CyVerse Documentation

Release 1.0

CyVerse

Jul 12, 2017

Contents

Learning Center Home

CHAPTER 1

Goal

Analyze RNA-Seq data for differential expression. Kallisto is a quick, highly-efficient software for quantifying transcript abundances in an RNA-Seq experiment. Even on a typical laptop, Kallisto can quantify 30 million reads in less than 3 minutes. Integrated into CyVerse, you can take advantage of CyVerse data management tools to process your reads, do the Kallisto quantification, and analyze your reads with the Kallisto companion software Sleuth in an R-Studio environment.


Learning Center Home

Organize Kallisto Input Data

Description:

Kallisto has relatively few input requirements. You will need to have either single or paired end reads, as well as a refference transcriptome. It is suggested that your RNA-Seq reads are analyzed using FastQC, followed by any additional trimming and filtering using and application such as Trimmomatic.

Note: About the Sample Dataset In this tutorial, we are using publically available data from the SRA. This tutorial will start with cleaned and processed reads. The SRA experiment used data from bioproject PRJNA272719 https://www.ncbi.nlm.nih.gov/bioproject/PRJNA272719. The abstract from that project is reprinted here:

'To survey transcriptome changes by the mutations of a DNA demethylase ROS1 responding to a phytohormone abscisic acid, we performed the Next-gen sequencing (NGS) associated RNA-seq analysis. Two ROS1 knockout lines

(ros1-3, ros1-4; Penterman et al. 2007 [PMID: 17409185]) with the wild-type Col line (wt) were subjected. Overall design: Three samples (ros1-3, ros1-4 and wt), biological triplicates, ABA or mock treatment, using Illumina HiSeq 2500 system' [citation].

Input Data:

Input	Description	Example
Refference transcriptome	fasta	Example transcriptome

Importing Reference Transcriptome

In this example, we will import a reference transcriptome for Arabidopsis from Ensembl. In many cases you can find an appropriate transcriptome from Ensembl for your organism of interest, or provide your own fasta-formmatted transcriptome.

1. Go to the Ensembl Plants Arabidopsis page: http://plants.ensembl.org/Arabidopsis_thaliana/Info/Index
2. In the 'Gene annotation' section, click on the 'Download genes, cDNAs, ncRNA, proteins' 'FASTA' link: ftp://ftp.ensemblgenomes.org/pub/plants/release-36/fasta/arabidopsis_thaliana/
3. The transcriptome files will be located in the 'cdna' folder: ftp://ftp.ensemblgenomes.org/pub/plants/release-36/fasta/arabidopsis_thaliana/cdna/
4. Right-click/command-click on the *Arabidopsis_thaliana.TAIR10.cdna.all.fa.gz* and copy the link location/URL to the clipboard
5. Login to the CyVerse Discovery Environment
6. In your home directory, create a folder to organize your Kallisto project
7. In the created folder, go to the 'Upload' menu, and select, 'Import from URL...'; paste in the Ensembl link and click 'Import from URL' to begin the import

Output/Results

Output	Description	Example
Reference transcriptome	fasta	Example transcriptome

Description

This example transcriptome will be indexed in the next step. RNA-Seq reads will be mapped against this set of transcripts.

Summary

Once you have the transcriptome and your RNA-Seq reads, you can proceed with the next step. We suggest organizing your RNA-Seq reads in the folder created in step 6 above.

More help and additional information

Search for an answer: CyVerse Learning Center or CyVerse Wiki

Post your question to the user forum: Ask CyVerse

Fix or improve this documentation

- On Github: Repo link
- Send feedback: Tutorials@CyVerse.org

Learning Center Home

Learning Center Home

Build Kallisto Transcriptome Index

Description:

As described in the Kallisto paper, RNA-Seq reads are efficiently mapped through a pseudoalignment process against a reference transcriptome index. We will build the index in this step.

Input Data:

Input	Description	Example
Reference transcriptome	fasta	Example transcriptome

Build Kallisto Index

We will now index the Arabidopsis transcriptome imported from Ensembl. This transcriptome can be used multiple times for future Kallisto analyses and only needs to be made once.

1. If necessary, login to the CyVerse Discovery Environment
2. Open the Kallisto-0.42.3-INDEX App
3. Name your analysis, and if desired enter comments. In the App's 'Input' step under 'Index name' enter a name for your index. For this tutorial, name your index **Arabidopsis_thaliana.TAIR10.36.cdna.all.fa.index**.
4. If desired adjust the k-mer length (See Kallisto paper for recommendations); we will use the default.
5. Click 'Launch Analyses' to start the job. Click on the Analyses button to monitor the job and results.

Output/Results

Output	Description	Example
Kallisto Index	This is the index file Kallisto will map RNA-Seq reads to.	Example Kallisto index

---

Next Steps:

With the index, we will now use the Kallisto Quant App to do a pesudoalignment of the RNA-Seq reads.

More help and additional information

Search for an answer: CyVerse Learning Center or CyVerse Wiki

Post your question to the user forum: Ask CyVerse

Fix or improve this documentation

- On Github: Repo link
- Send feedback: Tutorials@CyVerse.org

Learning Center Home

Learning Center Home

Quantify Reads with Kallisto

Description:

Kallisto uses a 'hash-based' pseudo alignment to deliver extremely fast matching of RNA-Seq reads against the transcriptome index. Each Kallisto job in this tutorial will take only several minutes to complete.

Input Data:

Input	Description	Example
Kallisto Index	Indexed transcriptome	Example Kallisto index
RNA-Seq Reads	Cleaned fastq files	Example fastq files

Quantify RNA-Seq reads with Kallisto

A Kallisto analyses must be run for each mapping of RNA-Seq reads to the index. In this tutorial, we have 36 fastq files (18 pairs), so you will need to launch 18 Kallisto analyses. It is sufficient here to launch a single Kallisto job to examine the input and then use the completed results (which are small files) for Sleuth analyses.

1. If necessary, login to the CyVerse Discovery Environment
2. Open the Kallisto-0.42.3-Quant-PE App
3. Name your analysis, and if desired enter comments. In the App's 'Input' step under 'Index file' browse to and select the Kallisto index generated in the previous tutorial section. In the output directory, enter the name for the output directory that will be created. For this tutorial, name your output directory **pair01_wt_mock_r1** (This is for our first pair of WT reads, mock treatment, replicate 1).
4. Under 'Input Read1&Reaad2 fastq files' enter a single pair (foward/reverse) of reads. In this tutorial click 'Add' to select the following files located in *Community Data* > *cyverse_training* > *tutorials* > *kallisto* > *00_input_fastq_trimmed*: - SRR1761506_R1_001.fastq.gz_fp.trimmed.fastq.gz - SRR1761506_R2_001.fastq.gz_fp.trimmed.fastq.gz
5. Click 'Launch Analyses' to launch the job and monitor its progress.

Output/Results

Each Kallisto job will generate 3 files

Output	Description	Example
abundances.h5	HDF5 binary file containing run info, abundance estimates, bootstrap estimates, and transcript length information length. This file can be read in by Sleuth	example abundance.h5
abundances.tsv	plaintext file of the abundance estimates. It does not contains bootstrap estimates. When plaintext mode is selected; output plaintext abundance estimates. Alternatively, kallisto h5dump will output an HDF5 file to plaintext. The first line contains a header for each column, including estimated counts, TPM, effective length.	example abundance.tsv
run_info.json	json file containing information about the run	example json

---

Summary

Kallisto quantifies RNA-Seq reads against an indexed transcriptome and generates a folder of results for each set of RNA-Seq reads.

Next Steps:

Sleuth will be used to examine the Kallisto results in R Studio.

More help and additional information

Search for an answer: CyVerse Learning Center or CyVerse Wiki

Post your question to the user forum: Ask CyVerse

Fix or improve this documentation

- On Github: Repo link
- Send feedback: Tutorials@CyVerse.org

Learning Center Home

Learning Center Home

Analyze Kallisto Results with Sleuth

Description:

Sleuth is a program for analysis of RNA-Seq experiments for which transcript abundances have been quantified with kallisto. In this tutorial, we will use R Studio being served from an Atmosphere instance.

Input Data:

Input	Description	Example
Kallisto results folder(s)	Outputs from a Kallisto quantification	Example directory of Kallisto results

Import Kallisto results into R and analyze with Sleuth

Tip: If you are not familiar with Atmosphere and transfering data into an instance see the Atmosphere guide

We will import the Kallisto results into an RStudio session being run from an Atmosphere image. Then we will follow a R script based on the Sleuth Walkthoughs

1. If necessary, launch the CyVerse Workshop Training Image; a 'Small1' instance size should be sufficient.
2. When the instance becomes available, open a web browser and navigate to your R studio session. This will be located at URL based on your image ip: "http ://your.image.ip:8787";
3. Login to Atmosphere via Webshell or SSH; initialize iCommands, and use the following commands to import the test data and R scripts to your home directory:

```
iget -rPV /iplant/home/shared/cyverse_training/tutorials/kallisto/03_output_
↪kallisto_results
iget -rPV /iplant/home/shared/cyverse_training/tutorials/kallisto/04_sleuth_R
```

4. Load the sample_kallisto_script.R into your R Studio session. Use the Kallisto_demo_tsv file for the step where you describe experimental design metadata.

```
# This tutorial is based on the one by the Pachterlab here:
# https://pachterlab.github.io/sleuth_walkthroughs/boj/analysis.html

# Load sleuth
suppressMessages({library("sleuth")})

# modify the following line so the location of your kallisto results (each
# in their own folder) are specified. Hint: don't use slashes to
# separate your directory names, use a comma-separated list of
# quoted directory names.
sample_id <- dir(file.path("./03_output_kallisto_results/"))

# Running the next line should give you the names of your samples
# assuming your kallisto result folders have been appropriately named
sample_id

#specify the full path relative to this script
kal_dirs <- file.path("./03_output_kallisto_results",sample_id)

# should show full path to results containing the kallisto outputs
kal_dirs

# create a table for the experimental design
s2c <- read.table(file.path("kallisto_demo.tsv"),
                  header = TRUE,
                  stringsAsFactors = FALSE,
                  sep = "\t")

# Check the table
s2c

# Add the directories as a final column called path
s2c <- dplyr::mutate(s2c, path = kal_dirs)

# check the table

print(s2c)


# install cowplot and load for better plots
install.packages("cowplot")
library(cowplot)


# prepare to associate transcripts to genes - you will have to
# do a bit of checking to ensure you have the proper names
# and attributes for your genome of interest
# this step is not essential
# see some instructions on getting your gene names here:
# http://www.ensembl.info/blog/2015/06/01/biomart-or-how-to-access-the-ensembl-data-
↪from-r/

#load biomaRt
library(biomaRt)

#load arabidopsis genes
```

```
mart <- biomaRt::useMart(biomart = "plants_mart",
                         dataset = "athaliana_eg_gene",
                         host = "plants.ensembl.org")
#get target transcripts

ttg <- biomaRt::getBM(
  attributes = c("ensembl_transcript_id", "transcript_version",
                 "ensembl_gene_id", "external_gene_name", "description",
                 "transcript_biotype"),
                  mart = mart)

# do some renaming
ttg <- dplyr::rename(ttg, target_id = ensembl_transcript_id,
                     ens_gene = ensembl_gene_id, ext_gene = external_gene_name)

# Check your listing of transcripts
head(ttg)

# Create the "Sleuth Object" a data structure that contains our results

so <- sleuth_prep(s2c,
                  target_mapping = ttg,
                  aggregation_column = 'ens_gene',
                  extra_bootstrap_summary = TRUE)

# Check the structure of the data with a PCA plot
# PCA by treatment shows clear difference between ABA and mock
# good! this was SRA data - so glad this worked!
# You will get some expected warnings

plot_pca(so, color_by = 'treatment_s')

# you can also add labels to the plot
plot_pca(so, color_by = 'treatment_s',
         text_labels = TRUE)

# We can also see genes involved in the the 1st PC by looking
# at the loadings (primary genes whose linear combinations define
# the principal components)

plot_loadings(so, pc_input = 1)

# See more about the gene most influential in this dataset
# https://www.arabidopsis.org/servlets/TairObject?type=gene&name=At2g34420.1


# Testing for differential genes
# Create the full model (i.e. a model that contains all covariates)
# then create an additional model with respect to one of the covariates
# finally compare both models to identify the differences

so <- sleuth_fit(so, ~ genotype_variation_s + treatment_s, 'full')
so <- sleuth_fit(so, ~genotype_variation_s, 'treatment_s')
# likelihood ratio test
so <- sleuth_lrt(so, 'treatment_s', 'full')

#get the full results
full_results <- sleuth_results(so, 'treatment_s:full', 'lrt',
```

```
                                show_all = FALSE)

# filter out the significant genes according to a set qvalue

sleuth_significant <- dplyr::filter(full_results, qval <= 0.05)

#view the first 20 genes in this list
head(sleuth_significant, 20)


# we can write the entire gene table out
write.csv(full_results,
          file = "sleuth_results.csv" )

# we can also use Rshiny to do some interactive visualizations

sleuth_live(so)
```

Summary

Together, Kallisto and Sleuth are quick, powerful ways to analyze RNA-Seq data.

More help and additional information

Search for an answer: CyVerse Learning Center or CyVerse Wiki

Post your question to the user forum: Ask CyVerse

Fix or improve this documentation

- On Github: Repo link
- Send feedback: Tutorials@CyVerse.org

 Learning Center Home

CHAPTER 2

Prerequisites

Downloads, access, and services

In order to complete this tutorial you will need access to the following services/software

Prerequisite	Preparation/Notes	Link/Download
CyVerse account	You will need a CyVerse account to complete this exercise	Register
Atmosphere access (optional)	This tutorial will use R studio in Atmosphere; if desired you can complete these sections by installing the Sleuth tools on your own R instance	Request Access

Platform(s)

We will use the following CyVerse platform(s):

Platform	Interface	Link	Platform Documentation	Quick Start
Data Store	GUI/Command line	Data Store	Data Store Manual	Guide
Discovery Environment	Web/Point-and-click	Discovery Environment	DE Manual	Guide
Atmosphere	Command line (ssh) and/or Desktop (VNC)	Atmosphere	Atmosphere Manual	Guide

Application(s) used

Discovery Environment App(s):

App name	Version	Description	App link	Notes/other links
Kallisto-0.42.3-INDEX	0.42.3	Kallisto Index Builder	DE App link	Original documentation
Kallisto-0.42.3-Quant-PE	0.43.3	Kallisto Quantification	DE App link	Original documentation

Atmosphere Image(s):

Image name	Version	Description	Link	Notes/other links
CyVerse Training Workshop	1.1.2	Image for use at CyVerse Training Workshops	Image	This image has Kallisto and Sleuth installed. Once started, an R Studio Server will be available at your image ip address, port 8787 (e.g. image.ip.address:8787)

Input and example data

In order to complete this tutorial you will need to have the following inputs prepared

Input File(s)	Format	Preparation/Notes	Example Data
RNA-Seq reads	Fastq (may also be compressed, e.g. fastq.gz)	These reads should have been cleaned by upstream tools such as Trimmomatic	Example fastq files
Refference transcriptome	fasta	Transcriptome for your organism of interest	Example transcriptome

---

Fix or improve this documentation

- On Github: Repo link
- Send feedback: Tutorials@CyVerse.org


Learning Center Home