ICAGEN
Advancing Early Drug Discovery

Kinetic Solubility: Measurement and Data Processing

Estelle Maes*, Victor Nikolaev, and Marcel Patek†

Icagen
Analytical Department,
2090 E. Innovation Park Drive,
Oro Valley, AZ 85755

Date: October 16, 2017

Abstract

This note describes the method of kinetic solubility as adapted in Icagen's eADME flow. To enable estimation of sample dissolution for a higher number of compounds, we chose a high-throughput workflow that utilizes LC-MS and light scattering detector (ELSD) for sample analysis and quantitation, respectively. Icagen's typical implementation includes measurement of kinetic solubility in phosphate buffer at pH 7.4 directly from 10 mM DMSO stock solution. For data processing, report generation, and to provide our customers with the detailed visual representation of data, we adapted the Jupyter Notebook supported by Python programming language.

*Contact: emaes@icagen.com, 520-544-6871
†Contact: mpatek@icagen.com, 520-544-6850

1 Aqueous solubility

Aqueous solubility is a critical physicochemical parameter in drug discovery that benefits from an early assessment [1, 2]. After several rounds of sample distribution, serial dilutions, and freeze-thaw cycles, actual sample concentration becomes of high importance. Solubility profile for larger sets of compounds is typically predicted computationally rather than measured experimentally. While *in silico* predictions are sufficient for the design of compound screening collections, *in vitro* assay validation, high-throughput screening campaigns, and hit confirmation benefit from measured values. Indeed, with a wide distribution of sample concentrations, screening results show erratic patterns and can produce a significant number of false positives or negatives. Consequently, inaccurate structure-activity relationship (SAR) and erroneous ADME/Tox data usually parallel compounds of low solubility [3].

At the early stage of hit and lead discovery, it is the kinetic solubility method that is primarily used. In this context, kinetic solubility can be defined as the concentration where the compound precipitates upon addition of a DMSO solution into an aqueous buffer [4]. The conditions of the kinetic solubility tests mimic conventional *in vitro* discovery screening assays in that compounds are sourced from the same DMSO stock solution. Kinetic solubility is used to rank-order hits, to flag compounds with potential liabilities, and to validate hits by comparing compound dose-response values with their apparent solubility values.

Icagen offers kinetic solubility measurements in a variety of aqueous solutions, such as pH 7.4 phosphate buffer using 10 mM DMSO solution of the sample of interest. We use the high throughput screening (HTS) 96-format assay with LC/MS for analysis and evaporative light scattering detector (ELSD) for quantitation. Agilent 1200 HPLC system coupled with Agilent 6140 or 6150 single quadrupole mass-spectrometer provide an adequate solution for sample characterization [5].

2 Kinetic solubility protocol

To enable high-throughput measurement of kinetic solubility on hundreds of samples per week, we implemented hardware setup based on Agilent 1200 LC-MS. The system includes autosampler and ELSD detector. LC-MS data is processed after the

run in Analytical Studio from Virscidian [1].
Tests are performed in shallow 96-well plates with samples prepared in local Compound Logistics group.

1. Distribute 4 µL of 10mM DMSO stock solution for each compound in two plate copies (2 x 4 µL).
2. Dilute the sample in one plate with 196 µL of 100mM phosphate buffer pH 7.4 (system **A**).
3. Dilute the sample in the second plate with 196 µL of H_2O : CH_3OH : iPrOH (2:3:1) (system **B**, "organic solvent").
4. Sonicate both plates at room temperature for 15min.
5. Centrifuge both plates for 15 min at room temperature.
6. Run LC-MS analysis from both plates; samples of the same compound from each plate are run consecutively.
7. ELSD response is used to quantify the compound (as standard _C) in the organic solvent mixture (sample source).
8. UV(220 nm) signals of the same compound from the phosphate buffer (**A**) and solution **B**, together with the ELSD quantification are being used to calculate kinetic solubility. The Compound is assumed to dissolve 100% in the system **B**.

Typical LCMS conditions include Zorbax Eclipse XDB-C18 RRHT column, 1.8 µm, 2.1 x 30 mm and gradient from 2% to 100% ACN (+ 0.08% TFA) in H_2O (+ 0.1% TFA) over 3.0 min at flow-rate of 0.8 mL/min.

3 Data processing

We implemented Jupyter Notebook with Python programming language to document workflows and to share data processing, analysis, and visualization outputs [6].

[1] `http://www.virscidian.com/workflows/medicinal-chemistry/automated-compound-qc/`

The Jupyter Notebook is an open source application that runs in a web browser[2]. The Notebook combines code in the language of choice (e.g., Python, R, MATLAB) and allows creating and sharing documents that contain live code, equations, visualizations, and explanatory text. The Notebook is interactive so that one can execute code directly from a web browser. Our implementation utilizes Anaconda Python version 3, which can be downloaded from the Continuum Analytics website [3].

3.1 From values to results

Assessment of the compound concentration starts from the calibration curve of a standard of known amount dissolved in DMSO. The determination of such calibration curve is not in the scope of this note and several calibration files typically exist for different concentration ranges. Jupyter Notebook provides several widgets that allow selection of calibration file matching the corresponding instrument and concentration range.

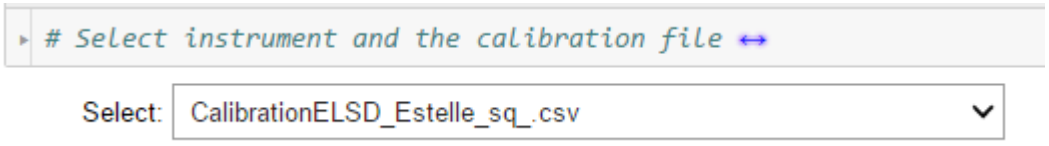


Figure 1: File selection widget in Jupyter Notebook.

Calibration file is a standard *csv* file of the structure shown in **Table 1**. "Inj (ng)" stands for the amount of injected standard and "ELS" refers to the response of ELSD detector (digital units). Linearity of the sample amount vs. ELSD response is expected in the log domain. Accordingly, values in each column ("Inj ng", "ELS") are transformed to log10 values. For linear regression analysis, we use Python package *Statsmodels* to find the best fit of sample amount to ELS response. The corresponding relationship follows in **eq (1)** :

$$ELS = \mathbf{k} \cdot Inj + \mathbf{b} \tag{1}$$

[2] http://jupyter.org/
[3] http://continuum.io/downloads

Inj (ng)	ELS
31	1.14
63	3.64
125	9.68
250	28.17
500	104.86
999	325.41

Table 1: Calibration curve data for sample amount (ng) vs ELS response

where **k** = slope and **b** = intercept of the regression line.
The following Python code outlines the key instructions to retrieve **k** and **b** parameters from the linear regression.

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

# Statsmodels linear regression
res = ols("logELS~logInj", data=cal).fit()
res_sum = res.summary()

# Get the slope and intercept as in y = kx +b
b = res.params.Intercept
k = res.params.logInj
```

Besides the slope and intercept, *statsmodels* package also returns four-pannel plot of regression details for a quick visual inspection of goodness of the data fit (**Fig 2**). Complete set of regression parameters is available and will not be used in this example. For illustration, ELS fit and residuals are shown in Table **2**.

In this particular case, the slope $\mathbf{k} = 1.624 \pm 0.033$ and $\mathbf{b} = -2.385 \pm 0.076$ With the key values in hand, we will approach calculations of the solubility based on experimental data. Selected columns and rows of the raw data file exported from Analytical Studio are shown in Table **3**.
Injection volume of the measured sample and its dilution before injection into LC-MS system are entered into Jupyter Notebook interactive widget as shown in Figure **3**. As outlined in the section **2**, each sample is measured in two conditions. These are

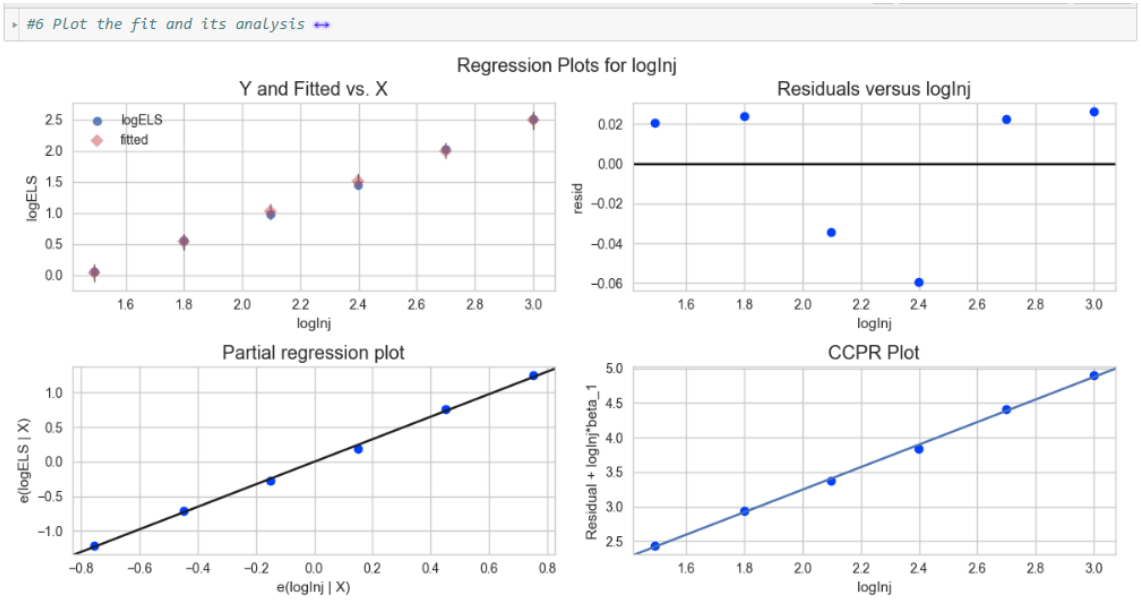


Figure 2: Regression plots.

Inj(logX)	ELS(logY)	fitted_Y	resid_Y	std_resid_Y
1.49	0.06	0.04	0.02	0.50
1.80	0.56	0.54	0.02	0.57
2.10	0.99	1.02	-0.03	-0.83
2.40	1.45	1.51	-0.06	-1.43
2.70	2.02	2.00	0.02	0.55
3.00	2.51	2.49	0.03	0.64

Table 2: Linear regression values with the corresponding statistics

referred to as $\_T$ (Target in buffer) and $\_C$ (Compound in an ideal solvent).

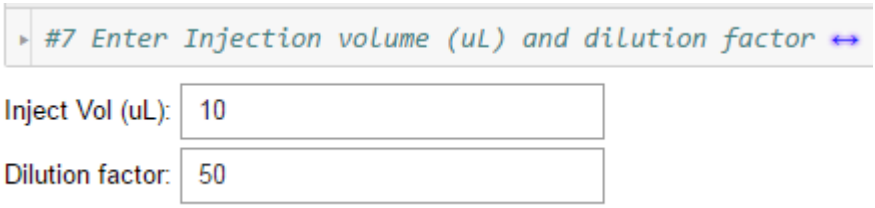


Figure 3: Fields for entering Vol (µL) and sample dilution.

Amount of sample ($\_C$) in (ng) is calculated using **2a** and converted to concen-

SampleName	Project	Formula	Mass	Found	RT(min)	QuantAbs	Purity%	Quant%	Comments	UV220_AreaAbs <1>
YG142SJL_T	companyA	C45H62N4O10	818.40	Yes	2.021	421.34	97.04	6.23	NaN	1404.59
ZS127HKH_C	companyA	C47H58F2N4O9	860.40	Yes	2.236	586.75	87.28	97.82	NaN	1622.99
ZS127HKH_T	companyA	C47H58F2N4O9	860.40	Yes	2.24	0.00	100.00	0.00	NaN	11.88
ZW037ZAW_C	companyA	C51H66N4O9	878.50	Yes	2.43	684.29	100.00	100.00	not detected in buffer	2039.74
ZW037ZAW_T	companyA	C51H66N4O9	878.50	No	ND	0.00	0.00	0.00	not detected in buffer	ND

Table 3: Sample of raw data from Analytical Studio

tration (mM) using eq. **2b**.

$$Amnt\_in\_ng = 10^{(log(QuantAbs) - b)/k} \qquad [ng] \tag{2a}$$

$$Injection\_mM(\_C) = \frac{Amnt\_in\_ng}{(inj\_vol \cdot M_w)} \qquad [mM] \tag{2b}$$

Relationship between peak areas UV (220 nm) of the compound dissolved in ideal solvent vs. buffer and the amount of injected compound is described in the key expression **(2c)**.

$$\frac{UV220\_AreaAbs\_C}{UV220\_AreaAbs\_T} = \frac{Injection\_mM(\_C)}{Sol220\mu M} \tag{2c}$$

From 2c, solubility (µM) at 220 nm can be calculated from eq. **(2d)**.

$$Sol220\mu M = 10^3 \cdot Injection\_mM \cdot \frac{UV220\_AreaAbs\_T}{UV220\_AreaAbs\_C} \qquad [\mu M] \tag{2d}$$

With only a few exceptions (referred to as "solubility reversal"), values for samples $\_C$ are greater than values for target sample $\_T$, the latter referring to a mixture of H_2O : CH_3OH : iPrOH (2:3:1). The fraction term in eq. **2d**, if greater than 1, represents the Sol_reversal term. In such cases, sample solubility is set to values in the system $\_T$.

$$Sol220\mu M_{max} = Sol220\mu M \cdot \frac{1}{Sol\_reversal} \tag{3}$$

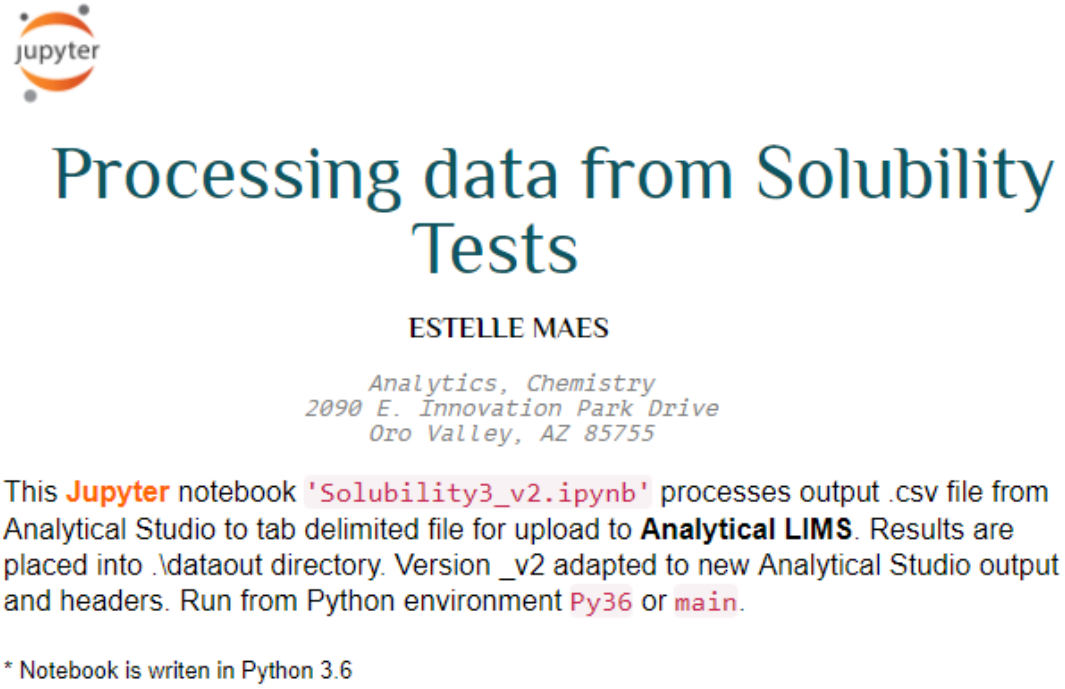
Processed data with calculated values are shown in Table **4**.

SampleName	Project	Sol220uM	Sol_ug_mL	Sol_reversal	Comments
AL931HSF	companyA	0.99	0.86	0.01	NaN
AX271LTF	companyA	nan	nan	nan	not detected in buffer
BP841AGU	companyA	1.54	1.35	0.01	NaN
BR006GCQ	companyA	17.09	13.96	0.10	NaN
CL741BLZ	companyA	nan	nan	nan	not detected in buffer
DP720ZZA	companyA	45.37	36.77	0.33	NaN

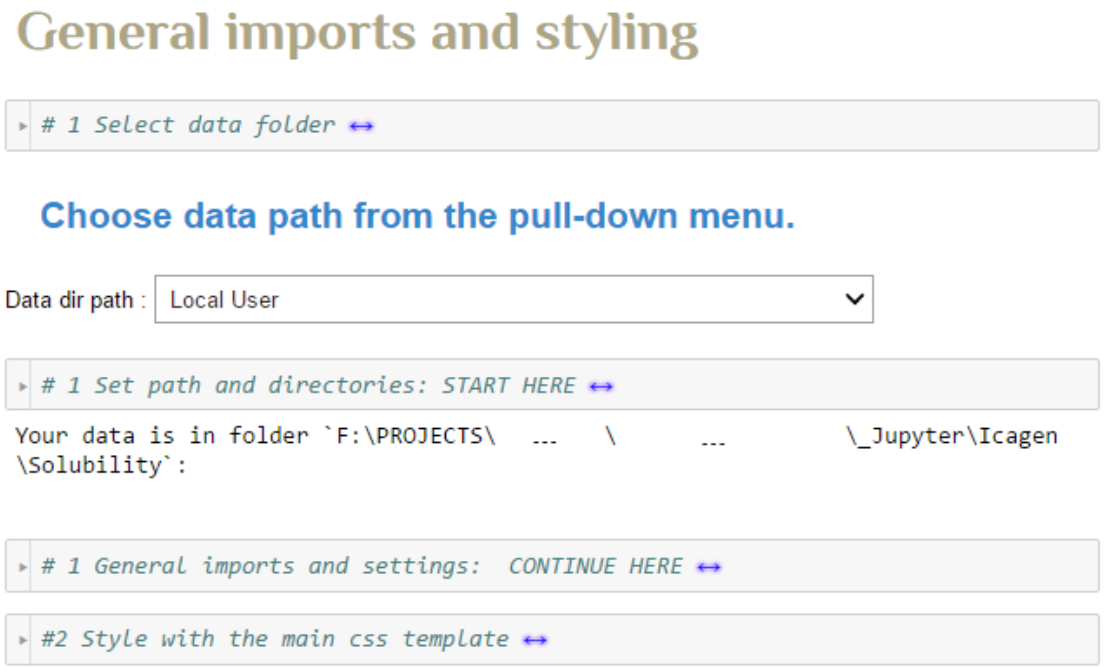
Table 4: Processed solubility data.

4 Jupyter Notebook implementation

In this section, we present multiple snapshots of "Solubility.ipynb" notebook in a sequence that corresponds to the steps described in section 3.1. Each notebook has an introductory description and details on how to proceed with the cell execution.



The Analyst can select the *path* to the result files from the pull-down menu and to direct Notebook to folder structure of choice.
Next few cells execute loading the necessary Python packages and style sheets. Since the Notebook is rendered in browser as .html file, visual presentation can be further improved by additional HTML and CSS markup. Descriptive text in the Notebook is based on Markdown language and it is rendered upon the first cell execution. Notebook section "Enter Project, method, and sample information" (below) pro-



vides interactive fields to select the result file name and sample analysis parameters (e.g., injected volume, dilution). An abbreviated data table is first retrieved to double-check the input file consistency (# 4). Data table with calculated values is subsequently created and provides first view of Solubility results (# 8).
To inspect the whole batch of compounds, plot of solubility values for each Batch_id is created next. The Analyst can better appreciate solubility profile across the set and to identify samples that require re-run. Table of samples that show "solubility reversal" and list of poorly soluble compounds are created as well (# 9).

Enter Project, method, and sample information

Start with files and processing

#3 Select Results_ file to process ↔

Select: 20170907_SOL_46845_cur.csv

#4 Read raw csv file from Analytical Studio ↔

Preview into the 'raw' file with added columns. There are 58 rows, 29 samples i
n the file '20170907_SOL_46845_cur.csv'

	SampleName	Project	Formula	Mass	Found	Acquired	RT(min)	QuantA
42	XY355JVM_T	companyA	C41H50FN3O7	715.40	Yes	9/12/2017 11:28:00 AM	2.19	0.

#7 Enter Injection volume (uL) and dilution factor ↔

Inject Vol (uL): 10

Dilution factor: 50

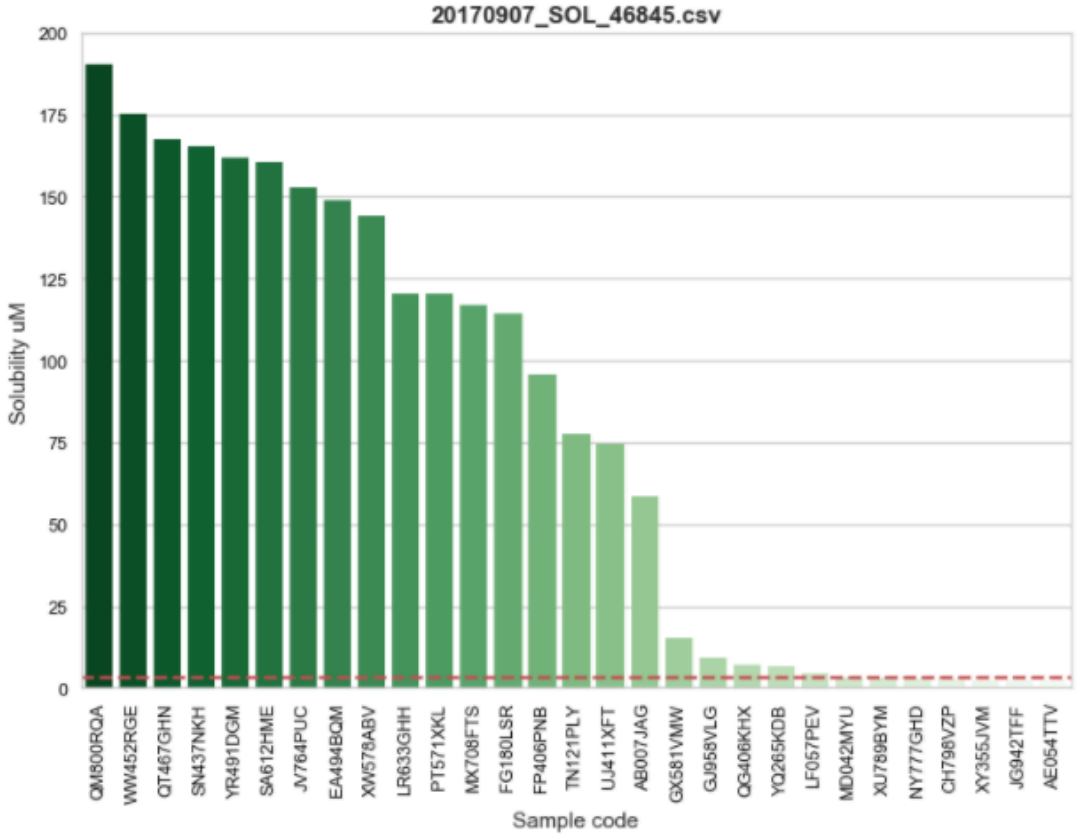
 # 8 Calculate additional values as in StockConcCalc and get output of 30 sample

View into first 30 sample values.

	SampleName	Project	Acquired	Sol220uM	Sol_ug_mL	Sol_reversal	Comments
0	AB007JAG	companyA	9/12/2017 1:41:00 PM	29.00	nan	2.56	NaN

```
1 # 9 Plot bar chart ↔
```

File `20170907_SOL_46845_report.csv` was saved in /dataout/ folder and can be s
hared with teams.



Report compounds (if any) that are more soluble in buffer then in water/ACN, aka `Solubility_reversal
`.

SampleName	Project	Sol220uM	Sol_reversal
AB007JAG	companyA	29.0	2.6
EA494BQM	companyA	129.0	1.2
FG180LSR	companyA	114.3	1.0
JV764PUC	companyA	152.9	1.0

Report poorly soluble compounds to project leader.

SampleName	Project	Sol220uM
AE054TTV	companyA	2.1
CH798VZP	companyA	2.6
JG942TFF	companyA	2.4
NY777GHD	companyA	3.0
XU789BYM	companyA	3.0
XY355JVM	companyA	2.5

Generate file for upload to Analytical LIMS

```
1 #10 Analyst ↔
```

Select: Estelle_Maes

```
1 # 11 Solvent, temperature↔
```

Temperature (oC): 25

Media: Phosphate buffer

```
1 #12 Conditions ↔
```

Select Method: Estelle

In the last several steps, we choose Analyst name, Temperature, Buffer, and LC-MS conditions from the respective pull-down menus. Upon running the cell (# 13), formatted file is created, ready for upload to the company primary data system.

To archive the processing parameters and current run results, the whole Notebook can be exported in HTML format.

```
# 13 Create LIMS file ↔
```

File was succesfully created in the ./dataout folder for period.

```
File was written to: F:\PROJECTS\   ...   \        ...        \_Jupyter\Icagen\Sol
ubility/dataout/20170907_SOL_46845_cur.tab
```

	BATCH_REF	ANALYST	DATE_MEASURED	SOLUBILITY_MICROMOLAR	SOLUBILITY_MICR
24	XU789BYM	Estelle_Maes	9/12/2017 11:57:00 AM	3.0	

Output html for archival purposes

```
1 # Create output with template. Input is removed ↔
```

Jupyter Notebook ***Solubility3_v2.ipynb****:*
version 2.1 updated on Aug 15, 2017

Conclusions

In this note, we presented and discussed the method of Kinetic solubility tests provided by Icagen as part of early ADME panel. We outlined processing of raw LC-MS data with the help of open source Jupyter Notebook and Python. Data processing for other compound properties, such as logD, PAMPA, and chemical stability is also implemented in the Jupyter environment.

References

[1] L. Di and E. H. Kerns, *Drug-like Properties: Concepts, Structure Design and Methods: from ADME to Toxicity Optimization*. Academic Press, Jul. 2010.

[2] S. Nakashima, K. Yamamoto, Y. Arai, and Y. Ikeda, "Impact of physicochemical profiling for rational approach on drug discovery," *Chemical and Pharmaceutical Bulletin*, vol. 61, no. 12, pp. 1228–1238, 2013. [Online]. Available: https://www.jstage.jst.go.jp/article/cpb/61/12/61_c13-00436/_article/-char/ja/

[3] K. T. Savjani, A. K. Gajjar, and J. K. Savjani, "Drug Solubility: Importance and Enhancement Techniques," *ISRN Pharmaceutics*, vol. 2012, Jul. 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3399483/

[4] C. Saal and A. C. Petereit, "Optimizing solubility: Kinetic versus thermodynamic solubility temptations and risks," *European Journal of Pharmaceutical Sciences*, vol. 47, no. 3, pp. 589–595, Oct. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0928098712002977

[5] M. Patek, M. Smrcina, E. Wegrzyniak, V. Nikolaev, and A. Mariscal, "Solid-Phase Combinatorial Chemistry," in *Small Molecule Medicinal Chemistry: Strategies and Technologies*, W. Czechtizky and P. Hamley, Eds. John Wiley&Sons, 2015, pp. 103–143.

[6] F. A. Nielsen, "Jupyter (formerly IPython notebook)," 2017, pp. 1–29. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/6973/pdf/imm6973.pdf