NMRdock: Lightweight and Modular NMR Processing

Kyle W. East,[1] Andrew Leith,[2] Ashok Ragavendran,[2] Frank Delaglio[3] and George P. Lisi*[1]

[1]Department of Molecular Biology, Cell Biology & Biochemistry, Brown University, Providence, RI, USA

[2]Computational Biology Core, COBRE Center for Computational Biology of Human Disease, Brown University, Providence, RI, USA

[3]Institute of Bioscience and Biotechnology Research, National Institute of Standards and Technology and the University of Maryland, Rockville, MD, USA

KEYWORDS: NMR, container, Docker

ABSTRACT: NMR is a widely employed tool in chemistry, biology, and physics for the study of molecular structure and dynamics. Advances in computation have produced scores of software programs necessary for the processing and analysis of NMR data. However, the production of NMR software has been largely overseen by academic labs, each with their own preferred OS, environment, and dependencies. This lack of broader standardization and the complexity of installing and maintaining NMR-related software creates a barrier of entry into the field. To further complicate matters, as computation evolves, many aging software packages become deprecated. To reduce the barrier for newcomers and to prevent deprecation of aging software, we have created the NMRdock container. NMRdock utilizes containerization to package NMR processing and analysis programs into a single, easy-to-install Docker image that can be run on any modern OS. The current image contains two bedrock NMR data processing programs (NMRPipe and NMRFAM Sparky). However, future development of NMRdock aims to add modules for additional analysis programs to build a library of tools in a standardized and easy-to-implement manner. NMRdock is open source and free to download at https://compbiocore.github.io/nmrdock/.

Nuclear magnetic resonance (NMR) has been one of the most valuable tools at the interface of chemistry, biology, and physics for the past 60 years. NMR can probe both the structure and dynamics of a target molecule with atomic resolution and has become ubiquitous in structural biology,[1-4] synthetic[5, 6] and medicinal chemistry,[7-9] and materials sciences.[10, 11] This utility is facilitated by a rapid improvement in technology (*i.e.* cryoprobes,[12, 13] high field magnets,[14] MAS/high-speed rotors[15, 16]) and pulse sequence design (*i.e.* TROSY,[17, 18] CRINEPT,[19, 20] NUS,[21] Ultrafast[22]) that has created avenues for studies of increasingly large and complex systems. An equally important advance in the NMR field has been the improvement in computational power that allows for faster and more consistent processing and analysis of NMR data.[23]

Spectroscopists have thrived in this environment, designing and implementing scores of software packages catalogued by the BioMagResBank.[24] Many of these programs make up the bedrock of modern NMR spectroscopy,[25, 26] allowing for the processing of Fourier and nonuniformly sampled data,[27] robustly fitting NMR spin relaxation parameters to extract information on chemical dynamics,[28-30] and adapting NMR distance constraints to elucidate biomolecular structures.[31] However, as the demand for NMR spectroscopy in biological settings grows, three significant issues have arisen: 1) the 100+ unique software packages reported in the literature have their own system-dependent environments, paths, and dependencies, complicating installation; 2) software development and utilization has been a laboratory-dependent process, with many research groups supporting different operating systems and relying on different software packages and 3) planned updates to modern operating systems are removing or altering libraries that are required by many older packages, forcing the deprecation of these programs. All three of these computational problems place large barriers for entry into the field.

Several projects attempting to reduce this barrier are underway, including those focused on automated data processing programs with helpful GUIs and powerful algorithms (*i.e.* MestreNova, CRAFT),[32] transitions to system-independent languages (*i.e.* Python-based software),[28] or the aggregation of software packages into a single source (*i.e.* NMRbox).[33] Each of these avenues has made NMR more accessible to the greater scientific community, but there are also disadvantages to each solution. For example, advanced GUIs are not always open-source and require expensive licenses, while system-independent languages have reduced installation complexity of NMR software but still require additional

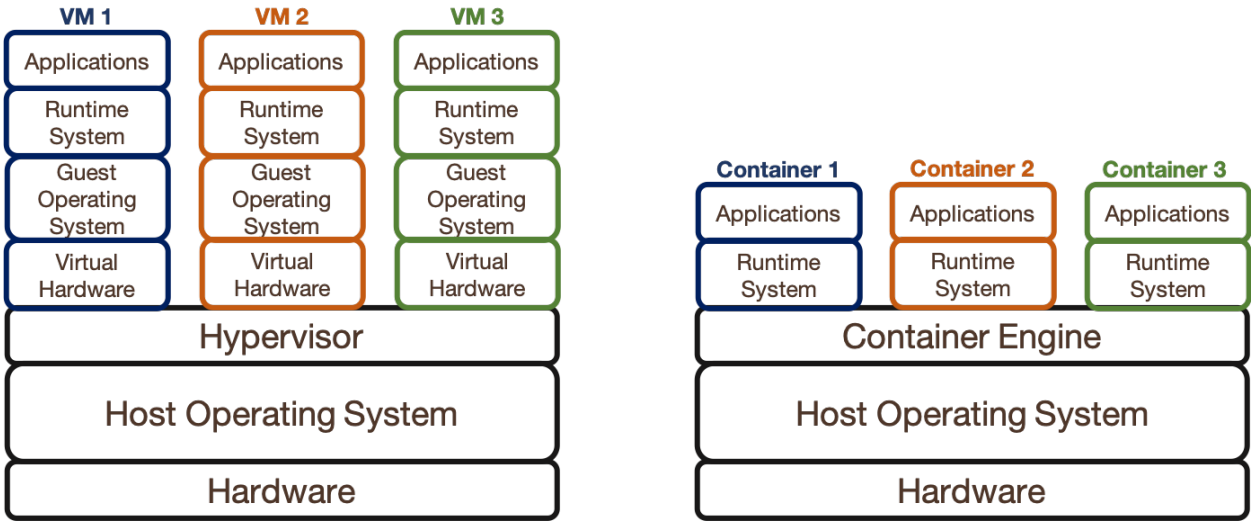

Figure 1. Virtualization vs. Containerization. Virtual Machines (left) require a Hypervisor that creates one or more virtual machines. Each virtual machine comprises virtual hardware that hosts a guest operating system containing the necessary runtime libraries and applications. Containers (right) require a container engine that can carry one or more containers and interfaces these containers with the host OS and hardware. Each container comprises the necessary runtime libraries and the applications.

integration and knowledge of proper dependencies. At the moment, a large suite of NMR software is available in NMRBox,[33] a highly versatile and powerful virtual machine (VM) that integrates available software into a single unit. We sought to build upon this approach by creating a lightweight, standardized and modular containerized system in order to 1) make it easy to install and use for new spectroscopists, 2) create a long-term solution for deprecating software, and 3) make it an accessible educational tool for NMR data processing and analysis in classroom and workshop settings. In this work, we highlight the potential of containerization for implementing NMR software with no dependence on the configuration of the host.

Why Containerization?

Our goal was to create a lightweight, easy-to-use, and adaptable tool for maintaining NMR software that can 1) run on any modern operating system (OSX, Windows, and Linux), 2) support 32-bit and deprecating software packages, and 3) maintain a modular structure to allow for future development. To meet all of these requirements, we turned to containerization.

The current standard in the field relies on virtual machines that abstract the hardware and run a guest operating system on this virtual hardware. This provides unparalleled versatility and security, but comes at the cost of massive computational overhead required to create the virtual hardware, a burden that is eased with the lightweight containerization paradigm. Unlike virtualization, containers package applications into an optimized environment, holding the program and necessary runtime libraries, but utilize the host operating system through the container engine (Figure 2). This structure is more lightweight and adaptable than a VM, and is becoming more widely employed by server-based companies like Google and Amazon.[34]

Several container engines exist, including Rkt, Singularity, and Docker. Unlike with the various implementations of virtual machines, each of these engines has been designed for different purposes and often supports only a single operating system: specifically, Rkt and Singularity both run exclusively on Linux. Fortunately, while Docker runs natively on Linux alone, it has been adapted to run on both Microsoft Windows and Mac OSX, with installers made easily available to end-users (https://docs.docker.com/). Therefore, we utilized the Docker Engine as the base for NMRdock.

Building NMRdock

The current NMRdock image includes NMRPipe[25] and NMR-FAM Sparky[35] for processing and viewing multidimensional NMR data. All necessary parties have consented to use. NMRPipe is available witout charge from NIST and NIH, while NMR-FAM Sparky is accessible under the GNU General Public License. Within the NMRdock image, NMRPipe and Sparky are both stored in the home directory (/home/ubuntu/). The home directory also includes a Data directory (/home/ubuntu/data) for use as a volume for the container to access data on the host computer. When the NMRdock image is run, the directory that contains the data of interest should be mounted to the image as the

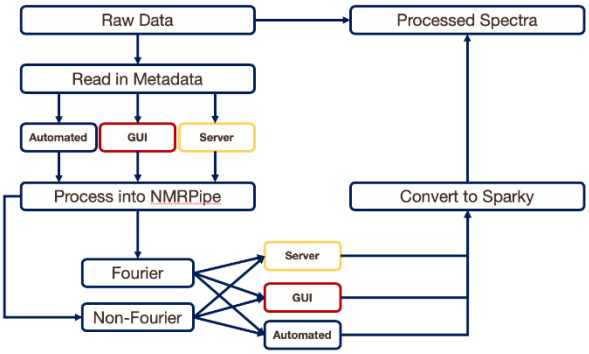

Figure 2. Workflow of NMRdock. The standard workflow of NMRdock allows for the simple processing of raw NMR data through NMRPipe and into Sparky. NMRdock has all of the functionality of both NMRPipe and NMRFAM Sparky and can process both Fourier and NUS data. Spectra can be further analyzed in NMRFAM Sparky.

data directory in order to allow access. NMRdock can be run in either C-shell or Bash; however, we have optimized the container to run in Bash. To accommodate both shells, we have included a series of wrappers that allow the C-shell (necessary for NMRPipe) to be accessed within Bash. The build files of the current NMRdock image are accessible on GitHub (https://github.com/compbiocore/nmrdock/). We encourage the opening of issues on GitHub for any bugs or requested features.

Utilizing NMRdock

The general workflow for NMRdock (Figure 2) uses NMRPipe and NMRFAM Sparky to process raw NMR data and to import the processed data into Sparky for analysis. Additional features are being built to expand the base system. NMRdock requires two dependencies: the Docker Engine and an X Window System. The OS-specific Docker Engine (https://docs.docker.com/) must be installed in order to interface the container image with the host operating system. Both NMRPipe and Sparky rely on X Windows for displaying data, and in order to use the GUIs, an XServer is required for your operating system. Furthermore, Mac and Windows operating systems both require a tool for TCP port forwarding to pass data between the container and the host; any such utility can be used for that purpose. On MacOSX, the Socat port forwarder can be installed using Homebrew or MacPorts, and XQuartz is the most widely used implementation of XServer for OSX. We suggest VcXsrv or Xming on Windows, while most distributions of Linux natively include a port forwarder and an X Window system.

There is a quick installation guide at https://compbiocore.github.io/nmrdock/ that will aid in installing Docker and the XServer for your operating system. Detailed instructions for installation of NMRdock are also available. Once NMRdock is installed, it can be accessed from a terminal window or through the executable. It should be noted that changes made within the NMRdock image are not persistent, so all changes (*i.e.* processed and analyzed data) should be stored on the host computer before exiting NMRdock.

Challenges

One of the major challenges of any software package is longevity. As modern computation evolves, libraries necessary for running older software are either updated or removed. An example that may impact the NMR community is the planned removal of 32-bit support from Mac OSX in Q4 2019 and from Microsoft Windows in 2019 or 2020. NMR software designed and compiled for 32-bit processing will likely experience issues with this transition as computer architectures move exclusively to 64-bit processing. For example, NMRDraw, contained within the NMRPipe package, will no longer be supported natively on Mac OSX. Although we are uncertain of the future changes being made to Mac OSX and Microsoft Windows, the NMRdock container is expected to allow continued use of the 32-bit NMRDraw until it can be replaced by a 64-bit application. The Docker-based NMRdock is stable and continuously supported, where any updates to the base Docker engine will automatically trigger a rebuild of the NMRdock image. NMRdock is also maintained on GitHub, so issues that arise and features to be considered for future builds can be opened as issues directly within GitHub for the developers.

Educational Tools and Ongoing Development

In addition to its utility for NMR data processing and analysis by intermediate-to-end users through GUIs or the command line, we aim to make NMRdock a widely accessible educational tool for bio-NMR data analysis, where little-to-no up-front computational and coding expertise is required to generate useful results. The NMRdock module may be of particular interest to

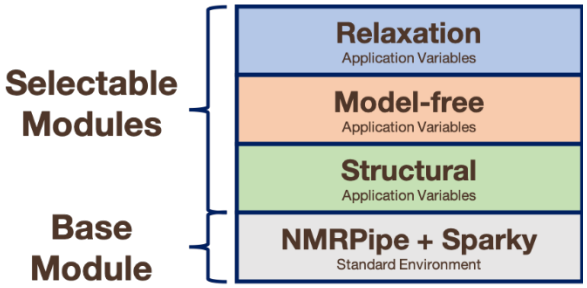

Figure 3. Modularity of NMRdock Containers. NMRdock can be expanded with additional modules. The base NMRdock module presented here can then interface with future modules to aid in the analysis of relaxation, model-free, and structural data.

laboratories that are "light" or "occasional" users of NMR that may not be equipped to install software written in multiple languages or seek out custom processing scripts.

We envision two directions for ongoing work on this module. First, we will work with other developers to increase the library of containers for current and future bio-NMR software. NMRdock is designed to be a modular system, with the base platform described here being supplemented by continued additions of other NMR software packages (Figure 3). This allows the user to easily install and control the necessary NMR packages for their workflow (Figure 4). Second, we will develop and implement automated pipelines for NMR data processing, requiring only the input of raw data files and execution of a single command to generate a complete NMR spectrum. This will allow the creation of server-based NMR data processing suites.

Conclusions

In this work, we have described a containerization method for the development and maintenance of bio-NMR data processing and analysis software. Using the Docker Engine framework, we have designed NMRdock, containing NMRPipe[25] and NMRFAM Sparky,[35] for the processing of raw NMR data into spectra and the analysis of those spectra. By introducing this tool, we aim to make NMR more accessible for new users, create a long-term solution for aging software, and design an accessible educational tool.

Corresponding Author

*George P. Lisi (george_lisi@brown.edu)

Acknowledgments

This work was supported by start-up funds from Brown University and funds from the COBRE Center for Computational Biology of Human Disease (P20GM109035) to GPL.

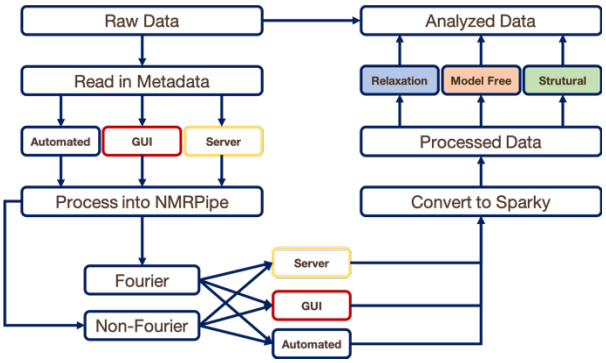

Figure 4. Modular Workflow of NMRdock. Advanced workflows of NMRdock include the current build in addition to future modules that can analyze processed data extracting relaxation, model-free, and structural parameters.

References

[1] Markwick, P. R., Malliavin, T., and Nilges, M. (2008) Structural biology by NMR: structure, dynamics, and interactions, *PLoS Comput Biol 4*, e1000168.

[2] Marion, D. (2013) An introduction to biological NMR spectroscopy, *Mol Cell Proteomics 12*, 3006-3025.

[3] van der Wel, P. C. A. (2018) New applications of solid-state NMR in structural biology, *Emerg Top Life Sci 2*, 57-67.

[4] Sugiki, T., Kobayashi, N., and Fujiwara, T. (2017) Modern Technologies of Solution Nuclear Magnetic Resonance Spectroscopy for Three-dimensional Structure Determination of Proteins Open Avenues for Life Scientists, *Comput Struct Biotechnol J 15*, 328-339.

[5] Rizzo, V., and Pinciroli, V. (2005) Quantitative NMR in synthetic and combinatorial chemistry, *J Pharm Biomed Anal 38*, 851-857.

[6] Gomez, M. V., and de la Hoz, A. (2017) NMR reaction monitoring in flow synthesis, *Beilstein J Org Chem 13*, 285-300.

[7] Wishart, D. (2005) NMR spectroscopy and protein structure determination: applications to drug discovery and development, *Curr Pharm Biotechnol 6*, 105-120.

[8] Singh, S., and Roy, R. (2016) The application of absolute quantitative (1)H NMR spectroscopy in drug discovery and development, *Expert Opin Drug Discov 11*, 695-706.

[9] Pandya, A., Howard, M. J., Zloh, M., and Dalby, P. A. (2018) An Evaluation of the Potential of NMR Spectroscopy and Computational Modelling Methods to Inform Biopharmaceutical Formulations, *Pharmaceutics 10*.

[10] Blumich, B., and Singh, K. (2018) Desktop NMR and Its Applications From Materials Science To Organic Chemistry, *Angew Chem Int Ed Engl 57*, 6996-7010.

[11] Ashbrook, S. E., Griffin, J. M., and Johnston, K. E. (2018) Recent Advances in Solid-State Nuclear Magnetic Resonance Spectroscopy, *Annu Rev Anal Chem (Palo Alto Calif) 11*, 485-508.

[12] Russell, D. J., Hadden, C. E., Martin, G. E., Gibson, A. A., Zens, A. P., and Carolan, J. L. (2000) A comparison of inverse-detected heteronuclear NMR performance: conventional vs cryogenic microprobe performance, *J Nat Prod 63*, 1047-1049.

[13] Kovacs, H., Moskau, D., and Spraul, M. (2005) Cryogenically cooled probes - a leap in NMR technology, *Prog Nucl Mag Res Sp 46*, 131-155.

[14] Quinn, C. M., Wang, M., and Polenova, T. (2018) NMR of Macromolecular Assemblies and Machines at 1 GHz and Beyond: New Transformative Opportunities for Molecular Structural Biology, *Methods Mol Biol 1688*, 1-35.

[15] Xue, K., Sarkar, R., Motz, C., Asami, S., Camargo, D. C. R., Decker, V., Wegner, S., Tosner, Z., and Reif, B. (2017) Limits of Resolution and Sensitivity of Proton Detected MAS Solid-State NMR Experiments at 111 kHz in Deuterated and Protonated Proteins, *Sci Rep 7*, 7444.

[16] Polenova, T., Gupta, R., and Goldbourt, A. (2015) Magic angle spinning NMR spectroscopy: a versatile technique for structural and dynamic analysis of solid-phase systems, *Anal Chem 87*, 5458-5469.

[17] Pervushin, K., Riek, R., Wider, G., and Wuthrich, K. (1997) Attenuated T2 relaxation by mutual cancellation of dipole-dipole coupling and chemical shift anisotropy indicates an avenue to NMR structures of very large biological macromolecules in solution, *Proc Natl Acad Sci U S A 94*, 12366-12371.

[18] Xu, Y., and Matthews, S. (2013) TROSY NMR spectroscopy of large soluble proteins, *Top Curr Chem 335*, 97-119.

[19] Riek, R., Pervushin, K., and Wuthrich, K. (2000) TROSY and CRINEPT: NMR with large molecular and supramolecular structures in solution, *Trends Biochem Sci 25*, 462-468.

[20] Riek, R., Wider, G., Pervushin, K., and Wuthrich, K. (1999) Polarization transfer by cross-correlated relaxation in solution NMR with very large molecules, *P Natl Acad Sci USA 96*, 4918-4923.

[21] Delaglio, F., Walker, G. S., Farley, K. A., Sharma, R., Hoch, J. C., Arbogast, L. W., Brinson, R. G., and Marino, J. P. (2017) Non-Uniform Sampling for All: More NMR Spectral Quality, Less Measurement Time, *Am Pharm Rev 20*.

[22] Giraudeau, P., and Frydman, L. (2014) Ultrafast 2D NMR: an emerging tool in analytical spectroscopy, *Annu Rev Anal Chem (Palo Alto Calif) 7*, 129-161.

[23] Gao, X. (2013) Recent advances in computational methods for nuclear magnetic resonance data processing, *Genomics Proteomics Bioinformatics 11*, 29-33.

[24] Ulrich, E. L., Akutsu, H., Doreleijers, J. F., Harano, Y., Ioannidis, Y. E., Lin, J., Livny, M., Mading, S., Maziuk, D., Miller, Z., Nakatani, E., Schulte, C. F., Tolmie, D. E., Kent Wenger, R., Yao, H., and Markley, J. L. (2008) BioMagResBank, *Nucleic Acids Res 36*, D402-408.

[25] Delaglio, F., Grzesiek, S., Vuister, G. W., Zhu, G., Pfeifer, J., and Bax, A. (1995) NMRPipe: a multidimensional spectral processing system based on UNIX pipes, *J Biomol NMR 6*, 277-293.

[26] D.G, G. T. D. K. (2008) SPARKY 3., University of California, San Francisco, San Francisco, CA.

[27] Mobli, M., and Hoch, J. C. (2014) Nonuniform sampling and non-Fourier signal processing methods in multidimensional NMR, *Prog Nucl Magn Reson Spectrosc 83*, 21-41.

[28] Morin, S., Linnet, T. E., Lescanne, M., Schanda, P., Thompson, G. S., Tollinger, M., Teilum, K., Gagne, S., Marion, D., Griesinger, C., Blackledge, M., and d'Auvergne, E. J. (2014) relax: the analysis of biomolecular kinetics and thermodynamics using NMR relaxation dispersion data, *Bioinformatics 30*, 2219-2220.

[29] Bieri, M., and Gooley, P. R. (2011) Automated NMR relaxation dispersion data analysis using NESSY, *Bmc Bioinformatics 12*.

[30] Kleckner, I. R., and Foster, M. P. (2012) GUARDD: user-friendly MATLAB software for rigorous analysis of CPMG RD NMR data, *J Biomol NMR 52*, 11-22.

[31] Shen, Y., Lange, O., Delaglio, F., Rossi, P., Aramini, J. M., Liu, G., Eletsky, A., Wu, Y., Singarapu, K. K., Lemak, A., Ignatchenko, A., Arrowsmith, C. H., Szyperski, T., Montelione, G. T., Baker, D., and Bax, A. (2008) Consistent blind protein structure generation from NMR chemical shift data, *Proc Natl Acad Sci U S A 105*, 4685-4690.

[32] Krishnamurthy, K. (2013) CRAFT (complete reduction to amplitude frequency table)--robust and time-efficient Bayesian approach for quantitative mixture analysis by NMR, *Magn Reson Chem 51*, 821-829.

[33] Maciejewski, M. W., Schuyler, A. D., Gryk, M. R., Moraru, II, Romero, P. R., Ulrich, E. L., Eghbalnia, H. R., Livny, M., Delaglio, F., and Hoch, J. C. (2017) NMRbox: A Resource for Biomolecular NMR Computation, *Biophys J 112*, 1529-1534.

[34] Sharma, P. C., L.; Shenoy, P.; Tay, Y.C. (2016) Containers and Virtual Machines at Scale: A Comparative Study, In *Proceedings of the 17th International Middleware Conference*, Trento, Italy.

[35] Lee, W., Tonelli, M., and Markley, J. L. (2015) NMRFAM-SPARKY: enhanced software for biomolecular NMR spectroscopy, *Bioinformatics 31*, 1325-1327.