Open, Transparent, and Reproducible Data Science in Clinical Metabolomics using Jupyter Notebooks and Cloud Computing.

Metabonews Vol 9 Issue 10 Dec 2019 feature article contributed by David Broadhurst, Centre for Integrative Metabolomics & Computational Biology, Edith Cowan University, Perth, Western Australia.

Most of us are familiar with the generic lifecycle of a clinical metabolomics study. We first propose a hypothesis, then design a study, obtain samples, choose an assay, design an analytical experiment, acquire raw data, deconvolve into a data matrix, annotate metabolite peaks, perform quality control (data cleaning), extract predictive/statistical insights from data, interpret biochemically, disseminate findings, and finally generate further hypotheses or translate knowledge into practice. As figure 1 suggests, we often loop through subsets of these stages multiple times.

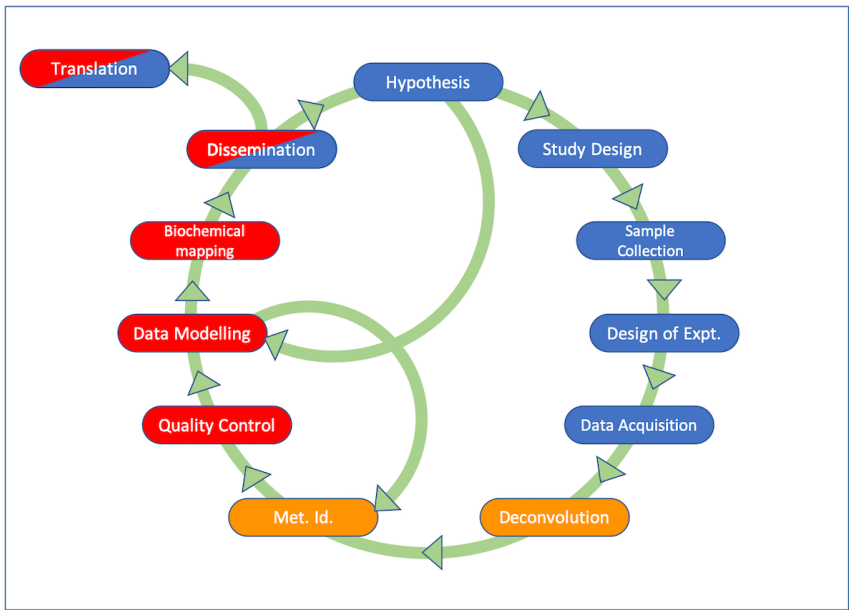

Figure 1. Lifecylce of a clincal metabolomics study

In terms of data management, each stage in the lifecycle is a discrete process (or workflow) to which data integrity checks can be applied to ensure validity and accuracy. After which data can be locked down (archived) before proceeding to the next stage. Parts of the lifecycle can be (semi)automated using off-the-shelf standard operating procedures. Typically a metabolomics lab

will have a small number of generic (biofluid specific) analytical assays each with a matching set of standard deconvolution/annotation algorithms with fixed parameter settings. Conversely, other parts of the lifecycle are bespoke to the individual study. Specifically, the design of experiments, data modelling, biochemical interpretation and dissemination.

What has this got to do with **data science**? Well, it is important to consider that the stages of a metabolomics lifecycle that require complex computational algorithms can be split into two distinct categories each with opposing intellectual and computational characteristics. Using vernacular taken from the artificial intelligence community, *peak deconvolution*, *annotation* and associated computational architecture can accurately be termed **data engineering** as the aim is to make the raw instrument data amenable to predictive modelling (e.g. disease classification) and/or biological interpretation - but no further than that. Whereas, the process of taking a curated data matrix and applying statistical, epidemiological, machine learning, visualisation techniques, together with mechanisms to translate results into biological and clinical insights can be termed **data science** (Figure 2).

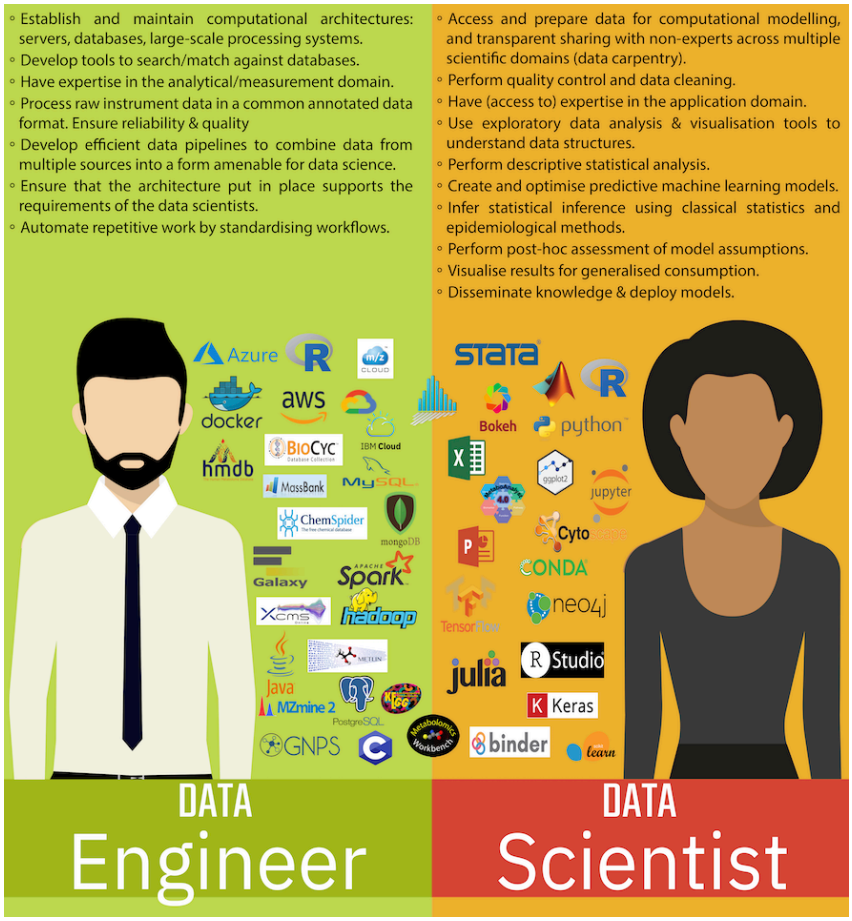

Figure 2. Data Engineer vs. Data Scientist

With this in mind, it would be reasonable to propose that any efforts to develop generalised computational frameworks for the metabolomics lifecycle should be approached independently with requirement specifications dependent on the specific needs of each job, and thus potentially completely separate computational platforms. **Peak deconvolution** and **peak annotation** are computationally intensive, often taking 10s of hours to process data, but require very little interaction or creativity. They often require access to (or development of) external web services/databases and understanding of associated computational architecture. Workflows are linear and rarely change once optimised for a given lab+instrument+assay+biofluid combination (e.g., CIMCB, LC-MS, HILIC+, Urine). As such, the underlying computational workflows take the form *"design once, document once, and run many"*. **Data modelling and visualisation**, on the other hand, is computationally cheap at execution, and can be executed in an isolated (virtual) environment. Workflows require a large amount of forethought and flexibility, with potentially multiple cascading modelling workflows, each requiring cross-validated parameter optimisation, domain expertise, exploratory investigation of data structures, post-hoc assessment of model assumptions, possibly adjusting for confounding variables or fusion with data sourced from other 'omic platforms. Each step also requires comprehensive documentation to guide reproducibility. As such, the underlying computational workflows are of the form *"design many, document many, and run once"*.

It is also important to note that an increasing number of metabolomics researchers, particularly in the clinical domain, outsource the complete *data engineering* stage of the metabolomics lifecycle to a service laboratory. Companies such as *Metabolon*, *Nightingale Health*, and *Biocrates* have business models that depend on providing high-quality fully annotated data sets in a format amenable for data science. Most large academic laboratories also provide some sort of similar service. As such, for more and more students, academics, and industry professionals, the clinical metabolomics lifecycle is driven by data science, not by analytical chemistry, cheminformatics, or data engineering. Such approaches have most recently been reported in the American Journal of Epidemiology by *You et al. (2019)* "Consortium of Metabolomics Studies (COMETS) Metabolomics in 47 Prospective Cohort Studies" .

You may or may not agree with this perspective, but one thing that most in the metabolomics community **will** agree upon is the urgent need for transparent and consistent reporting of all aspects of the metabolomics study lifecycle. The metabolomics community has made substantial efforts to align with FAIR (Findable, Accessible, Interoperable, and Reusable) data principles by developing open data formats (e.g. mzXML), data repositories (e.g. MetaboLights and Metabolomics Workbench), online spectral reference (e.g. METLIN, mzCloud, MassBank, GNPS), and online databases for metabolite identification and biochemical association (e.g.

HMDB). These data engineering resources and others like them are essential to ensure the future integrity of metabolomics as a science.

Numerous groups within the metabolomics community also actively work to standardise computational workflows and provide open source and online tools for both data engineering and data science. An excellent and comprehensive review of R packages developed for the metabolomics community has recently been published by Stanstrup el al (2019). That said, most attempts at a unified computational framework (e.g. Galaxy or KNIME) have primarily focussed on the requirements for data engineering, with data science forced to fit into a very prescriptive linear workflow. In our recent review for Metabolomics "Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing" *Mendez et al. (2019)*, we provide a brief overview of current data science programming frameworks relevant to the metabolomics community, corresponding barriers to achieving open science, and finally an introduction to one practical solution specific to data science in the form of Jupyter Notebooks. Jupyter Notebooks are an open-source, web tool for creating seamless integration of text, code (typically Python or R), and outputs (tables, figures) into a single living interactive document. This framework is particularly amenable for the needs of data science. When used alongside data repositories, such as GitHub, and open cloud-based deployment services, such as Binder, these computational notebooks can greatly enhance transparent dissemination of data science methods and results during the publication process. We provide a set of experiential learning tutorials, hosted on the Github repository, introducing the Jupyter Notebook framework, specifically tailored to the needs of a metabolomics researcher with limited programming experience.

Several researchers in the metabolomics community are already using the Jupyter/Binder framework as a means for transparent dissemination for publications (e.g.*Nett et al. 2018*; Github), or as tutorials for new data science methodologies (e.g. *AlAkwaa et al. 2018*; GitHub Binder), data engineering algorithms (e.g. *van der Hooft et al. 2016*; Github), or both (e.g.*Sands et et al. 2019*; Github Binder). Also, it would be remiss not to link to the excellent Jupyter Notebook based METASPACE Python-client.

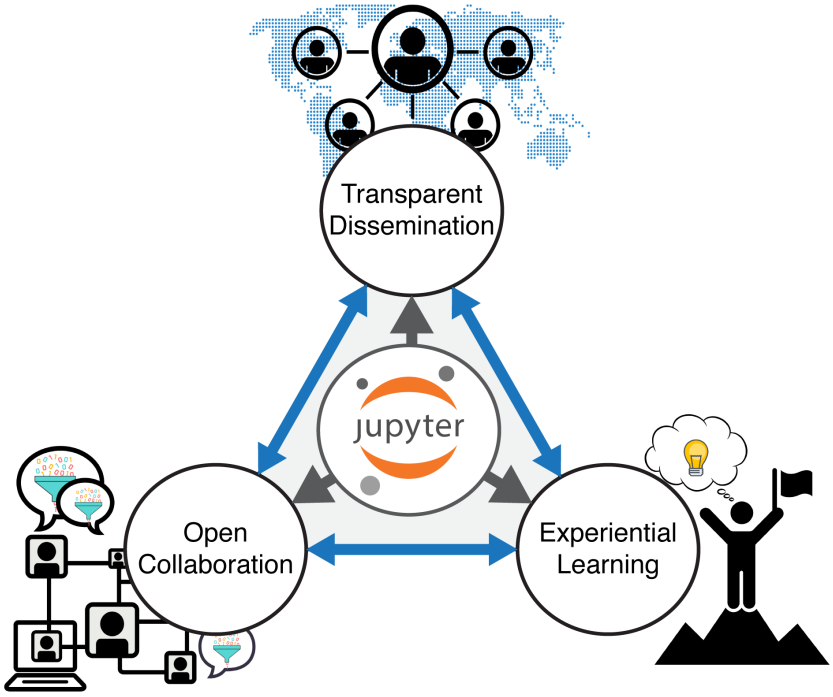

Figure 3. Taken from Mendez et al. Metabolomics (2019)

Jupyter Notebooks may not be suitable for all metabolomics applications (particularly if the primary objective is identification of novel metabolites) and for computationally intensive data engineering it is probably not as efficient as other frameworks such as local installations of Galaxy or KNIME, or dedicated commercial software such as *Compound Discover*, *Progenesis QI*, or *AnalyzerPro*, or dedicated open tools such as MS-Dial and XCMS-online. However, there is a desperate need for accessible and transparent reporting of data science methods and results. Jupyter Notebooks coupled with serverless cloud-computing, provides a surprisingly intuitive and rapid means of enabling transparent dissemination, open collaboration, and experiential learning.