Jan-Gerd Tenberge, Patrick Schiffler
University of Münster, Münster, Germany
jan-gerd.tenberge@uni-muenster.de

Reproducible Neuroimaging Pipelines with Docker

Introduction

Reproducibility has always been regarded a central goal in scientific research. As scientists we share our data, materials, and methods in every scientific paper to allow independent replication and testing of our results (Peng 2011). There are, however, difficulties in achieving this goal if a lot of (possibly unknown) software packages and dependencies between them are involved. The output of a neuroimaging pipeline might depend on the installed versions of tools like Matlab and SPM, additional toolboxes, the underlying operating system, environment variables, configuration files, etc. Even if all of those are known to and shared by an author, replicating the exact configuration is still a difficult task even for well-versed researchers from the same field. We present a way built upon free and open source software that bundles all components and dependencies of a pipeline. Complete pipelines can be shared and used to reliably recreate findings or apply identical processing steps to new datasets on any computer capable of running a recent version of the Docker application. We also present a system for the automated execution and management of such pipelines in abstract 1688.

Methods

Docker (https://www.docker.com) provides a way of packing up and running applications in a reproducible environment, called a container. A Docker container consists of some processes running in a defined environment that is different from the host system's. This environment includes the complete file system as well as environment variables and mapped memory areas. Docker is not a virtualization platform since the kernel is shared between the host system and the running containers but containers are otherwise isolated from the host and each other. A Docker image contains the complete environment necessary to launch a container. An image can be built by executing defined commands in an underlaying base image to extend or alter existing images. A base image will usually contain a Linux distribution like NeuroDebian (Halchenko 2014) and additional software packages will be added in the creation step by running "apt-get" commands or copying local data into the image. Once the image has been built it can be archived as a single immutable file containing and all Docker containers created from this image will have an identical environment regardless of the computer they run on. For an overview of the Docker operations, see in the listing. A workflow for reproducible science with Docker has already been proposed in (Boettiger 2015), the same workflow is applicable to complex pipelines by bundling all applications in a single image. For reusability it is however preferable to build a separate image for each step in the pipeline and specify the flow of data between the containers in a pipeline definition file as seen in the figure. This JSON pipeline definition format is human- and machine-readable and allows for easy manipulation of the pipeline, e.g. by replacing a registration utility in an early step whilst keeping the exact same tools and configurations for all others. These pipeline specifications can also be read by the Neuroimage Processing System (see abstract 1688) to perform the specified steps in an automated, parallel way on a local computer, a compute cluster or a cloud environment. We publish the specification of a pipeline definition file at https://github.com/neuro/nps-docs.
A schematic overview of the system and its components is shown in the figure . It shows the interaction of the single subsystems as well as the reciprocal action between the free, open-source, and self-developed components.

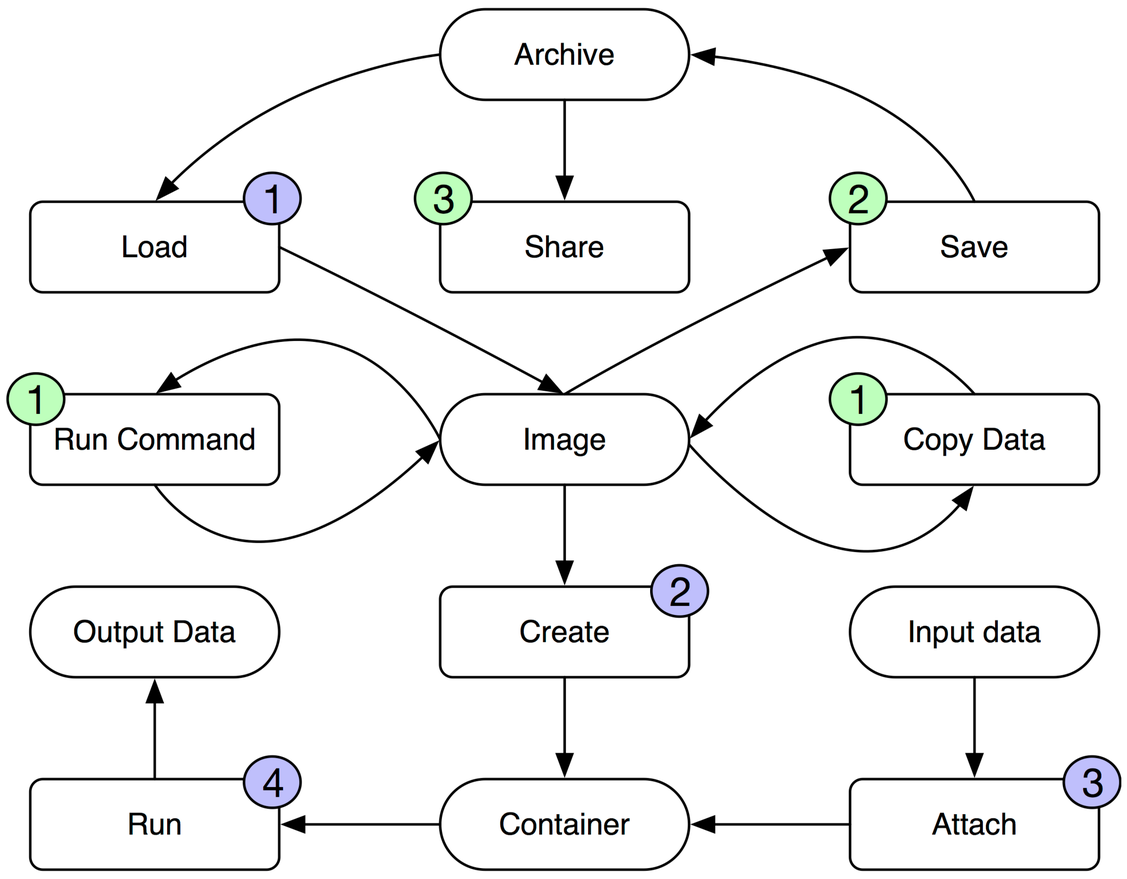
Results

Docker offers an easy way to achieve reproducibility in data processing. Even whole computation pipelines can be describe using Docker. The ability to share containers and thus an exact copy of the data processing steps is a big step towards reproducibility and transparency in scientific data processing.

Conclusions

Software, toolkits, pipelines and their configurations should be shared along with research results. We propose the usage of Docker and pipeline definitions for this task because they are widely supported, require few external dependencies, and can easily be extended upon.

```
{
    "name": "FreeSurfer",
    "description": "Run FreeSurfer on T1 dataset",
    "steps": [
        {
            "image": "4c4fc4bd707e",
            "mounts": [
                {
                    "type": "user",
                    "name": "t1",
                    "path": "/input"
                }
            ]
        },
        {
            "image": "9fe201acd932",
            "mounts": [
                {
                    "type": "step",
                    "name": "convert-t1-to-nifti",
                    "path": "/input/t1"
                }
            ]
        }
    ]
}
```



References

Boettiger, C. (2015). An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review, 49(1), 71-79

Halchenko, Y. O. (2014). NeuroDebian: an integrated, community-driven, free software platform for physiology. In Proceedings of The Physiological Society. The Physiological Society.

Peng, R. D. (2011). Reproducible research in computational science. Science, 334(6060), 1226-1227