    NitrogenCount",
                "selectedByDefault": true,
                "unit": null
            }
400        ],
        "subFolders": null,
        "type": "task",
        "version": "RDKit12.12.1"
405    },
    {
        "display": true,
        "displayname": "SulfurCount_RDK12.12.1",
        "name": "SulfurCount",
410        "outputProperties": [
            {
                "dataType": "int",
                "descriptorId": "prediction",
                "displayname": "SulfurCount_RDK12.12.1",

```

```

415         "selectedByDefault": true,
            "unit": null
        },
        {
420             "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status SulfurCount",
            "selectedByDefault": true,
            "unit": null
        }
425    ],
    "subFolders": null,
    "type": "task",
    "version": "RDKit12.12.1"
},
430 {
    "display": true,
    "displayname": "AtomCount_RDKit12.12.1",
    "name": "AtomCount",
    "outputProperties": [
435         {
            "dataType": "int",
            "descriptorId": "prediction",
            "displayname": "AtomCount_RDKit12.12.1",
            "selectedByDefault": true,
440             "unit": null
        },
        {
            "dataType": "string",
            "descriptorId": "descStatus",
445             "displayname": "Calculation status AtomCount",
            "selectedByDefault": true,
            "unit": null
        }
    ],
450    "subFolders": null,
    "type": "task",
    "version": "RDKit12.12.1"
}
455 ],
    "type": "folder",
    "version": null
},
460 {
    "displayname": "AllAPendpoints",
    "name": "AllAPendpoints",
    "outputProperties": null,
    "subFolders": [
465         {
            "display": true,
            "displayname": "AllAPendpoints_AP7.1",
            "name": "AllAPendpoints",
            "outputProperties": [
470                 {
                    "dataType": "string",
                    "descriptorId": "prediction",

```

```

    "displayname": "AllAPendpoints_AP7.1",
    "selectedByDefault": true,
    "unit": null
  },
  {
    "dataType": "string",
    "descriptorId": "descStatus",
    "displayname": "Calculation status
      AllAPendpoints",
    "selectedByDefault": true,
    "unit": null
  }
],
"subFolders": null,
"type": "task",
"version": "AP7.1"
}
],
"type": "folder",
"version": null
},
{
  "displayname": "PhysChem",
  "name": "PhysChem",
  "outputProperties": null,
  "subFolders": [
    {
      "displayname": "Lipophilicity",
      "name": "Lipophilicity",
      "outputProperties": null,
      "subFolders": [
        {
          "display": false,
          "displayname": "logP_AP7.1",
          "name": "logP",
          "outputProperties": [
            {
              "dataType": "float",
              "descriptorId": "prediction",
              "displayname": "logP_AP7.1",
              "selectedByDefault": true,
              "unit": null
            },
            {
              "dataType": "string",
              "descriptorId": "confidence",
              "displayname": "logP_AP7.1 Confidence",
              "selectedByDefault": true,
              "unit": null
            }
          ]
        }
      ],
      "subFolders": null,
      "type": "task",
      "version": "AP7.1"
    },
    {
      "display": false,

```

```

    "displayname": "logD_AP7.1",
    "name": "logD",
    "outputProperties": [
530     {
        "dataType": "float",
        "descriptorId": "prediction",
        "displayname": "logD_AP7.1",
        "selectedByDefault": true,
535         "unit": null
    },
    {
        "dataType": "string",
        "descriptorId": "confidence",
540         "displayname": "logD_AP7.1 Confidence",
        "selectedByDefault": true,
        "unit": null
    }
    ],
545     "subFolders": null,
    "type": "task",
    "version": "AP7.1"
  }
],
550   "type": "folder",
  "version": null
},
{
  "displayname": "Solubility",
555   "name": "Solubility",
  "outputProperties": null,
  "subFolders": [
    {
      "display": false,
560       "displayname": "Sp_AP7.1",
      "name": "Sp",
      "outputProperties": [
        {
          "dataType": "float",
565           "descriptorId": "prediction",
          "displayname": "Sp_AP7.1 (mg/mL)",
          "selectedByDefault": true,
          "unit": "mg/mL"
        },
        {
570           "dataType": "string",
          "descriptorId": "confidence",
          "displayname": "Sp_AP7.1 Confidence",
          "selectedByDefault": true,
          "unit": null
575        }
      ]
    },
    {
      "subFolders": null,
580       "type": "task",
      "version": "AP7.1"
    }
  ]
},
  "type": "folder",

```

```

    "version": null
  },
  {
    "displayname": "pKa",
    "name": "pKa",
    "outputProperties": null,
    "subFolders": [
      {
        "display": false,
        "displayname": "Acidic_pKa_AP7.1",
        "name": "Acidic_pKa",
        "outputProperties": [
          {
            "dataType": "string",
            "descriptorId": "prediction",
            "displayname": "Acidic_pKa_AP7.1",
            "selectedByDefault": true,
            "unit": null
          }
        ],
        "subFolders": null,
        "type": "task",
        "version": "AP7.1"
      },
      {
        "display": false,
        "displayname": "Acidic_pKa_74prox_AP7.1",
        "name": "Acidic_pKa_74prox",
        "outputProperties": [
          {
            "dataType": "float",
            "descriptorId": "prediction",
            "displayname": "Acidic_pKa_74prox_AP7.1",
            "selectedByDefault": true,
            "unit": null
          }
        ],
        "subFolders": null,
        "type": "task",
        "version": "AP7.1"
      },
      {
        "display": false,
        "displayname": "Basic_pKa_AP7.1",
        "name": "Basic_pKa",
        "outputProperties": [
          {
            "dataType": "string",
            "descriptorId": "prediction",
            "displayname": "Basic_pKa_AP7.1",
            "selectedByDefault": true,
            "unit": null
          }
        ],
        "subFolders": null,
        "type": "task",
        "version": "AP7.1"
      }
    ]
  }
]

```

```

    },
    {
      "display": false,
      "displayname": "Basic_pKa_74prox_AP7.1",
      "name": "Basic_pKa_74prox",
      "outputProperties": [
        {
          "dataType": "float",
          "descriptorId": "prediction",
          "displayname": "Basic_pKa_74prox_AP7.1",
          "selectedByDefault": true,
          "unit": null
        }
      ],
      "subFolders": null,
      "type": "task",
      "version": "AP7.1"
    },
    {
      "display": false,
      "displayname": "Mixed_pKa_AP7.1",
      "name": "Mixed_pKa",
      "outputProperties": [
        {
          "dataType": "string",
          "descriptorId": "prediction",
          "displayname": "Mixed_pKa_AP7.1",
          "selectedByDefault": true,
          "unit": null
        }
      ],
      "subFolders": null,
      "type": "task",
      "version": "AP7.1"
    },
    {
      "display": false,
      "displayname": "Mixed_pKa_74prox_AP7.1",
      "name": "Mixed_pKa_74prox",
      "outputProperties": [
        {
          "dataType": "float",
          "descriptorId": "prediction",
          "displayname": "Mixed_pKa_74prox_AP7.1",
          "selectedByDefault": true,
          "unit": null
        }
      ],
      "subFolders": null,
      "type": "task",
      "version": "AP7.1"
    },
    {
      "display": false,
      "displayname": "pKa_mostBasic_AP7.1",
      "name": "pKa_mostBasic",
      "outputProperties": [

```



```

700         {
            "dataType": "float",
            "descriptorId": "prediction",
            "displayname": "pKa_mostBasic_AP7.1",
            "selectedByDefault": true,
            "unit": null
        },
705         {
            "dataType": "string",
            "descriptorId": "descStatus",
            "displayname": "Calculation status
                          AllAPendpoints",
            "selectedByDefault": true,
710            "unit": null
        }
    ],
    "subFolders": null,
    "type": "task",
715    "version": "AP7.1"
},
{
    "display": false,
    "displayname": "pKa_mostAcidic_AP7.1",
720    "name": "pKa_mostAcidic",
    "outputProperties": [
        {
            "dataType": "float",
            "descriptorId": "prediction",
725            "displayname": "pKa_mostAcidic_AP7.1",
            "selectedByDefault": true,
            "unit": null
        }
    ],
730    "subFolders": null,
    "type": "task",
    "version": "AP7.1"
}
],
735    "type": "folder",
    "version": null
},
{
    "displayname": "Rules",
740    "name": "Rules",
    "outputProperties": null,
    "subFolders": [
        {
            "display": false,
            "displayname": "RuleOf3_AP7.1",
745            "name": "RuleOf3",
            "outputProperties": [
                {
                    "dataType": "int",
                    "descriptorId": "prediction",
                    "displayname": "RuleOf3_AP7.1",
                    "selectedByDefault": true,
750                    "unit": null
                }
            ]
        }
    ]
}

```

```

    },
    "subFolders": null,
    "type": "task",
    "version": "AP7.1"
  },
  {
    "display": false,
    "displayname": "RuleOf5_AP7.1",
    "name": "RuleOf5",
    "outputProperties": [
      {
        "dataType": "int",
        "descriptorId": "prediction",
        "displayname": "RuleOf5_AP7.1",
        "selectedByDefault": true,
        "unit": null
      }
    ],
    "subFolders": null,
    "type": "task",
    "version": "AP7.1"
  }
],
"type": "folder",
"version": null
},
],
"type": "folder",
"version": null
},
{
  "displayname": "DMPK",
  "name": "DMPK",
  "outputProperties": null,
  "subFolders": [
    {
      "displayname": "Permeability",
      "name": "Permeability",
      "outputProperties": null,
      "subFolders": [
        {
          "display": false,
          "displayname": "MDCK_AP7.1",
          "name": "MDCK",
          "outputProperties": [
            {
              "dataType": "float",
              "descriptorId": "prediction",
              "displayname": "MDCK_AP7.1 (cm/s*10^7)",
              "selectedByDefault": true,
              "unit": "cm/s*10^7"
            }
          ],
          {
            "dataType": "string",
            "descriptorId": "confidence",
            "displayname": "MDCK_AP7.1 Confidence",

```

```

        "selectedByDefault": true,
        "unit": null
    }
    ],
    "subFolders": null,
    "type": "task",
    "version": "AP7.1"
},
{
    "display": false,
    "displayname": "Peff_AP7.1",
    "name": "Peff",
    "outputProperties": [
        {
            "dataType": "float",
            "descriptorId": "prediction",
            "displayname": "Peff_AP7.1 (cm/s*10^4)",
            "selectedByDefault": true,
            "unit": "cm/s*10^4"
        },
        {
            "dataType": "string",
            "descriptorId": "confidence",
            "displayname": "Peff_AP7.1 Confidence",
            "selectedByDefault": true,
            "unit": null
        }
    ],
    "subFolders": null,
    "type": "task",
    "version": "AP7.1"
}
],
    "type": "folder",
    "version": null
}
],
    "type": "folder",
    "version": null
}
],
    "type": "folder",
    "version": null
}
}
}

```

Listing 16: Output of the D360endpoints method.