Markus List[1]

Using Docker Compose for the Simple Deployment of an Integrated Drug Target Screening Platform

[1] Computational Biology and Applied Algorithmics, Max Planck Institute for Informatics, Saarland Informatics Campus, 66123 Saarbrücken, Germany, E-mail: markus.list@mpi-inf.mpg.de

Abstract:
Docker virtualization allows for software tools to be executed in an isolated and controlled environment referred to as a container. In Docker containers, dependencies are provided exactly as intended by the developer and, consequently, they simplify the distribution of scientific software and foster reproducible research. The Docker paradigm is that each container encapsulates one particular software tool. However, to analyze complex biomedical data sets, it is often necessary to combine several software tools into elaborate workflows. To address this challenge, several Docker containers need to be instantiated and properly integrated, which complicates the software deployment process unnecessarily. Here, we demonstrate how an extension to Docker, Docker compose, can be used to mitigate these problems by providing a unified setup routine that deploys several tools in an integrated fashion. We demonstrate the power of this approach by example of a Docker compose setup for a drug target screening platform consisting of five integrated web applications and shared infrastructure, deployable in just two lines of codes.

Keywords: docker, virtualization, scalability, deployment, high-throughput screening

DOI: 10.1515/jib-2017-0016

Received: March 21, 2017; **Accepted:** April 18, 2017

1 Introduction

Dealing with massive amounts of biological data is unthinkable without state-of-the-art tools. Over time, these applications have become increasingly complex and can often only be used when a long list of preconditions are met. There are serious issues with the installation and maintenance of tools due to version conflicts, outdated repositories and poor documentation. Consequently, few tools targeted at the biomedical community are used outside of the institutions that initiated their development.

Recently, the virtualization software Docker (https://docker.com) has gained much attention as a potential solution to this widespread issue. Docker containers allow for applications to be instantiated based on predefined Docker images which, in turn, are generated by instructions stored in Dockerfiles. Ready-to-use Docker images for various applications are available from public repositories such as DockerHub (https://hub.docker.com).

While both Docker containers and virtual machines enable their users to execute applications in an isolated and well controlled environment, Docker containers are much more lightweight than virtual machines. This is achieved by sharing functionality of the host system's kernel, dramatically reducing the overhead otherwise introduced by the operating system needed to support a virtual machine. Note that although Docker requires a linux host system, it is possible to deploy Docker containers on MacOS and Windows, on which a stripped down linux-based virtual machine is automatically configured as host system.

The bioinformatics community has recently begun to embrace Docker and its potential for distribution of scientific software tools [1], [2], [3], [4]. Realizing that researchers face challenges in locating and using scientific software distributed through public registries such as DockerHub motivated initiatives such as Biocontainers (https://biocontainers.pro) and BioShaDock [5] to offer community-driven, curated alternatives.

A typical concern with Docker is that the virtualization of software might lead to an intolerable decrease in performance. However, Di Tommaso et al. [6] conclude that the performance loss for long-running computational tasks such as those found in genomics data analysis pipelines is negligible.

Markus List is the corresponding author.
©2017, Markus List, published by De Gruyter.
This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 3.0 License.

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd

Another issue concerns user access, which is only feasible through establishing a network connection to the container. While this issue does not concern web applications, the majority of bioinformatics tools are accessible through graphical user- or command line interface. Therefore, projects such as GUIdock [7], which enables the use of graphical user interfaces through an emulated X11 server, or AlgoRun [8], which provides a unified interface to equip packaged algorithms with a browser interface, were established.

A third issue is inherent to the paradigm of Docker, which dictates that each container represents a single application. However, solving complex tasks in computational biology typically requires integrating several tools into a common workflow. Open platforms like Galaxy [9] and Taverna [10] have emerged to simplify building and operating such workflows. Nevertheless, ensuring the fulfillment of all preconditions remains a critical issue with these systems. To use Docker even in such complex scenarios, multiple containers need to be linked and operated together, leading back to a complicated setup procedure that may be difficult to reproduce.

A promising solution to this issue is an extension called Docker compose (https://docs.docker.com/compose/), which was specifically designed to facilitate interaction of several Docker containers in a coherent software configuration. Here, we demonstrate the power of this approach by creating and deploying an integrated drug discovery platform through Docker compose.

In high-throughput functional genomics studies, sophisticated genome manipulation techniques are combined with the power of a robotic high-throughput screening (HTS) platform [11]. This allows for large-scale drug target discovery experiments, aimed at, for instance, identifying disease-specific vulnerabilities systematically [12]. HTS experiments typically yield fluorescence readouts of metabolic activity or cell viability. To increase the information gain of these costly experiments, it is desirable to obtain further readouts, which can be achieved, for instance, by depositing cell lysates on reverse-phase protein arrays (RPPAs) for protein expression quantification [13].

The HTS and RPPA readouts provide a wealth of data that can be used to identify drug targets systematically. To conduct experiments on this scale successfully, a robust and powerful bioinformatics infrastructure is necessary. Computational challenges have to be addressed on several levels, reaching from sample logistics, over management of experimental meta- and readout data, as well as data processing, normalization and analysis down to the level of systems biology analysis and hypothesis generation.

It is unrealistic to solve all of these tasks in a monolithic piece of software. Thus several tools addressing the individual tasks need to be tightly integrated to enable researchers to efficiently track large scale experiments across multiple high-throughput technologies while sample complexity increases in each step.

At the NanoCAN Center of Excellence in Nanomedicine at the University of Southern Denmark, several web application tools have been integrated into a comprehensive platform that facilitate extensive support for the design and execution of drug target discovery experiments as described above (Figure 1). The platform comprises tools for laboratory information management [14], HTS sample and plate management [15], HTS data analysis [16], systems biology analysis [17], and reverse-phase-protein array data management/analysis [18]. In spite of best efforts for the documentation of each of these tools, their integrated setup is complex and time-consuming, which makes it challenging to establish the same platform at a different screening center. This is an ideal use case to demonstrate how Docker compose can be used to simplify a complex setup routine involving several tools with different software requirements.

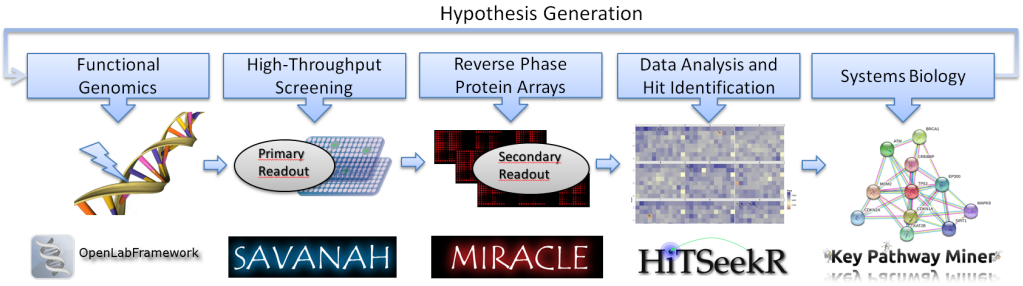

Figure 1: The web tools integrated here via Docker compose constitute a drug target screening platform. (i) OpenLabFramework is used to track functional genomics experiments that yield modified cell lines. (ii) These cell lines are then typically seeded across dozens of microtiter plates in a high-throughput screening experiment logistically tracked by SAVANAH. Here, each well constitutes its own experiment, e.g. a single gene knockout. The outcome is often a fluorescence based value of metabolic activity. (iii) Cells left-over from high-throughput screens can be lysed and deposited on reverse-phase protein arrays. MIRACLE tracks samples throughout this process and allows for the analysis of protein readout data. (iv) Primary and secondary screening results are processed and normalized in HiTSeekR to facilitate identification of samples that exhibit a phenotype of interest. (v) Genes of interest are finally subjected to systems biology analysis, leading to the generation of new hypotheses.

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd

2 Architecture

Before setting up the drug target discovery platform, we constructed individual Docker images for each of the software tools and added them to DockerHub to make them easily accessible (Table 1). Furthermore, to guarantee that these images remain up-to-date, we configured an automated build option that is triggered when a new commit is made to the corresponding github source code repository. This can be achieved by adding a Dockerfile with the necessary build instruction to the root of the github repository (https://docs.docker.com/docker-hub/github/). Next, we created another github repository for the drug target discovery platform, including the necessary config files for each of the tools as well as a docker-compose.yml file with the joint build instructions (https://github.com/NanoCAN/Docker-HTS-platform). Figure 2 illustrates how the instantiated Docker containers communicate within the isolated virtual network and how additional service containers provide relational database management (MySQL, https://hub.docker.com/_/mysql/), a key-value store (Redis, https://hub.docker.com/_/redis/), and a search engine (ElasticSearch, https://hub.docker.com/_/elasticsearch/). All user access is browser-based, with requests channeled through a dedicated web server container that serves as entry point to the platform. With the exception of service containers but including the scientific software tools, the source code used here is licensed under GPLv3 and can be freely distributed and modified.

Table 1: DockerHub URLs for the web applications used in the drug discovery platform.

Software	Docker image location
OpenLabFramework	https://hub.docker.com/r/nanocan/openlabframework/
SAVANAH	https://hub.docker.com/r/nanocan/savanah/
MIRACLE	https://hub.docker.com/r/nanocan/miracle/ https://hub.docker.com/r/nanocan/rmiracle/
HiTSeekR	https://hub.docker.com/r/nanocan/hitseekr/
KeyPathwayMiner	https://hub.docker.com/r/baumbachlab/keypathwayminer-web/

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd

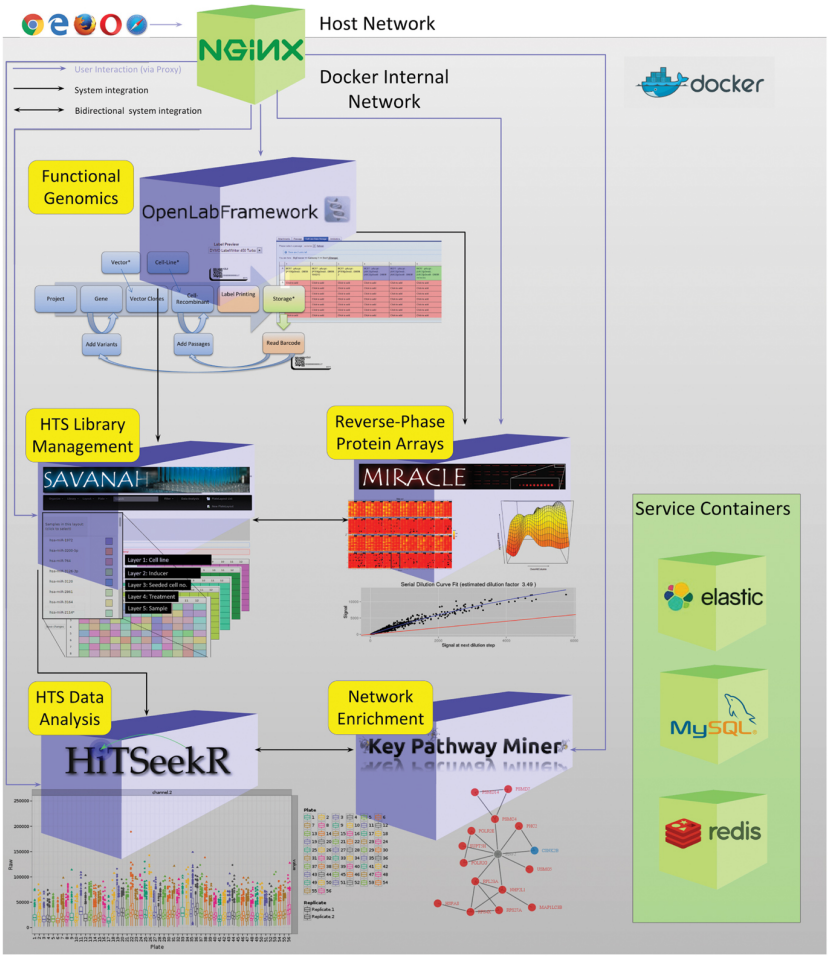

Figure 2: Overview of the high-throughput screening platform with individual Docker containers for each application (purple boxes). Pre-defined network connections facilitate protected internal communication among the tools where needed (black arrows). Service containers (green boxes) are shared by all applications (black arrows omitted here for clarity). An NGINX Docker container serves as the single entry point to the platform and enables user access to individual tools through the web browser (purple arrows).

2.1 Requirements and Setup

The drug target discovery platform presented here requires a system with at least 8GB of memory and a linux kernel (minimum v. 3.1). We recommend installation of Docker (minimum v.1.9.1) and git through the system's package manager, since all popular linux distributions provide binary installations that are preconfigured for the respective operating system. The installation of Docker compose (min v. 1.8) is facilitated via command line as described in the online documentation (https://github.com/docker/compose/releases). On Windows or MacOS, Docker and Docker compose should be installed through the corresponding setup routines (https://-docs.docker.com/). Once Docker and Docker compose are installed, the drug target discovery pipeline can be set up with the following commands:

```
git clone https://github.com/NanoCAN/Docker-HTS-platform.git
cd Docker-HTS-platform && docker-compose up -d
```

The start page of the platform is now accessible in any web browser at http://localhost/.

3 Application

Drug target screening is an elaborate process that often begins on the level of functional genomics, where advanced molecular techniques allow for targeted manipulation of the genome. With this, fine-grained studies of wild type genes and/or their mutated variants are feasible on a large scale. In high-throughput functional

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd

genomics, an entire panel of genetically modified cell lines can be created, serving as the basis for testing the effect of genetic perturbations on the genome level.

In particular robotic HTS technology is suited for testing the effects of, for instance, gene or miRNA knockouts, small compounds or dosage effects of known drugs for thousands of experiments in parallel. However, the information gained through HTS experiments is often quite limited, since the readout is typically a single measurement of a specific metabolic activity or cellular viability.

Here, reverse-phase array technology, where cells are lysed post-screening for quantifying protein expression, can serve as an appropriate strategy for increasing the information yield manifold. The combination of functional genomics, HTS and RPPA thus constitute an effective platform for drug target screening.

Dedicated bioinformatics tools are essential to support sample management, data processing and analysis within such a platform. The tools used here are briefly described in the following.

3.1 OpenLabFramework

OpenLabFramework [14], [19] was developed to handle the sample management challenges of creating and maintaining large panels of isogenic cell lines and associated vector constructs. The web application was designed in a modular fashion with optional extensions for physical sample tracking, barcode label printing, and an electronic laboratory notebook.

3.2 Sample Management and Visual Analysis of High-throughput Screens (SAVANAH)

SAVANAH [15] was created to deal with the challenges of sample management in HTS. Molecular libraries that are used for screening need to be diluted in multiple iterations, leading to the creation of a large number of microtiter plates sharing basic sample information. Additional information of HTS readouts and the experiment is added when the fully diluted assay plates are selected for screening.

3.3 Microarray R-based Analysis of Complex Lysate Experiments (MIRACLE)

MIRACLE [18] enables researchers to deal with the particular challenges of RPPA sample management owing to the complex spotting process in which originally neighboring samples often end up on distinct locations on the final array. Moreover, MIRACLE provides capabilities for analyzing the resulting data in a user-friendly fashion, including several mathematical models for estimating relative protein amounts from sigmoidal serial dilution curves.

3.4 High-Through Screening kit for R (HiTSeekR)

HiTSeekR [16] is intended as a one-stop solution for several types of HTS experiments and offers dedicated features for gene-targeting, miRNA-targeting and small compound screens. The user is guided through the process of data normalization, during which technical bias is visualized, allowing users to test different strategies in an explorative fashion. Similarly, the user can choose between several strategies for hit selection, i.e. for selecting those samples that show the phenotype of interest. These can then be analyzed on the systems biology level via miRNA target enrichment, gene set enrichment analysis or they can be submitted to KeyPathwayMiner for *de novo* network enrichment.

3.5 KeyPathwayMiner

KeyPathwayMiner [17] integrates experimental information about active genes (in this case, genes identified as potential drug targets) with molecular interaction networks, e.g. protein-protein interaction or gene-regulatory networks. The aim of KeyPathwayMiner is to extract subnetworks that are enriched in active genes. The size of the solution can be controlled by adjusting the number of inactive genes that are allowed.

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd

3.6 Software Integration

As illustrated in Figure 1, the tools presented here are suitable for supporting an experimental drug target screening platform based on functional genomics, HTS and RPPA screening. For avoiding duplication of sample information and for increasing research productivity, these tools share data through web service interfaces and shared infrastructure such as a relational database management system and a search engine. For instance, OpenLabFramework implements a REST web service that is utilized by both, SAVANAH and MIRACLE to link plate based samples to cell lines created in OLF. Moreover, SAVANAH and MIRACLE share a common database and all microtiter plate specific functionality, enabling researchers to generate and access assay-plate and readout data in either of the two applications. This also guarantees that microtiter plate layouts produced in SAVANAH in an HTS experiment can be further processed in MIRACLE to produce RPPA layouts needed for managing protein readouts. SAVANAH offers a REST web service with which raw data and plate layout information can be directly exported to HiTSeekR for normalization, explorative analysis and hit detection. HiTSeekR in turn, uses the Application Programming Interface of KeypathwayMiner to perform *de novo* network enrichment analysis.

4 Discussion

Scientific software is often poorly documented and only tested on a limited number of system configurations. Moreover, researchers often lack the resources to properly maintain software over extended periods of time. Consequently, many potential users of a software tool are already driven off by the challenges of the setup process, including missing or outdated dependencies or unclear installation instructions.

Using Docker, software tools can be executed in a controlled environment. Software developers thus save time by checking only a single system configuration that they can expect to work regardless of the particular system configuration of the prospective users. Moreover, automated build systems can be used to guarantee that the users have access to the latest version of a software tool, while developers simply need to commit their source code in regular intervals.

On the other hand, users no longer need to interpret cryptic installation instructions but can deploy and use complex applications effortlessly. Due to the archiving of dependencies, software tools shipped via Docker remain usable for a wide audience even when the underlying dependencies evolve dramatically or when active development has ceased.

Here, we show that even the deployment of complex workflows and integrated platforms comprising multiple software tools can be greatly simplified through the use of Docker compose. For example, the required infrastructure, e.g. database management systems or key-value stores can be part of the installation. Docker compose further allows multiple instances of a container to be started, thus enabling system administrators to quickly scale their workflows for demand. Moreover, all communication between Docker containers takes place on an isolated virtual network, increasing security at no cost.

The drug target discovery platform described here as an application case has grown organically over the course of 5 years. Duplicating this setup on an independent server system proved difficult and time-consuming in spite of best efforts for documentation. The consistent use of Docker and Docker compose now allows for the same setup to be performed without expert knowledge within two lines of commands.

Docker and Docker compose thus hold great potential for facilitating reproducible research in computational biology and for increasing the robustness of bioinformatics infrastructure in spite of short development cycles and rapid turnover of research staff. However, long-term availability of images hosted on commercial registries such as DockerHub is not clear. Fortunately, the underlying technology is open source and thus current efforts to set up independent and curated Docker registries dedicated to bioinformatics are critical steps on the way to establish Docker as the new community standard for software distribution.

Acknowledgements

Thanks to all former and present members of the NanoCAN Center for Excellence in Nanomedicine at the University of Southern Denmark for their thorough feedback on usability. Furthermore, thanks to Thomas Lengauer for his helpful comments on the manuscript.

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd

Conflict of interest statement: Authors state no conflict of interest. All authors have read the journal's Publication ethics and publication malpractice statement available at the journal's website and hereby confirm that they comply with all its parts applicable to the present scientific work.

References

[1] Löbe M, Ganslandt T, Lotzmann L, Mate S, Christoph J, Baum B, et al. Simplified deployment of health informatics applications by providing docker images. Stud Health Technol Inform. 2016;228:643–7.
[2] Aranguren ME, Wilkinson MD. Enhanced reproducibility of SADI web service workflows with Galaxy and Docker. Gigascience. 2015;4:59.
[3] Muchmore B, Alarcon-Riquelme ME. CymeR: cytometry analysis using KNIME, Docker and R. Bioinformatics. 2017;33:776–778. DOI:10.1093/bioinformatics/btw707.
[4] Devisetty UK, Kennedy K, Sarando P, Merchant N, Lyons E. Bringing your tools to CyVerse discovery environment using Docker. [version 1; referees: 3 approved]. F1000Res. 2016;5:1442.
[5] Moreews F, Sallou O, Ménager H, Le Bras Y, Monjeaud C, Blanchet C, et al. BioShaDock: a community driven bioinformatics shared Docker-based tools registry. [version 1; referees: 2 approved]. F1000Res. 2015;4:1443.
[6] Di Tommaso P, Palumbo E, Chatzou M, Prieto P, Heuer ML, Notredame C. The impact of Docker containers on the performance of genomic pipelines. Peer J. 2015;3:e1273.
[7] Hung LH, Kristiyanto D, Lee SB, Yeung KY. GUIdock: using Docker containers with a common graphics user interface to address the reproducibility of research. PLoS One. 2016;11:e0152686.
[8] Hosny A, Vera-Licona P, Laubenbacher R, Favre T. AlgoRun: a Docker-based packaging system for platform-agnostic implemented algorithms. Bioinformatics. 2016;32:2396–8.
[9] Afgan E, Baker D, van den Beek M, Blankenberg D, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res. 2016;44:W3–10.
[10] Wolstencroft K, Haines R, Fellows D, Williams A, Withers D, Owen S, et al. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Res. 2013;41:W557–61.
[11] Entzeroth M, Flotow H, Condron P. Overview of high-throughput screening. Curr Protoc Pharmacol 2009 Chapter 9:Unit 9.4.
[12] Mohr S, Bakal C, Perrimon N. Genomic screening with RNAi: results and challenges. Annu Rev Biochem. 2010;79:37–64.
[13] Wachter A, Bernhardt S, Beissbarth T, Korf U. Analysis of reverse phase protein array data: from experimental design towards targeted biomarker discovery. Microarrays. 2015;4:520–39.
[14] List M, Schmidt S, Trojnar J, Thomas J, Thomassen M, Kruse TA, et al. Efficient sample tracking with OpenLabFramework. Sci Rep. 2014;4:4278.
[15] List M, Elnegaard MP, Schmidt S, Christiansen H, Tan Q, Mollenhauer J, et al. Efficient management of high-throughput screening libraries with SAVANAH. SLAS Discov. 2017;22:196–202.
[16] List M, Schmidt S, Christiansen H, Rehmsmeier M, Tan Q, Mollenhauer J, et al. Comprehensive analysis of high-throughput screens with HiTSeekR. Nucleic Acids Res. 2016;44:6639–48.
[17] List M, Alcaraz N, Dissing-Hansen M, Ditzel HJ, Mollenhauer J, Baumbach J. KeyPathwayMinerWeb: online multi-omics network enrichment. Nucleic Acids Res. 2016;44:W98–104.
[18] List M, Block I, Pedersen ML, Christiansen H, Schmidt S, Thomassen M, et al. Microarray R-based analysis of complex lysate experiments with MIRACLE. Bioinformatics. 2014;30:i631–8.
[19] List M, Franz M, Tan Q, Mollenhauer J, Baumbach J. OpenLabNotes–an electronic laboratory notebook extension for OpenLabFramework. J Integr Bioinform. 2015;12:274.

Automatically generated rough PDF by *ProofCheck* from River Valley Technologies Ltd