

Emerging topic: reproducible  
research methods for genome-scale  
data analysis

Vince Carey, PhD

Harvard Medical School

## CHALLENGES IN IRREPRODUCIBLE RESEARCH

No research paper can ever be considered to be the final word, and the replication and corroboration of research results is key to the scientific process. In studying complex entities, especially animals and human beings, the complexity of the system and of the techniques can all too easily lead to results that seem robust in the lab, and valid to editors and referees of journals, but which do not stand the test of further studies. *Nature* has published a series of articles about the worrying extent to which research results have been found wanting in this respect. The editors of *Nature* and the *Nature* life sciences research journals have also taken substantive steps to put our own houses in order, in improving the transparency and robustness of what we publish. Journals, research laboratories and institutions and funders all have an interest in tackling issues of irreproducibility. We hope that the articles contained in this collection will help.

### Free full access

- ▼ Editorial
- ▼ News and analysis
- ▼ Comment
- ▼ Perspectives and reviews

## EDITORIAL

### Reducing our irreproducibility

*Nature* 496, 398 ( 25 April 2013 )

### Further confirmation needed

A new mechanism for independently replicating research findings is one of several changes required to improve the quality of the biomedical literature.

*Nature Biotechnology* 30, 806 ( 10 September 2012 )

### Error prone

Biologists must realize the pitfalls of work on massive amounts of data.

*Nature* 487, 406 ( 26 July 2012 )

Imperial College

Marie Curie Initiative  
Engineering for  
Medical School

Genomics Manager  
National Cancer  
Health

Pos

### Most read

1. **Mutational t**  
**search for n**  
*Nature* | 16 J
2. **High-molecu**  
**the cancer n**  
*Nature* | 19 J
3. **In vivo cardi**  
**to zebrafish**  
*Nature* | 19 J

*Nature Medicine*  
Volkswagen Foun

**Herrenhaus**  
**Symposium**  
**Cells and Re**  
**Medicine**

October 8-10, 2

ADVERTISEMENT

**INTRODUCING THE S3™ CELL SORTER**  
AUTOMATED SETUP + AFFORDABILITY



[▶ Watch the Video](#)

**BIO-RAD**

ADVERTISEMENT

**nature | methods**  
Techniques for life scientists and chemists

in vitro | in vivo | MICROSCOPY | GENOMICS | RNA | NMR | IMAGING

Search

[nature.com](#) ▶ [journal home](#) ▶ [archive](#) ▶ [issue](#) ▶ [editorial](#) ▶ [full text](#)

NATURE METHODS | EDITORIAL








 **MACMILLAN**  
SCIENCE COMMUNITY

**Submit to**

## Enhancing reproducibility


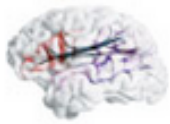
*Nature Methods* **10**, 367 (2013) | doi:10.1038/nmeth.2471  
Published online 29 April 2013

-  [PDF](#)
-  [Citation](#)
-  [Reprints](#)
-  [Rights & permissions](#)
-  [Article metrics](#)

**New reporting standards for Nature journal authors are intended to improve transparency and reproducibility.**

Difficulty in reproducing published biomedical research studies has become a matter of increasing concern that, if unaddressed, will waste limited research funding and may erode public support for research. *Nature Methods* is therefore adopting new editorial measures in an attempt to improve the consistency and quality of reporting in submitted manuscripts.

Selected features  
**Focus on Maps**



# Road map

- Thoughts on barriers to reproducibility
- Integrative documents: Sweave, explicitness and recovery
- Containers for distributable analytic workflows
- VMs for publications/families of publications

# Barriers to reproducibility

- Frank (hidden) error, requires “forensics”
- Excess sensitivity of assay to local conditions
- LOW MILEAGE: only one/few persons used or looked at the tools/runs
  - Some is inevitable: cutting edge assays, statistical procedures
  - Some comes from avoidable difficulties in propagation and redeployment

“Mileage” terminology from Henrik Bengtsson

# Enhancing reproducibility with Sweave and R packages

- Sweave or knitr can improve code/analysis readability/sharing of workflow steps, increasing mileage on comprehension
- Increasing mileage on use: make a package that can be installed in any installation of R with sufficient
  - Version/packages
- biocLite-like facilities can enhance distributability/resolution of dependencies

# CDE: Automatically create portable Linux applications

CDE (formerly known as CDEpack) automatically packages up the **Code, Data, and Environment** required to deploy and run your Linux programs on other machines without any installation or configuration. CDE is the easiest way to completely eliminate dependency hell.

To get started, download the CDE binary ([32-bit](#) or [64-bit](#)) and follow these steps:

## 1. Package



Prepend any set of Linux commands with the "cde" binary, and CDE will run them and automatically package up all files (e.g., executables, libraries, plug-ins, config/data files) accessed during execution.

## 2. Deliver



A package is simply a directory that can be compressed and delivered to any x86-Linux machine. It contains all the files and environment variables required to run your original commands. Packages can range from 10 to 100 MB in size.

## 3. Run



After receiving the package, the user can now run those same commands from within the package on **any** modern x86-Linux distro. The user does not need to first compile, install, or configure anything.



# DOCKER

## The Linux container engine

Docker is an open-source engine which automates the deployment of applications as highly portable, self-sufficient containers which are independent of hardware, language, framework, packaging system and hosting provider.

Let's get started





JUNE 26, 2013

## ANNOUNCING A NEW LOGO AND STYLE FOR DOCKER

The initial release of Docker pretty much took everyone by surprise. That is, including those working on the visual style. So our initial style was put together quickly. We are now proud to announce a new logo and visual style for Docker. Over the past weeks we have worked with some of the best designers of 99Designs to create a logo that fits our project and communicates our values of expedition, automation, encapsulation and simplification. Out of 84 total designs, [a final poll](#) was created. Over 50 people from our community participated to help choose this final logo.

We owe a big thanks to Lachlan Donald (@lox) of 99Designs for generously sponsoring a gold package. It allowed us to work with the best designers in this community. Not surprisingly 99Designs is one of the first companies that started using Docker.

Over the next couple of weeks you will see the new logo and branding appear on our websites, documentation and all other places where you expect Docker. As we focus on improvements across the board you will see the style appear gradually, at the points we touch. — *including of course, T-shirts and stickers!*

### About the logo:



99Designs community.

### SEARCH

### SIGN UP FOR NEWS IN YOUR E-MAIL

[Subscribe](#)

### CATEGORIES

- [Community](#)
- [Demos](#)
- [Design](#)
- [Docker releases](#)
- [Dockerization](#)
- [Features](#)
- [Hackday](#)
- [Index](#)
- [Installation](#)
- [Meetups](#)
- [News](#)
- [OpenStack](#)
- [Talks & presentations](#)
- [Uncategorized](#)

# 14 GREAT TUTORIALS ON DOCKER

Here are 14 tutorials and articles written by the community on different subjects, that would certainly help you improve your docker skills in minutes.

## **Getting Docker to Run on Linode & Push-button Deployment with Docker**

by [Nick Stinemat](#) - Jun 19 2013

> <http://nick.stinemat.es/>

## **Deploy Java Apps With Docker = Awesome**

by [Nicola Paolucci](#) - Jun 13, 2013

> <http://blogs.atlassian.com/2013/06/deploy-java-apps-with-docker-awesome/>

## **Deploying django using docker**

by [Javed Khan](#) - Jun 14, 2013

> <http://agiliq.com/blog/2013/06/deploying-django-using-docker/>

## **Building Your Own Platform Service Using Docker**

by [Jeff Lindsay](#) & [Solomon Hykes](#) at GlueCon 2013 - May 22, 2013

> <http://vimeo.com/67284401>

## **Using Docker to build FireFox**

by [Gregory Szorc](#) - May 19, 2013

> <http://gregoryszorc.com/blog/2013/05/19/using-docker-to-build-firefox/>

# Summary

- For some reasonably modest workflows, a redeployable linux-oriented “package” can be useful distribution targets
- Documentation and checking protocols will be important for achieving scientific utility
- Now turn to a full virtual machine for a large collection of data/software components in the 2012 ENCODE release

- (no quotes, but do include everything else) in the "Share an instance" cluster startup box, and initialize your cluster.
6. CloudMan will now customize your instance as a clone of the ENCODE machine. This process may take a minute or two but is finished when the log shows 'Done running post\_start\_script' and all services are 'green' as indicated by the interface.
  7. Now, simply ssh in to your instance as the 'figures' user and you're good to go. The command should be something like this: `ssh -i cloudman_keypair.pem figures@{your_amazon_instance}'`
  8. Remember to terminate your instance when finished to avoid incurring additional costs. You can always start a new instance with the same cluster name to get back to where you were, until you terminate and delete (via the cloud console).

## Downloadable Virtual Machine

1. [Download VirtualBox](#)
2. Download the Virtual Machine [ENCODE.OVA \(18GB\)](#)
3. Import the VM into VirtualBox
  1. First, make a backup copy of the downloaded .ova file(s). If something goes wrong you can always make a new copy.
  2. Import the VM image into VirtualBox by either starting the downloaded .ova file directly, or by launching VirtualBox and navigating to File → Import Appliance and opening the file.
  3. This will display the Appliance Import Settings window. Click the Import button.
  4. It may then take several minutes for VirtualBox to import the VM. Once it is done, a new VM will appear in the left pane in the 'powered off' state.
  5. The default settings should be mostly appropriate, but one that must be turned on is VT-x/AMD-V Hardware Virtualization. You can find this for your virtual machine in Settings -> System -> Acceleration.
4. Start your new ENCODE VM
5. Log in as the user '**figures**' with the password '**PY4G8GAR**' (case sensitive)

# apologia

We have emphasized transparency in this process, meaning that we have exposed the large diversity of scripting languages and software components used by the various analysts in the project. This diversity in analysis methods should not be a surprise to any scientist working in large-scale genomics, but might be confusing or frustrating for people with less large-scale data handling experience. We apologize in advance for this diversity, but it is important to realize that our goal here is not to provide easy-to-use programs, or robust engineering solutions (there are separately funded projects to create such things), but rather to provide scientific transparency of our analytical results. By having the input data sets, a text description of the method, functioning code implementing the method and finally the output, we hope to provide a highly transparent view of the analysis we have performed. During implementation of the code bundles to establish the VM, there have necessarily been tweaks to the code and installation of packages that had been omitted from the code bundles through oversight. We have trialled the code in VM ourselves, and using only the VM can recreate the expected output.

For inquiries about the content of the supplementary information VM and specifically the content of the code, please email first the joint author email [encode\\_authors@ebi.ac.uk](mailto:encode_authors@ebi.ac.uk) , stating the inquiry and section; please do not email the



# The ENCODE VM layout for users

```
figures@figures-vm:~$ more README
Directories in ~ (/home/figures/) are as follows:
```

```
bin - some executables and script in common between different analyses.
commonData - data in common between some analysis. Structure is the same as the public ftp
             archive as given in the supplementary info.
figures - Runnable versions of the code for each of the 10 figures in the ENCODE integration paper.
          Directories are numbered as for the final figures.
lib - R libraries used in common
manuscript - copies of the submitted version of the manuscript, supplementary info and figures.
R - required R packages
supplementary - code for supplementary information and tables, plus some code dropped during the review
               process.
tables - code for the generation of main paper table 1. Other tables are in supplementary/
```

Overall to run the code for a figure or supplementary info, you should cd into the relevant directory, where there should be a README file with step by step instructions for running the code. In some case, because the analysis is the result of a long and complex pipeline, the code here may work on an intermediate result. Also in some cases the full analysis requires large scale analysis of many datasets which are typically implemented on a compute farm. In these case we have provided a subset of the analysis as an example, which can then be scaled up if the user chooses to do so.

# User layout not unlike an R package/ distro

- But
  - Formal checking of satisfaction of the spec not possible (“Why spec when there’s only one implementation?”)
  - Maintenance/maintainability of the VM code base/VM not clear
  - You can reuse/recreate but adding mileage is hard
- Can we justify thinking about the VM as an evolving analytic workflow artifact, and engineering its evolution like any other important software/data environment?



**Package STATUS** - Package status is indicated by one of the following glyphs:

- **TIMEOUT** BUILD, CHECK or BUILD BIN of package took more than 40 minutes
- **ERROR** BUILD, CHECK or BUILD BIN of package returned an error
- **WARNINGS** CHECK of package produced warnings
- **OK** BUILD, CHECK or BUILD BIN of package was OK
- **skipped** CHECK or BUILD BIN of package was skipped because the BUILD step failed (or because something bad happened with the Build System itself)

Click on any glyph in the report below to see the status details (command output).

Use the check boxes to show only packages with the selected status types:

**TIMEOUT**
 **ERROR**
 **WARNINGS**
 **OK**

SUMMARY		OS / Arch	BUILD	CHECK	BUILD BIN
<a href="#">lamb1</a>		Linux (openSUSE 12.1) / x86_64	0 6 603	0 3 23 577	
<a href="#">moscato1</a>		Windows Server 2008 R2 Enterprise SP1 (64-bit) / x64	0 9 579	0 4 22 553	0 1 578
<a href="#">perceval</a>		Mac OS X Leopard (10.5.8) / i386	0 6 593	0 3 32 558	0 1 592
Package 9/609		Hostname OS / Arch	BUILD	CHECK	BUILD BIN
<b>ACME 2.14.0</b>		<a href="#">lamb1</a> Linux (openSUSE 12.1) / x86_64	<b>OK</b>	<b>ERROR</b>	
Sean Davis		<a href="#">moscato1</a> Windows Server 2008 R2 Enterprise SP1 (64-bit) / x64	<b>OK</b>	<b>ERROR</b>	<b>OK</b>
Last Changed Rev: 70050 / Revision: 72489		<a href="#">perceval</a> Mac OS X Leopard (10.5.8) / i386	<b>OK</b>	<b>ERROR</b>	<b>OK</b>
Last Changed Date: 2012-10-01 15:16:24 -0700		Hostname OS / Arch	BUILD	CHECK	BUILD BIN
Package 15/609		<a href="#">lamb1</a> Linux (openSUSE 12.1) / x86_64	<b>OK</b>	<b>ERROR</b>	
<b>AffyCompatible 1.18.1</b>		<a href="#">moscato1</a> Windows Server 2008 R2 Enterprise SP1 (64-bit) / x64	<b>OK</b>	<b>ERROR</b>	<b>OK</b>
Martin Morgan		<a href="#">perceval</a> Mac OS X Leopard (10.5.8) / i386	<b>OK</b>	<b>ERROR</b>	<b>OK</b>
Last Changed Rev: 72385 / Revision: 72489		Hostname OS / Arch	BUILD	CHECK	BUILD BIN
Last Changed Date: 2013-01-08 17:46:12 -0800		<a href="#">lamb1</a> Linux (openSUSE 12.1) / x86_64	<b>OK</b>	<b>OK</b>	
Package 110/609		<a href="#">moscato1</a> Windows Server 2008 R2 Enterprise SP1 (64-bit) / x64	<b>ERROR</b>	skipped	skipped
<b>ChIPXpress 1.0.0</b>		<a href="#">perceval</a> Mac OS X Leopard (10.5.8) / i386	<b>OK</b>	<b>OK</b>	<b>OK</b>
George Wu		Hostname OS / Arch	BUILD	CHECK	BUILD BIN
Last Changed Rev: 70050 / Revision: 72489		<a href="#">lamb1</a> Linux (openSUSE 12.1) / x86_64	<b>ERROR</b>	skipped	
Last Changed Date: 2012-10-01 15:16:24 -0700		<a href="#">moscato1</a> Windows Server 2008 R2 Enterprise SP1 (64-bit) / x64	<b>ERROR</b>	skipped	skipped
Package 141/609		<a href="#">perceval</a> Mac OS X Leopard (10.5.8) / i386	<b>ERROR</b>	skipped	skipped
<b>cosmo 1.24.0</b>		Hostname OS / Arch	BUILD	CHECK	BUILD BIN
Oliver Bombom		<a href="#">lamb1</a> Linux (openSUSE 12.1) / x86_64	<b>ERROR</b>	skipped	
Last Changed Rev: 70050 / Revision: 72489		<a href="#">moscato1</a> Windows Server 2008 R2 Enterprise SP1 (64-bit) / x64	<b>ERROR</b>	skipped	skipped
Last Changed Date: 2012-10-01 15:16:24 -0700		<a href="#">perceval</a> Mac OS X Leopard (10.5.8) / i386	<b>ERROR</b>	skipped	skipped

[Home](#) » [Help](#) » Cloud AMI

## Bioconductor in the cloud

---

[Obtain](#) an Amazon Web Services account and [start the AMI](#). Additional instructions below.

### Contents

---

- [Overview](#)
- [Preloaded AMI](#)
- [First-Time Steps](#)
- [Launching The AMI](#)
- [Connecting to your AMI using SSH](#)
- [AMI IDs](#)
- [Scenarios for using your Bioconductor instance](#)
- [Using Rgraphviz](#)
- [Parallelization using multicore](#)
- [Using an MPI cluster in the cloud](#)
- [Using a parallel cluster in the cloud](#)
- [Creating a custom version of the Bioconductor AMI](#)
- [Moving data to and from your Bioconductor AMI instance](#)

# Why it may pay off to learn about VMs for workflow component management

- Your hardware will be obtained to satisfy economies of scale
- You may need to use commercial “cloud” approaches to elastic machine acquisition and release
- Bind the VMs to nodes, use multicore within – don’t need (?) a ton of software redesign vs Hadoop or other cloud-oriented programming approaches

From Brad Chapman's [bcbio.wordpress.com](http://bcbio.wordpress.com)

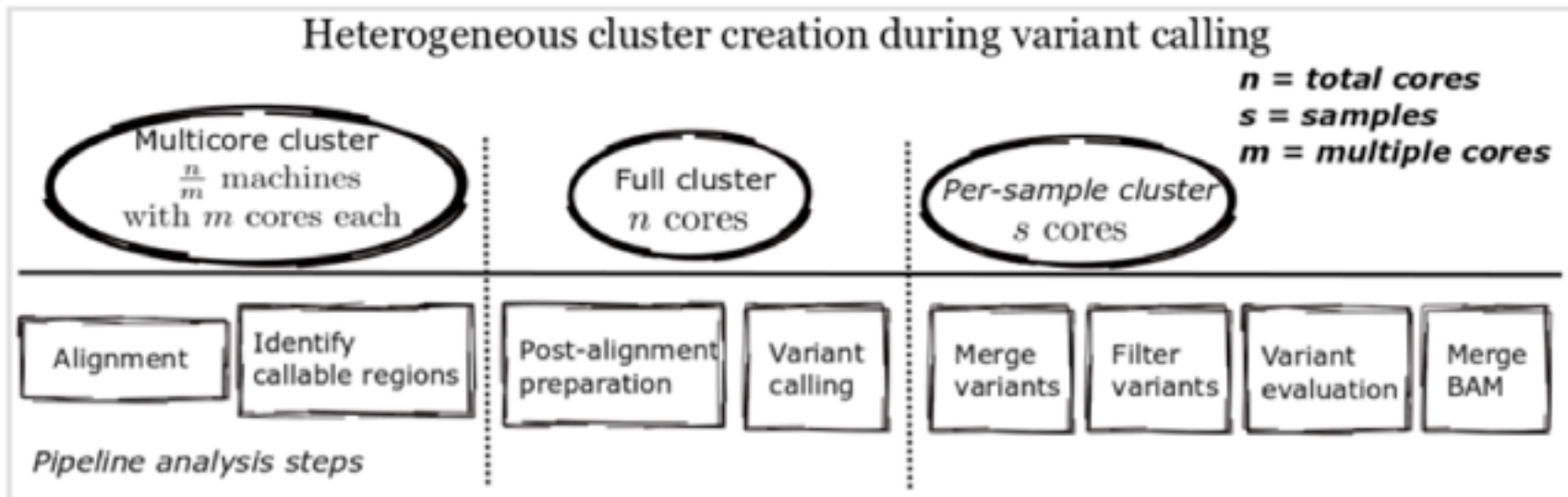


Figure 5. Adaptation of cluster computing model in phases of variant calling.

	primary bottle neck	1 sample 16 cores Lustre	1 sample 96 cores Lustre	1 sample 96 cores NFS	30 samples 480 cores Lustre	30 samples 480 cores NFS
alignment	cpu/mem	4.3h	4.3h	3.9h	4.5h	6.1h
align post-process	io	3.7h	1.0h	0.9h	7.0h	20.7h
variant calling	cpu/mem	2.9h	0.5h	0.5h	3.0h	1.8h
variant post-process	io	1.0h	1.0h	0.6h	4.0h	1.5h
total		11.9h	6.8h	5.9h	18.5h	30.1h

Figure 6. Timing results for variant calling benchmarks as reported in (22).

# Conclusions

- Reproducibility is obviously important for scientific process
  - “reproducible in our hands” not legitimate for statistical procedures
- Sweave/knitr/R packages are helpful for organizing complex data/software interactions for individual/team work
- VMs for large, multilanguage data/software archives can enhance transparency
  - With engineering, can enhance reproducibility and extensibility as well