This page contains basic information and user-friendly guides to running RIS-vetted application containers on Compute1. This is not an inclusive list of usable applications, by any means, and is intended solely as a starting point for new users.

This page does not include application containers where the user does not want, or is forbidden, to put code into a public registry.

The application containers listed on this page have been vetted to start a shell in an interactive job at the time of testing and were confirmed working at the time they were tested on Compute1. The level of security with non-RIS-hosted images is not guaranteed and has not been tested.

As such, we recommend to use RIS-hosted Docker images when possible. Please visit this page for a list of RIS developed Docker images for Compute1.

11falsenonelisttrue

.NET Core

Registry Location: https://hub.docker.com/_/microsoft-dotnet-core-sdk/

".NET Core is an open-source, general-purpose development platform maintained by Microsoft and the .NET community on GitHub. It's cross-platform (supporting Windows, macOS, and Linux) and can be used to build device, cloud, and IoT applications." - Source: https://docs.microsoft.com/en-us/dotnet/core/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(mcr.microsoft.com/dotnet/core/sdk)' /bin/bash]]>

Anaconda

Registry Location:

Using Python 3.5: https://hub.docker.com/r/continuumio/anaconda3

Using Python 2.7: https://hub.docker.com/r/continuumio/anaconda

"Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution and includes over 100 of the most popular Python packages for data science. Additionally, it provides access to over 720 Python and R packages that can easily be installed using the conda dependency and environment manager, which is included in Anaconda." - Source: https://hub.docker.com/r/continuumio/anaconda3

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(continuumio/anaconda3)' /bin/bash # Using Python 2.7: > bsub -G ${group_name} -Is -q general-interactive -a 'docker(continuumio/anaconda)' /bin/bash]]>

AnnovarR

Registry Location: https://registry.hub.docker.com/r/bioinstaller/annovarr

"The annovarR package provides R functions as well as database resources which offer an integrated framework to annotate genetic variants from genome and transcriptome data. The wrapper functions of annovarR unified the interface of many published annotation tools, such as VEP, ANNOVAR, vcfanno and AnnotationDbi." - Source: https://registry.hub.docker.com/r/bioinstaller/annovarr

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(bioinstaller/annovarr)' R]]>

BamTools

Registry Location: https://bioconda.github.io/recipes/bamtools/README.html

"C++ API & command-line toolkit for working with BAM data" - Source: https://bioconda.github.io/recipes/bamtools/README.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bamtools:2.5.1--he860b03_5)' /bin/bash]]>

BCFtools

Registry Location: https://bioconda.github.io/recipes/bcftools/README.html

"BCFtools is a set of utilities that manipulate variant calls in the Variant Call Format (VCF) and its binary counterpart BCF. All commands work transparently with both VCFs and BCFs, both uncompressed and BGZF-compressed. Most commands accept VCF, bgzipped VCF and BCF with filetype detected automatically even when streaming from a pipe. Indexed VCF and BCF will work in all situations. Un-indexed VCF and BCF and streams will work in most, but not all situations." - Source: https://bioconda.github.io/recipes/bcftools/README.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bcftools:1.10.2--hd2cd319_0)' /bin/bash]]>

bedtools

Registry Location: https://bioconda.github.io/recipes/bedtools/README.html

"…fast, flexible toolset for genome arithmetic." - Source: https://bedtools.readthedocs.io/en/latest/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bedtools:2.29.2--hc088bd4_0)' /bin/bash]]>

BLAST

Registry Location: https://bioconda.github.io/recipes/blast/README.html

"Basic Local Alignment Search Tool (BLAST) is a sequence similarity search program." - Source: https://www.ncbi.nlm.nih.gov/pubmed/18440982

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/blast:2.2.31--pl526h3066fca_3)' /bin/bash]]>

Bowtie

Registry Location: https://bioconda.github.io/recipes/bowtie/README.html

"Bowtie is anultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome (2.9 GB for paired-end)." - Source: http://bowtie-bio.sourceforge.net/index.shtml

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bowtie:1.2.3--py37hc9558a2_0)' /bin/bash]]>

Bowtie2

Registry Location: https://quay.io/repository/biocontainers/bowtie2?tab=info

"Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters, and particularly good at aligning to relatively long (e.g. mammalian) genomes. Bowtie 2 indexes the genome with an FM Index to keep its memory footprint small: for the human genome, its memory footprint is typically around 3.2 GB. Bowtie 2 supports gapped, local, and paired-end alignment modes." - Source: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

Run interactive job:

- LSF_DOCKER_PRESERVE_ENVIRONMENT=false bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bowtie2:2.4.1--py38he513fc3_0)' /bin/bash]]>

BreakDancer

Registry Location: https://bioconda.github.io/recipes/breakdancer/README.html

"BreakDancer-1.3.6, is a Cpp package that provides genome-wide detection of structural variants from next generation paired-end sequencing reads. It includes two complementary programs. BreakDancerMax predicts five types of structural variants: insertions, deletions, inversions, inter- and intra-chromosomal translocations from next-generation short paired-end sequencing reads using read pairs that are mapped with unexpected separation distances or orientation. BreakDancerMini focuses on detecting small indels (usually between 10bp and 100bp) using normally mapped read pairs." - Source: https://github.com/genome/breakdancer

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/breakdancer:1.4.5--2)' /bin/bash]]>

BWA

Registry Location: https://bioconda.github.io/recipes/bwa/README.html

“BWA is a software package for mapping DNA sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM.” - Source: https://github.com/lh3/bwa

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bwa:0.7.8--hed695b0_5)' /bin/bash]]>

Cell Ranger

Registry Location: https://hub.docker.com/r/cumulusprod/cellranger

“Cell Ranger is a set of analysis pipelines that process Chromium single-cell RNA-seq output to align reads, generate feature-barcode matrices and perform clustering and gene expression analysis.” - Source: https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger

Run Interactive job:

- PATH=/software/cellranger-4.0.0/:$PATH bsub -G ${group_name} -Is -q general-interactive -a 'docker(cumulusprod/cellranger:4.0.0)' /bin/bash]]>

Circos

Registry Location: https://bioconda.github.io/recipes/circos/README.html

“Circos is a software package for visualizing data and information. It visualizes data in a circular layout — this makes Circos ideal for exploring relationships between objects or positions.” - Source: http://circos.ca/

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/circos:0.69.8--0)' /bin/bash]]>

cn.mops

Registry Location: https://bioconda.github.io/recipes/bioconductor-cn.mops/README.html

“cn.mops (Copy Number estimation by a Mixture Of PoissonS) is a data processing pipeline for copy number variations and aberrations (CNVs and CNAs) from next generation sequencing (NGS) data.” - Source: https://bioconda.github.io/recipes/bioconductor-cn.mops/README.html

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-cn.mops:1.32.0--r36he1b5a44_0)' R]]>

CNVkit

Registry Location: https://bioconda.github.io/recipes/cnvkit/README.html

“A command-line toolkit and Python library for detecting copy number variants and alterations genome-wide from high-throughput sequencing.” - Source: https://github.com/etal/cnvkit

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/cnvkit:0.9.6--py_2)' /bin/bash]]>

Cufflinks

Registry Location: https://bioconda.github.io/recipes/cufflinks/README.html

“Cufflinks assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA-Seq samples. It accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols.” - Source: http://cole-trapnell-lab.github.io/cufflinks/

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/cufflinks:2.2.1--py27_2)' /bin/bash]]>

edgeR

Registry Location: https://bioconda.github.io/recipes/bioconductor-edger/README.html

“Differential expression analysis of RNA-seq expression profiles with biological replication. Implements a range of statistical methodology based on the negative binomial distributions, including empirical Bayes estimation, exact tests, generalized linear models and quasi-likelihood tests. As well as RNA-seq, it be applied to differential signal analysis of other types of genomic data that produce counts, including ChIP-seq, Bisulfite-seq, SAGE and CAGE.” - Source: https://bioconductor.org/packages/3.10/bioc/html/edgeR.html

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-edger:3.28.0--r36he1b5a44_0)' R]]>

Ensembl VEP

Registry Location: https://bioconda.github.io/recipes/ensembl-vep/README.html

“The VEP determines the effect of your variants (SNPs, insertions, deletions, CNVs or structural variants) on genes, transcripts, and protein sequence, as well as regulatory regions.” - Source: https://bioconda.github.io/recipes/ensembl-vep/README.html

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/ensembl-vep:99.0--pl526hecc5488_0)' /bin/bash]]>

freebayes

Registry Location: https://bioconda.github.io/recipes/freebayes/README.html

“freebayes is a Bayesian genetic variant detector designed to find small polymorphisms, specifically SNPs (single-nucleotide polymorphisms), indels (insertions and deletions), MNPs (multi-nucleotide polymorphisms), and complex events (composite insertion and substitution events) smaller than the length of a short-read sequencing alignment.” - Source: https://github.com/ekg/freebayes

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/freebayes:1.3.1--py37h56106d0_0)' /bin/bash]]>

GenomicRanges

Registry Location: https://bioconda.github.io/recipes/bioconductor-genomicranges/README.html

“The ability to efficiently represent and manipulate genomic annotations and alignments is playing a central role when it comes to analyzing high-throughput sequencing data (a.k.a. NGS data). The GenomicRanges package defines general purpose containers for storing and manipulating genomic intervals and variables defined along a genome. More specialized containers for representing and manipulating short alignments against a reference genome, or a matrix-like summarization of an experiment, are defined in the GenomicAlignments and SummarizedExperiment packages, respectively. Both packages build on top of the GenomicRanges infrastructure.” - Source: https://bioconductor.org/packages/3.10/bioc/html/GenomicRanges.html

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-genomicranges:1.38.0--r36h516909a_0)' R]]>

GenVisR

Registry Location: https://bioconda.github.io/recipes/bioconductor-genvisr/README.html

“Produce highly customizable publication quality graphics for genomic data primarily at the cohort level.” - Source: https://bioconductor.org/packages/3.10/bioc/html/GenVisR.html

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-genvisr:1.18.1--r36_0)' R]]>

Go

Registry Location: https://hub.docker.com/_/golang

“Go is syntactically similar to C, but with memory safety, garbage collection, structural typing,[6] and CSP-style concurrency.” - Source: https://en.wikipedia.org/wiki/Go_%28programming_language%29

Run Interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(golang)' /bin/bash]]>

GROMACS

Registry Location: https://hub.docker.com/r/gromacs/gromacs

“GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.” - Source: http://www.gromacs.org/About_Gromacs

Run interactive job:

- export PATH=$PATH:/gromacs/bin > bsub -G ${group_name} -Is -q general-interactive -a 'docker(gromacs/gromacs)' /bin/bash]]>

HISAT2

Registry Location: https://bioconda.github.io/recipes/hisat2/README.html

“HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads (both DNA and RNA) to a population of human genomes (as well as to a single reference genome). Based on an extension of BWT for graphs [Sirén et al. 2014], we designed and implemented a graph FM index (GFM), an original approach and its first implementation to the best of our knowledge. In addition to using one global GFM index that represents a population of human genomes, HISAT2 uses a large set of small GFM indexes that collectively cover the whole genome (each index representing a genomic region of 56 Kbp, with 55,000 indexes needed to cover the human population). These small indexes (called local indexes), combined with several alignment strategies, enable rapid and accurate alignment of sequencing reads. This new indexing scheme is called a Hierarchical Graph FM index (HGFM).” - Source: https://ccb.jhu.edu/software/hisat2/index.shtml

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/hisat2:2.1.0--py37hc9558a2_4)' /bin/bash]]>

HOOMD

Registry Location: https://hub.docker.com/r/glotzerlab/software/

“HOOMD-blue is a general-purpose particle simulation toolkit. It scales from a single CPU core to thousands of GPUs. You define particle initial conditions and interactions in a high-level python script. Then tell HOOMD-blue how you want to execute the job and it takes care of the rest. Python job scripts give you unlimited flexibility to create custom initialization routines, control simulation parameters, and perform in situ analysis.” - Source: http://glotzerlab.engin.umich.edu/home/resources/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(glotzerlab/software)' /bin/bash]]>

IRanges

Registry Location: https://bioconda.github.io/recipes/bioconductor-iranges/README.html

“Provides efficient low-level and highly reusable S4 classes for storing, manipulating and aggregating over annotated ranges of integers. Implements an algebra of range operations, including efficient algorithms for finding overlaps and nearest neighbors. Defines efficient list-like classes for storing, transforming and aggregating large grouped data, i.e., collections of atomic vectors and DataFrames.” - Source: https://bioconductor.org/packages/3.10/bioc/html/IRanges.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-iranges:2.20.0--r36h516909a_0)' R]]>

Julia

Registry Location: https://hub.docker.com/_/julia

“Julia is a high-level, high-performance dynamic programming language for technical computing, with syntax that is familiar to users of other technical computing environments. It provides a sophisticated compiler, distributed parallel execution, numerical accuracy, and an extensive mathematical function library.” - Source: https://julialang.org/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(julia:alpine)' /bin/bash]]>

Jupyter

Registry Location: https://hub.docker.com/r/jupyter/datascience-notebook

“Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data.” - Source: https://jupyter.org

“jupyter/datascience-notebook includes libraries for data analysis from the Julia, Python, and R communities.” - Source: https://jupyter-docker-stacks.readthedocs.io/en/latest/using/selecting.html#jupyter-datascience-notebook

Run interactive job:

Choose a port number from 8000 to 8999 to be able to connect your browser to. In the example below, this value is set to 8888.

Please see our documentation for more information on selecting a port.

Submit the interactive job:

- NB_USER=$USER XDG_CACHE_HOME=$HOME/.cache LSF_DOCKER_PORTS='8888:8888' PATH="/opt/conda/bin:$PATH" bsub -G ${group_name} -Is -q general-interactive -R 'select[port8888=1]' -a 'docker(jupyter/datascience-notebook:ubuntu-20.04)' jupyter-notebook --ip=0.0.0.0 --NotebookApp.allow_origin=* # Running JupyterLab > NB_USER=$USER XDG_CACHE_HOME=$HOME/.cache JUPYTER_ENABLE_LAB=True LSF_DOCKER_PORTS='8888:8888' PATH="/opt/conda/bin:$PATH" bsub -G ${group_name} -Is -q general-interactive -R 'select[port8888=1]' -a 'docker(jupyter/datascience-notebook:ubuntu-20.04)' jupyter-lab --ip=0.0.0.0 --NotebookApp.allow_origin=*]]>

Connect to your server by pointing your web browser to the node, port and token from the output of the interactive job.

-]]>

When you are finished with your server, use Ctrl+C to stop this server (twice to skip confirmation) and “exit” the interactive shell.

If you encounter an error stating your shell has not been properly configured, please see this section on initializing your shell to use conda.

If you encounter an error during startup, specifically with applications such as fluxbox, x11vnc, or xterm, in addition to seeing something like “(exited status 1; not expected)” in your terminal, you will need to ssh into the same host again. Assuming you are currently on `compute1-client-1.ris.wustl.edu`, please ssh again to `compute1-client-1.ris.wustl.edu` and try to run your job again. Meaning, SSH from your local machine to compute1 and then from compute1 again to compute1. For example: localhost –> compute1-client-1 –> compute1-client-1 These issues appear to exist due to incompatible local terminal configuration, MobaXTerm for Windows and Terminal for Mac, being carried forward and applied on the compute1 system.

Mac Alternative: please install iTerm2, ssh back into compute1 via iTerm2, and submit your job again as normal.

Installation can be performed through homebrew, if already installed, or through direct download via the following link

Direct Download: https://iterm2.com/downloads.html

iTerm2 Documentation: https://iterm2.com/documentation.html

Windows Alternative: please install Windows Terminal from the Microsoft Store and enable the OpenSSH client under “Optional Features”

Windows Terminal: https://learn.microsoft.com/en-us/windows/terminal/install OpenSSH Client: https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui#install-openssh-for-windows

If the expected python/conda environment is not loaded by Jupyter and unable to be selected through the GUI’s “kernel” option atop the window, ipykernel should be installed. This process is performed by simply installing ipykernel within a given python/conda environment, followed by registering that environment with the newly installed kernel (ipykernel). Once completed, restart the running Jupyter container. The configured python environment should now be accessible through the aforementioned “kernel” option, atop the Jupyter Notebooks web GUI. (This assumes previous env setup was properly completed.)

Install necessary package python3 -m pip install ipykernel # Assign a name to be seen in the Jupyter Notebooks "kernels" option drop down python3 -m ipykernel install --name]]>

For further information please see the official https://ipython.readthedocs.io/en/stable/install/index.html#installation

GPU-enabled Jupyter

GPU-enabled Jupyter notebooks are available via the RAPIDS Docker image.

maftools

Registry Location: https://bioconda.github.io/recipes/bioconductor-maftools/README.html

“Analyze and visualize Mutation Annotation Format (MAF) files from large scale sequencing studies. This package provides various functions to perform most commonly used analyses in cancer genomics and to create feature rich customizable visualzations with minimal effort.” - Source: https://bioconductor.org/packages/3.10/bioc/html/maftools.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-maftools:2.2.0--r36_0)' R]]>

Manta

Registry Location: https://bioconda.github.io/recipes/manta/README.html

“Structural variant and indel caller for mapped sequencing data” - Source: https://bioconda.github.io/recipes/manta/README.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/manta:1.6.0--py27_0)' /bin/bash]]>

monocle

Registry Location: https://bioconda.github.io/recipes/bioconductor-monocle/README.html

“Monocle performs differential expression and time-series analysis for single-cell expression experiments. It orders individual cells according to progress through a biological process, without knowing ahead of time which genes define progress through that process. Monocle also performs differential expression analysis, clustering, visualization, and other useful tasks on single cell expression data. It is designed to work with RNA-Seq and qPCR data, but could be used with other types as well.” - Source: https://bioconductor.org/packages/release/bioc/html/monocle.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-monocle:2.14.0--r36he1b5a44_1)' R]]>

NovoAlign

Registry Location: https://bioconda.github.io/recipes/novoalign/README.html

“Powerful tool designed for mapping of short reads onto a reference genome from Illumina, Ion Torrent, and 454 NGS platforms.” - Source: http://www.novocraft.com/products/novoalign/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/novoalign:3.09.00--h7311fa2_2)' /bin/bash]]>

OpenJDK (Java)

Registry Location: https://hub.docker.com/_/openjdk

“OpenJDK (Open Java Development Kit) is a free and open-source implementation of the Java Platform, Standard Edition (Java SE).” - Source: https://en.wikipedia.org/wiki/OpenJDK

Run interactive job:

- LSF_DOCKER_PRESERVE_ENVIRONMENT=false bsub -G ${group_name} -Is -q general-interactive -a 'docker(openjdk:11-slim)' /bin/bash # Using Java 8: > LSF_DOCKER_PRESERVE_ENVIRONMENT=false bsub -G ${group_name} -Is -q general-interactive -a 'docker(openjdk:8-slim)' /bin/bash]]>

Oncotator

Registry Location: https://bioconda.github.io/recipes/oncotator/README.html

“Oncotator is a tool for annotating human genomic point mutations and indels with data relevant to cancer researchers.” - Source: https://software.broadinstitute.org/cancer/cga/oncotator

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/oncotator:1.9.9.0--py27h516909a_2)' /bin/bash]]>

Organism.dplyr

Registry Location: https://bioconda.github.io/recipes/bioconductor-organism.dplyr/README.html

“This package provides an alternative interface to Bioconductor ‘annotation’ resources, in particular the gene identifier mapping functionality of the ‘org’ packages (e.g., org.Hs.eg.db) and the genome coordinate functionality of the ‘TxDb’ packages (e.g., TxDb.Hsapiens.UCSC.hg38.knownGene).” - Source: https://bioconductor.org/packages/3.10/bioc/html/Organism.dplyr.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-organism.dplyr:1.14.0--r36_0)' R]]>

Perl

Registry Location: https://hub.docker.com/_/perl

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(perl:slim)' /bin/bash]]>

Picard

Registry Location: https://bioconda.github.io/recipes/picard/README.html

“Picard is a set of command line tools for manipulating high-throughput sequencing (HTS) data and formats such as SAM/BAM/CRAM and VCF.” - Source: http://broadinstitute.github.io/picard/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/picard:2.21.7--0)' /bin/bash]]>

Pindel

Registry Location: https://bioconda.github.io/recipes/pindel/README.html

“Pindel can detect breakpoints of large deletions, medium sized insertions, inversions, tandem duplications and other structural variants at single-based resolution from next-gen sequence data. It uses a pattern growth approach to identify the breakpoints of these variants from paired-end short reads.” - Source: http://gmt.genome.wustl.edu/packages/pindel/index.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/pindel:0.2.5b9--he527e40_3)' /bin/bash]]>

PLINK

Registry Location: https://bioconda.github.io/recipes/plink/README.html

“PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, large-scale analyses in a computationally efficient manner.” - Source: http://zzz.bwh.harvard.edu/plink/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/plink:1.90b6.12--heea4ae3_0)' /bin/bash]]>

PLINK2

Registry Location: https://hub.docker.com/r/skwalker/plink2

“The new release…is a complete rewrite of the original code and represents a very significant improvement in overall speed and functionality. Moving forward, these changes should enable PLINK to meet the demands of ever-larger genetic datasets.” - Source: https://zzz.bwh.harvard.edu/plink/plink2.shtml

Run interactive job:

- PATH=$PATH:/usr bsub -G ${group_name} -Is -q general-interactive -a 'docker(skwalker/plink2)' /bin/bash]]>

PRSice-2

Registry Location: https://hub.docker.com/r/lifebitai/prsice2

“PRSice (pronounced ‘precise’) is a Polygenic Risk Score software for calculating, applying, evaluating and plotting the results of polygenic risk scores (PRS) analyses.” - Source: https://www.prsice.info/

Run interactive job:

- PATH="/opt/conda/bin:$PATH" bsub -G ${group_name} -Is -q general-interactive -a "docker(lifebitai/prsice2)" /bin/bash]]>

Activate conda environment:

conda activate prs]]>

You will now be able to use `PRSice.R` and `PRSice_linux` commands.

If you encounter an error stating your shell has not been properly configured, please see this section on initializing your shell to use conda.

Python

Registry Location: https://hub.docker.com/_/python

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(python:slim)' /bin/bash]]>

QIIME2

Registry Location: https://hub.docker.com/r/qiime2/core

“QIIME 2™ is a next-generation microbiome bioinformatics platform that is extensible, free, open source, and community developed.” - Source: https://qiime2.org/

Run interactive job:

- LSF_DOCKER_PRESERVE_ENVIRONMENT=false bsub -G ${group_name} -Is -q general-interactive -a 'docker(qiime2/core:2020.2)' /bin/bash]]>

R

Registry Location: https://hub.docker.com/_/r-base

“R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files.” - Source: https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f

Run interactive job:

bsub -G ${group_name} -Is -q general-interactive -a 'docker(r-base:)' R]]>

Where `<tag>` is replaced by the version tag you wish to use. These can be found on the r-base Docker Hub page.

R Version Issues

There are known issues with the latest image of Debian which affects the r-base images and causes some commands not to cooperate in regards to the Storage Platform.

The latest version (4.1.2) also will not run R as expected and users should use an earlier version than the latest.

These are known and the RIS Team is working on fixing these issues.

RAPIDS GPU-enabled Jupyter

Registry Location: https://hub.docker.com/r/rapidsai/rapidsai

“The RAPIDS suite of open source software libraries and APIs gives you the ability to execute end-to-end data science and analytics pipelines entirely on GPUs.” - Source: https://rapids.ai/about.html

Run interactive job:

Choose a port number from 8000 to 8999 to be able to connect your browser to. In the example, this value is set to 8888.

Please see our documentation for more information on selecting a port.

Submit the interactive job:

LSF_DOCKER_PORTS="8888:8888" PATH="/opt/conda/bin:$PATH" bsub -Is -q general-interactive -R 'select[gpuhost && port8888=1]' -gpu "num=1:gmodel=TeslaV100_SXM2_32GB" -a 'docker(rapidsai/rapidsai:0.18-cuda11.0-runtime-ubuntu20.04-py3.8)' /bin/bash]]>

Connect to your server by pointing your web browser to the node, port and token from the output of the interactive job.

When you are finished with your server, use Ctrl+C to stop this server (twice to skip confirmation) and “exit” the interactive shell.

If you encounter an error stating your shell has not been properly configured, please see this section on initializing your shell to use conda.

Setting Up A Jupyter Notebook Password

It is recommended to password-protect your Jupyter sessions. This can be done after submitting a RAPIDS interactive job. After your interactive job has landed, please follow the instructions below.

Start an iPython session.

ipython]]>

Enter in the following commands at the iPython prompt, each in their own line.

You will be prompted to enter and verify a password. The result will be a string of characters.

Copy the string of characters and exit out of iPython and RAPIDS interactive job.

exit]]>

Open the Jupyter configuration file with `Vim`.

Note

For help on using Vim, please see this https://vimhelp.org/ .

vim ~/.jupyter/jupyter_notebook_config.py]]>

Add the copied string of characters to the `c.NotebookApp.password` variable. The file should look like the example below (replacing the string after u with the string you copied):

Save the file and exit out of `Vim`.

From now on, when you submit a job that uses Jupyter notebook, you will be prompted to enter your password.

If you ever forget the password, delete the configuration file and start over.

rm ~/.jupyter/jupyter_notebook_config.py]]>

RStudio

Registry Location: https://hub.docker.com/r/rocker/verse

RStudio a is graphical interface for using R + python and its packages. See above for the description of ‘R’

Source: https://rstudio.com/

Version 4.0.2

Note this is vetted only for image version up to 4.0.2. Use koetjen/rstudio:4.0.3 for R version 4.0.3 and RStudio version 1.3.1093.

An initial setup of dependent files and directory structure is required. Please see this section for details.

Choose a port number from 8001 to 8999 to be able to connect your browser to.

Please see our documentation for more information on selecting a port.

In the example, this value is set to 8081

Note: you are telling compute1 to reserve this port exclusively for you on the node.

Run interactive job:

- LSF_DOCKER_VOLUMES='/home/${compute_username}:/home/${compute_username}' PATH=/home/${compute_username}:$PATH LSF_DOCKER_PORTS='8081:8787' bsub -Is -G ${group_name} -q general-interactive -R 'select[port8081=1]' -a 'docker(rocker/verse:4.0.2)' /bin/bash > rstudio-server start]]>

The node your job lands on, typically looks like <<Starting on compute1-exec-nn.ris.wustl.edu>>, where `nn` is the node number.

Connect to your server using the node and port from the interactive job, replacing `nn` with the node your job landed on and `${port}` with the port chosen earlier. If you followed the example interactive job, the port is 8081:

Point your web browser to http://compute1-exec-nn.ris.wustl.edu:${port}

When you are finished with your server, Ctrl+C to stop it and “exit” the interactive shell.

Version 4.1.0

Requirements

Please follow the initial setup steps for RStudio Version 4.0.2 before continuing

Create a directory for the RStudio database files

- mkdir -p $HOME/rstudio_db]]>

Run an interactive job using port 8081 (Please see our documentation for more information on selecting a port for RStudio.

- LSF_DOCKER_VOLUMES="$HOME:$HOME $HOME/rstudio_db/:/var/lib/rstudio-server" PATH="$HOME:$PATH" LSF_DOCKER_PORTS='8081:8787' bsub -Is -G ${group_name} -q general-interactive -R 'select[port8081=1]' -a 'docker(rocker/verse:4.1.0)' /bin/bash > rstudio-server start]]>

The node your job lands on, typically looks like <<Starting on compute1-exec-nn.ris.wustl.edu>>, where `nn` is the node number.

Connect to your RStudio 4.1.0 job using the same directions for version 4.0.2.

samtools

Registry Location: https://bioconda.github.io/recipes/samtools/README.html

“Samtools is a suite of programs for interacting with high-throughput sequencing data. It consists of three separate repositories: Samtools - Reading/writing/editing/indexing/viewing SAM/BAM/CRAM format, BCFtools - Reading/writing BCF2/VCF/gVCF files and calling/filtering/summarising SNP and short indel sequence variants, HTSlib - A C library for reading/writing high-throughput sequencing data” - Source: https://www.htslib.org/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/samtools:1.2-0)' /bin/bash]]>

Seurat

Registry Location: https://hub.docker.com/r/koetjen/rstudio

“R package designed for QC, analysis, and exploration of single-cell RNA-seq data.” - Source: https://satijalab.org/seurat/

Run an interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(koetjen/rstudio:seurat5)' R]]>

GUI can be run following instructions shown at https://washu.atlassian.net/wiki/spaces/RUD/pages/1782055006/noVNC?atlOrigin=eyJpIjoiZGJlZTM4ZGI3ZDllNDQyZmFiNmNlMDA2YzllOThiNjUiLCJwIjoiYyJ9 .

RIS also hosts Seurat images. More information can be found at https://washu.atlassian.net/wiki/spaces/RUD/pages/1782382858/Rstudio?atlOrigin=eyJpIjoiZjIwYmU1ZjQ3MjcwNGY0MTk5MmY1MmRmYzUwMmQyYzkiLCJwIjoiYyJ9 .

seurat-scripts

Registry Location: https://bioconda.github.io/recipes/seurat-scripts/README.html

“A set of wrappers for individual components of the Seurat package. …this package adds a set of simple wrappers with robust argument parsing. …ultimate objective is to have language-agnostic intermediate formats allowing composite workflows using a variety of software packages.” - Source: https://bioconda.github.io/recipes/seurat-scripts/README.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/seurat-scripts:0.0.6--0)' R]]>

Shiny

Registry Location: https://hub.docker.com/r/rocker/shiny

“Shiny is an R package that makes it easy to build interactive web apps straight from R.” - Source: https://shiny.rstudio.com/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(rocker/shiny)' R]]>

SingleR

Registry Location: https://bioconda.github.io/recipes/bioconductor-singler/README.html

“Performs unbiased cell type recognition from single-cell RNA sequencing data, by leveraging reference transcriptomic datasets of pure cell types to infer the cell of origin of each single cell independently.” - Source: https://bioconductor.org/packages/3.10/bioc/html/SingleR.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/bioconductor-singler:1.0.0--r36he1b5a44_0)' R]]>

SnpSift

Registry Location: https://bioconda.github.io/recipes/snpsift/README.html

“SnpSift is a toolbox that allows you to filter and manipulate annotated files.” - Source: http://snpeff.sourceforge.net/SnpSift.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/snpsift:4.2--4)' /bin/bash]]>

STAR

Registry Location: https://bioconda.github.io/recipes/star/README.html

“Ultrafast universal RNA-seq aligner.” - Source: https://www.ncbi.nlm.nih.gov/pubmed/23104886

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/star:2.7.3a--0)' /bin/bash]]>

SRA Toolkit

Registry Location:

“The SRA Toolkit, and the source-code SRA System Development Kit (SDK), will allow you to programmatically access data housed within SRA and convert it from the SRA format to the following formats: ABI SOLiD native (colorspace fasta / qual), fasta, fastq, sff, sam (human-readable bam, aligned or unaligned), Illumina native…” - Source: https://www.ncbi.nlm.nih.gov/books/NBK158900/

Run interactive job:

- PATH="/usr/local/ncbi/sra-tools/bin:$PATH" bsub -G ${group_name} -Is -q general-interactive -a "docker(inutano/sra-toolkit:latest)" /bin/sh]]>

Strelka2

Registry Location: https://bioconda.github.io/recipes/strelka/README.html

“Strelka2 is a fast and accurate small variant caller optimized for analysis of germline variation in small cohorts and somatic variation in tumor/normal sample pairs.” - Source: https://github.com/Illumina/strelka

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/strelka:2.9.10--0)' /bin/bash]]>

TopHat

Registry Location: https://bioconda.github.io/recipes/tophat/README.html

“TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.” - Source: http://ccb.jhu.edu/software/tophat/index.shtml

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/tophat:2.1.1--py27_3)' /bin/bash]]>

Valgrind

Registry Location: https://hub.docker.com/r/greymail/gcc-cmake-valgrind

“Valgrind is an instrumentation framework for building dynamic analysis tools. There are Valgrind tools that can automatically detect many memory management and threading bugs, and profile your programs in detail. You can also use Valgrind to build new tools.” - Source: https://valgrind.org/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(greymail/gcc-cmake-valgrind:0.1)' /bin/bash]]>

Varscan

Registry Location: https://bioconda.github.io/recipes/varscan/README.html

“VarScan is a platform-independent mutation caller for targeted, exome, and whole-genome resequencing data generated on Illumina, SOLiD, Life/PGM, Roche/454, and similar instruments. The newest version, VarScan 2, is written in Java, so it runs on most operating systems. It can be used to detect different types of variation.” - Source: http://dkoboldt.github.io/varscan/

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/varscan:2.3.7--3)' /bin/bash]]>

VCFtools

Registry Location: https://bioconda.github.io/recipes/vcftools/README.html

“A set of tools written in Perl and C++ for working with VCF files. This package only contains the C++ libraries whereas the package perl-vcftools-vcf contains the perl libraries.” - Source: https://bioconda.github.io/recipes/vcftools/README.html

Run interactive job:

- bsub -G ${group_name} -Is -q general-interactive -a 'docker(quay.io/biocontainers/vcftools:0.1.12b--pl526hdbcaa40_0)' /bin/bash]]>

Velocyto

Registry Location: https://hub.docker.com/r/genomicpariscentre/velocyto

“Velocyto is a package for the analysis of expression dynamics in single cell RNA seq data.” - Source: http://velocyto.org/

Run interactive job:

- PATH="/opt/conda/bin:$PATH" bsub -G ${group_name} -Is -q general-interactive -a "docker(genomicpariscentre/velocyto)" /bin/bash]]>

  Activate conda environment:

  conda activate base]]>

  You will now be able to use `velocyto` command.

If you encounter an error stating your shell has not been properly configured, please see this section on initializing your shell to use conda.

Anaconda Environment Errors

If you encounter the following error when invoking the `conda` command:

You will need to initialize your shell for use with Anaconda. This only has to be done once. Copy and paste the code block below into your terminal.

> In rare cases, $HOME is not /home/${compute_username}. See the quick-start for details.
>
> If this is the case echo $HOME and use this instead at the appropriate places below.

> $HOME/.bashrc # >>> conda initialize >>> # !! Contents within this block are managed by 'conda init' !! __conda_setup="$('/opt/conda/bin/conda' 'shell.bash' 'hook' 2> /dev/null)" if [$? -eq 0]; then eval "$__conda_setup" else if [-f "/opt/conda/etc/profile.d/conda.sh"]; then . "/opt/conda/etc/profile.d/conda.sh" else export PATH="/opt/conda/bin:$PATH" fi fi unset __conda_setup # <<< conda initialize <<< EOF]]>

Re-initialize your `bashrc`

You should now be able to activate an Anaconda environment.

Port Selection

Some applications require access to a port inside the running Docker container to run an application. Two examples would be Jupyter and RStudio, which use ports 8888 and 8787 respectively. When submitting a job, a port is exposed and forwarded to a port inside the Docker container using the `LSF_DOCKER_PORTS` environment variable. Please see our documentation for more information on using `LSF_DOCKER_PORTS`.

The Jupyter example job submissions expose port 8888 to access the running Jupyter Docker container while the RStudio example job submissions expose port 8000. If you choose to use a different port, you will also need to update the job submission command. This is true for all jobs requiring access to a port.

For example, if an application is running on port 8999, and you would like to expose port 8001 to access the application running in the Docker container on port 8999, you would need to update the job submission command as follows:

Modify `LSF_DOCKER_PORTS`:

Modify the LSF port resource request:

Notice that only the first port in `LSF_DOCKER_PORTS` is changed since port 8999 is the port defined by the Docker image maintainer to run the application with. Any port between 8000-8999 can be exposed.

If the application requires access via a web browser, the port in the URL will also need to be changed. As an example, to access a Jupyter session using port 8001, the URL would need to be changed to:

]]>