Automating data processing and analysis with Mnova Gears

- White paper -

NOVA

GEARS

Mestrelab Research

chemistry software solutions

Gary Sharman
September 2021

The need for automation of analytical processing

Modern NMR and MS instruments, with their high sensitivity, rapid methods, and efficient sample changers, are extremely well adapted to acquiring large amounts of data for many samples, which has transformed what we might consider possible in terms of high throughput experimentation via these techniques. However, data acquisition is merely one part of the process that goes from sample to result. If we think in terms of total analytical pipelines, it is clear that faster data acquisition in itself may not solve a problem but could merely shift the bottleneck from data acquisition to data analysis. Thus, there has never been a greater need to have good software that can be used to automate the processing of analytical data.

Figure 1. Modern instrumentation such as the Agilent Rapidfire or the Bruker Samplejet are game changers in terms of sample throughput. However, they can shift the problem of automation to that of data analysis.

Common threads to automation

Automation of data analysis is a blanket term which covers a large and growing number of diverse processes. Users of Mnova software probably recall a number of batch analysis module and tools aimed at automating sample analysis. MGears was born out of the learning gained from these developments – although data analysis pipelines are varied, they nevertheless display many common traits that can be used to design a general solution to automation problems. With this thought in mind, let us step back and think about data analysis workflows in the most general way.

1.1. Core processes

Most processes have what we might call a ‘business end’ – the core analysis we were trying to perform. It is perhaps useful to think about breaking down such calculations to their smallest unit - an atom to use a relevant analogy - as such a unit is probably reusable in more than one process. For example, let us suppose we have a workflow whose ultimate aim is to calculate concentration by NMR. This would probably involve two fundamental steps: checking the structure is consistent by assigning the NMR, and then using those assignments to calculate a concentration. It is useful to think of the process in these terms, that is, assignment followed by concentration determination, as either component could be useful on its own in other workflows. For example, in a different process we perhaps already have assignments - as this structure has been seen many times - so only a concentration determination is required. Or perhaps we are not concerned with concentration and just want confirmation that our structure is correct. Either way, we allow for maximum reuse of components by ensuring each one performs one task, and by combining tasks together to achieve more complex ends.

1.2. Peripheral functionality

Although calculations of the kind described above are the core of any process, they are by no means the only components of a complete workflow. For example, selecting the data on which to operate or reporting results may be considered peripheral, but they are nevertheless essential to a fully integrated process. Just as for the core analysis, we can think of these as reusable components, but perhaps there are a number of ways by which one can achieve the same ends. For example, deciding which data to operate on and how that data is grouped together is part of virtually all analyses. It may be achieved in many different ways, such as by obtaining the details from a spreadsheet or by searching for files in a folder, but the overall aim is always to create a list of samples for subsequent operations. Likewise, all processes have some kind of output, be that a single result ('pass' or 'fail') or a complex document. It is also very common that a human readable, formatted report will need to be produced as part of the output, as well as any aggregated or tabulated results. We can thus think of data selection and reporting as two reusable parts of an overall process that can be configured to meet the needs of a specific problem.

1.3. Sample *versus* batch automation

Another useful distinction to make is between elements of a process that operate at the *sample* level, and those that operate at the level of the *batch*. For the most part, core functions can be thought of as operating at the sample level. 'Sample', in this context, doesn't necessarily mean only a single piece of data, but all the data that relates to the same physical sample. Reporting can operate at the sample level (detailed reports produced for each sample) or at the batch level (a summary report, such as a spreadsheet, for an entire batch). Finding and grouping data is a batch-related process as it needs to occur for all potential data for a batch before any downstream processing can itself occur.

Review is another common thread of automated analyses. Recognizing that any automated process is unlikely to be perfect, a review by exception strategy can be a very powerful tool, but one that requires a good way to present summarized results to a reviewer and that can allow them to be changed. Again, this process may differ in its details, but as a concept is very similar in many processes. Review is perhaps itself best thought of as a batch process too, as although it involves reviewing individual samples, an overview of the entire batch is necessary in order to decide which ones to review.

Figure 2 summarizes these concepts and shows how they can be considered as layers in the automation workflow.

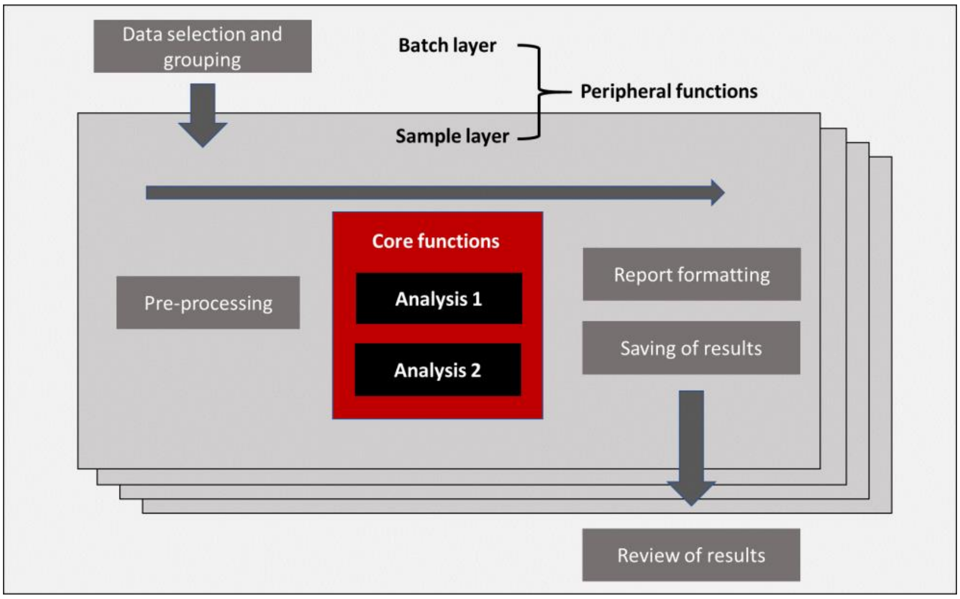

Figure 2. Any data processing workflow can be thought of conceptually as comprising core modules along with peripheral ones which together allow all aspects of the task to be automated. The modules are reusable and can be selected and configured as required, and many permutations can be made to suit the needs of a particular environment. Core and sample layer functions are repeated for each sample, but data selection, grouping, and review are applicable at the batch level rather than the sample level.

The MGears platform

The central tenet of MGears is to provide a palette of tools that offer a concrete implementation of the above concepts of analytical automation. All of this is performed under the umbrella of the familiar Mnova platform, providing a consistent interface and set of tools on which to build.

To build an automation, we select the options that best match our requirements and combine them to create a specific workflow. There are a huge number of permutations in the configurations from which you can select that are able meet the requirements of many diverse situations while sharing common features that are reusable among applications. The platform provides tools to address each of the areas discussed above – core functionality to enable complex analyses, peripheral functions to enable both complete sample analysis and a batch functionality.

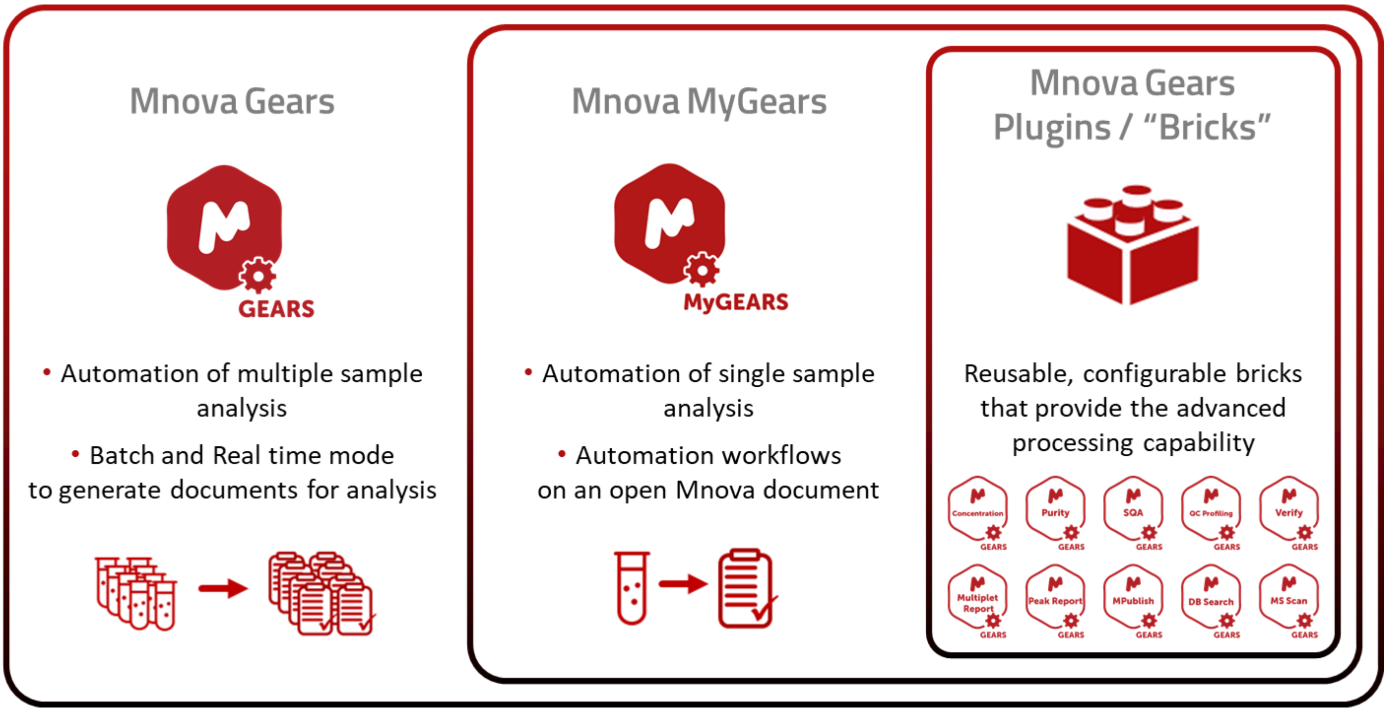
Mnova Gears
GEARS
• Automation of multiple sample analysis
• Batch and Real time mode to generate documents for analysis
Mnova MyGears
MyGEARS
• Automation of single sample analysis
• Automation workflows on an open Mnova document
Mnova Gears Plugins / "Bricks"
Reusable, configurable bricks that provide the advanced processing capability
Concentration GEARS
Purity GEARS
SQA GEARS
QC Profiling GEARS
Verify GEARS
Multiplet Report GEARS
Peak Report GEARS
MPublish GEARS
DB Search GEARS
MS Scan GEARS

Figure 3. Mnova Gears provides a concrete implementation of the general automation problem set out above. Core functionality is provided via reusable bricks. Peripheral capabilities to process, built reports, and save output for individual samples (MyGears), and expanding such sample analyses to batches (MGears) are also provided.

1.4. Bricks and plugins

The core functionality in MGears is provided by 'bricks'. These are reusable modules, each of which performs a specific task. In many cases, these wrap an advanced plugin available in Mnova, though some of them carry out specific analyses that are not available elsewhere in the software suite.

MGears bricks are actually built on the Mnova scripting engine. This provides two benefits. Firstly, they access the full power of Mnova through its API, but in a way that makes development fast. As a consequence, we see a rapidly growing and ever improving selection of bricks. Secondly, as they are based on the script engine, it is even possible for users to create their own bricks for their specific tasks, or potentially for Mestrelab to rapidly develop such solution for you.

As the bricks generally provide access to underlying Mnova functionality, configuring them is simple as in most cases the exposed parameters are already familiar from Mnova. Sensible defaults are also provided, so that in any given case either no or very little configuration is needed. However, should performance need to be more heavily customized, familiar parameters allow you to do so in the same manner as in the Mnova interface itself.

1.5. Automating sample analysis

The ability to custom process, format reports, and save output is supplied by a number of configurable modules that together turn the core analysis achieved by the bricks into a complete solution for your samples. Each aspect is managed by a simple configuration interface. Furthermore, this configuration relies on familiar Mnova concepts to achieve these goals; thus, processing can be as simple as selecting 'advised processing' or an existing Mnova processing template. Report formatting can be achieved by selecting an Mnova report template. Output can be saved to an Mnova file, an easily shred pdf, or even an Mnova spectral DB, all with a single, simple click.

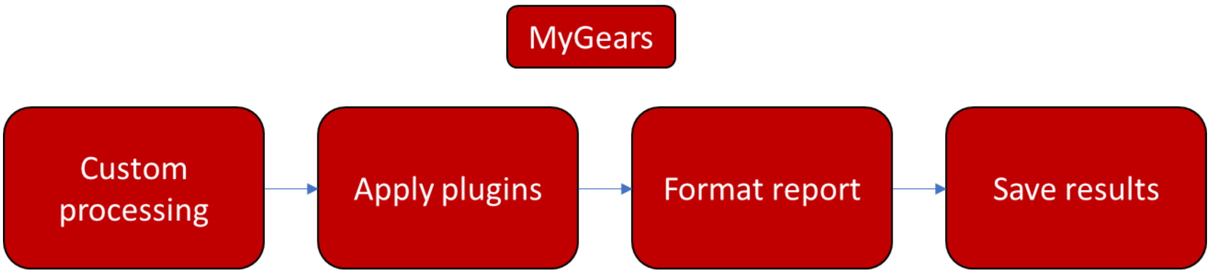

Figure 4. The steps in sample automation, which are provided by the MyGears product

It is worth stressing that Mnova Gears can operate purely in this sample-based 'MyGears' mode. The user is expected to open the relevant data in an Mnova document, and as such this represents 'sample automation' rather than full-blown batch automation. However, this is important for two reasons. First it can provide an excellent solution for laboratories with modest throughput requirements, where the burden of data preparation is relatively small compared to the sample analysis itself. Secondly, as MyGears is licensed separately due to this limitation, this may provide a significant savings to users who only need this level of automation.

1.6. Growing to batches

A batch analysis is perhaps best thought of as a superset of the sample-based analysis described above. All other steps are the same, as once data is identified the problem reduces to that described above for sample analysis – it is just that this is repeated many times. A full MGears workflow, therefore, simply has an additional set of configurations that describe how the batch is created and handled. The batch-based functionality in MGears thus builds on the sample-based analysis described above but using a powerful module that is responsible for data selection and grouping.

A number of methods are supported for data selection and grouping, from searching a file system to retrieving matching records from an Mnova database. Real-time processing of data as it becomes available is also supported. At its simplest, enabling batch functionality can mean simply selecting the type of data you want to operate on and letting MGears find it automatically. Such a simple approach may not always work, depending on how the data is structured, but a number of advanced configuration options allow more complex schemes to be quickly applied, and the fallback of a so-called master file, which simply lists the data that needs to be operated on, is also available. It is possible, therefore, to provide the necessary logic to batch up samples in a huge variety of situations. For the most difficult situations, there is also the ultimate fallback of customization via scripting, as described in the next section.

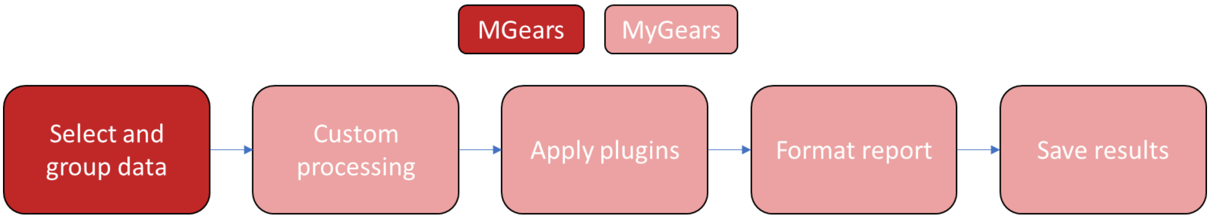

Figure 5. For full batch automation, an additional input tab allows the user to define the way data is selected and grouped for subsequent operations.

Review and batch-level reporting can also form key parts of a high throughput workflow. MGears provides two ways to achieve this. The first is a flexible review tool, which features an intuitive UI that allows the output of batch analysis to be rapidly reviewed as required. This can form part of a review by exception strategy, which can be an efficient way to combine the benefits of human intuition and know-how with the speed of automation. A second solution to review and batch reporting is the html output from MGears, which provides an interactive document, and is available even for those who do not have an Mnova license.

It should also be noted that some bricks can append each result to a combined output file such as a spreadsheet, providing another way to achieve batch-level reporting.

Customization

Although the reusable modules in MGears provide a high degree of flexibility, it nevertheless needs to be recognized that situations will arise in real processes that do not fit well with its configuration options. Faced with such a scenario, an analyst has two choices: change the process, or customize MGears. We are perhaps understandably reluctant to change our processes for many reasons, but it is important that this possibility is not dismissed out of hand as it may be the path of least resistance. However, it is recognized that in some situations, particularly where there are dependencies on other systems that are 'non-negotiable', this may mean that changing the process is not always an available option.

Thankfully, this concept was recognized in the design of MGears, and at every stage of the workflow, the user has the opportunity to supply a script to carry out some key functionality. To take one example, imagine that we are designing a workflow to verify NMR data. Perhaps your candidate structures are not conveniently available as structure files that can be matched to data by the MGears input engine. Let us further suppose that your structures are available via a REST web service – a common arrangement in some corporate environments. It would be possible to solve this by writing a small script in the Mnova script engine, making use of the built-in API to access a web service, and bring the structure back to the MGears workflow. It would be hard to add a sufficiently general configuration option to deal with situation like this, but by providing access to custom scripts, workflows can be adapted to meet just such diverse requirements.

Of course, a good rule of thumb is to use the available functionality in MGears as far as possible, but having this option is nevertheless an excellent 'get out of jail' card.

Conclusions

Automation of sample data analysis is a vital part of building streamlined end-to-end processes starting with physical samples and ending with results in a requestor's hands. Many of the activities that are needed to do this are generic and reusable, and MGears builds on this idea to provide a flexible platform that can be adapted to many situations. Whether you have modest sample automation needs and require MyGears, or whether full batch capability is required for your workflow, Mnova Gears provides a flexible and general platform, designed from the ground up as a reusable automation tool.

You can read more about Mnova Gears and its applications here.

Contact us:
info@mestrelab.com for general queries
sales@mestrelab.com for sales queries

We look forward hearing about your own automations!