DNAnexus Command Line Interface

Stephen Hartley, PhD

2019-12-02

Contents

Chapter 1

Setup

First we have to load the DNAnexus command line interface utility. Both CCAD and Biowulf have it installed as a module.

```
#CCAD:
module load dx
#BIOWULF:
#module load DNAnexus
```

Next we have to login, select a project, and

```
#Log in (only have to do this the first time)
#dx login

#See what projects you have access to:
dx find projects

#Select the project you want from the above list:
dx select project-FbvPXyQ0pgP1PQbkJG1vfQpz
```

```
## project-FbvPXyQ0pgP1PQbkJG1vfQpz : Steve@NCI (ADMINISTER)
## project-FXfKVQj99yPxP6YqB7ykfBZ8 : AzureWest_Project (CONTRIBUTE)
## project-Ff54zy09462pQB99BpVZv7kG : steveTestProject (ADMINISTER)
## project-FYf1F7j9BppKqvGPK5GvqBzz : Genetic Susceptibility to Subsequent Neoplasms (CONTRIBUTE)
## Selected project project-FbvPXyQ0pgP1PQbkJG1vfQpz
```

Once dx is loaded and a project is selected, you can use many standard linux commands on the project by prefacing the command with "dx"

```
dx ls
dx cd test
dx cd ..

dx mkdir test2
dx rmdir test2
```

```
## test/
## findNi
```

For a full list of available commands:

```
dx help all
```

```
## dx all: All commands
##
## Commands:
```

```
##
##   add               Add one or more items to a list
##   add developers    Add developers for an app
##   add member        Grant a user membership to an org
##   add stage         Add a stage to a workflow
##   add users         Add authorized users for an app
##   add_types         Add types to a data object
##   api               Call an API method
##   build             Upload and build a new applet/app, or a workflow
##   build_asset       Build an asset bundle
##   cat               Print file(s) to stdout
##   cd                Change the current working directory
##   clearenv          Clears all environment variables set by dx
##   close             Close data object(s)
##   cp                Copy objects and/or folders between different projects
##   describe          Describe a remote object
##   download          Download file(s)
##   env               Print all environment variables in use
##   exit              Exit out of the interactive shell
##   find              Search functionality over various DNAnexus entities
##   find analyses     List analyses in the current project
##   find apps         List available apps
##   find data         List data objects in the current project
##   find executions   List executions (jobs and analyses) in the current project
##   find globalworkflowsList available global workflows
##   find jobs         List jobs in the current project
##   find org          List entities within a specific org.          "dx find org
##                     members" lists members in the
##                     specified org          "dx find org
##                     projects" lists projects billed to the
##                     specified org          "dx find org
##                     apps" lists apps billed to the
##                     specified org  Please execute "dx find
##                     org -h" for more information.
##   find org apps     List apps billed to the specified org
##   find org members  List members in the specified org
##   find org projects List projects billed to the specified org
##   find orgs         List orgs
##   find projects     List projects
##   generate_batch_inputsGenerate a batch plan (one or more TSV files) for batch
##                     execution
##   get               Download records, apps, applets, workflows, and files
##   get_details       Get details of a data object
##   head              Print part of a file
##   help              Display help messages and dx commands by category
##   install           Install an app
##   invite            Invite another user to a project or make it public
##   list              Print the members of a list
##   list database     List entities associated with a specific database. For
##                     example,          "dx list database
##                     files" lists database files associated
##                     with a specific database.
##                     Please execute "dx list database -h"
##                     for more information.
##   list database filesList files associated with a specific database
##   list developers   List developers for an app
##   list stages       List the stages in a workflow
##   list users        List authorized users for an app
##   login             Log in (interactively or with an existing API token)
##   logout            Log out and remove credentials
##   ls                List folders and/or objects in a folder
##   make_download_url Create a file download link for sharing
##   mkdir             Create a new folder
##   mv                Move or rename objects and/or folders inside a project
##   new               Create a new project or data object
##   new org           Create new org
##   new project       Create a new project
##   new record        Create a new record
##   new user          Create a new user account
##   new workflow      Create a new workflow
##   publish           Publish an app or a global workflow
##   pwd               Print current working directory
##   remove            Remove one or more items to a list
##   remove developers Remove developers for an app
##   remove member     Revoke the org membership of a user
##   remove stage      Remove a stage from a workflow
##   remove users      Remove authorized users for an app
##   remove_types      Remove types from a data object
##   rename            Rename a project or data object
##   rm                Remove data objects and folders
##   rmdir             Remove a folder
##   rmproject         Delete a project
##   run               Run an applet, app, or workflow
##   select            List and select a project to switch to
##   set_details       Set details on a data object
##   set_properties    Set properties of a project, data object, or execution
##   set_visibility    Set visibility on a data object
##   setenv            Sets environment variables for the session
##   ssh               Connect to a running job via SSH
##   ssh_config        Configure SSH keys for your DNAnexus account
##   tag               Tag a project, data object, or execution
##   terminate         Terminate jobs or analyses
##   tree              List folders and objects in a tree
##   uninstall         Uninstall an app
##   uninvite          Revoke others' permissions on a project you administer
##   unset_properties  Unset properties of a project, data object, or execution
##   untag             Untag a project, data object, or execution
##   update            Update certain types of metadata
##   update member     Update the membership of a user in an org
##   update org        Update information about an org
##   update project    Updates a specified project with the specified options
##   update stage      Update the metadata for a stage in a workflow
##   update workflow   Update the metadata for a workflow
##   upgrade           Upgrade dx-toolkit (the DNAnexus SDK and this program)
##   upload            Upload file(s) or directory
##   wait              Wait for data object(s) to close or job(s) to finish
##   watch             Watch logs of a job and its subjobs
##   whoami            Print the username of the current user
```

Chapter 2

The DNANexus Web Interface:

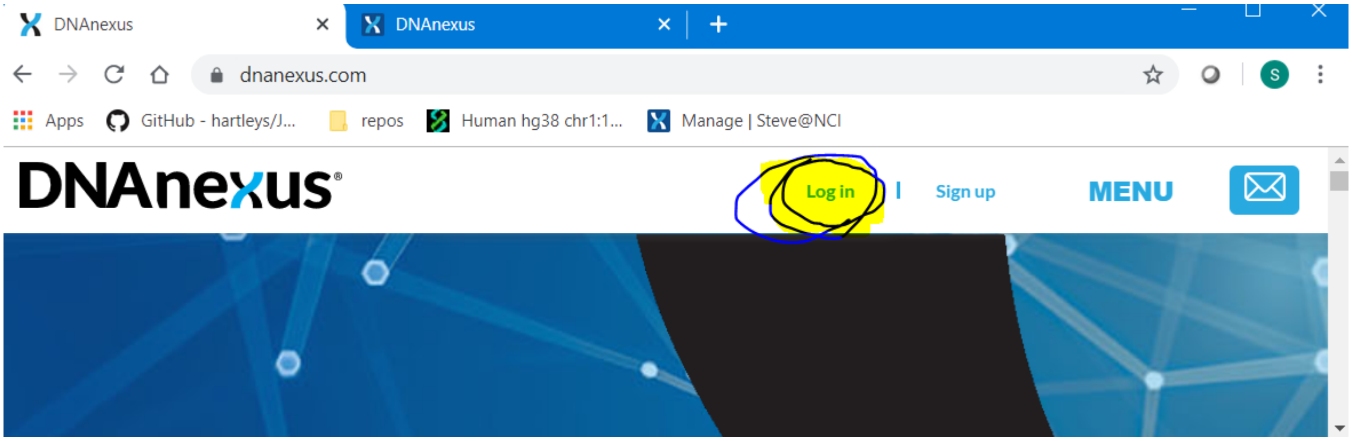

Figure 2.1: A figure caption.

Chapter 3

A Simple Example:

We are going to do a very simple example in which we upload some data, run a program on it, and then download the result.

Say we want to extract all the dictionary words that begin with “ni”.

Chapter 4

The App Builder Wizard:

The first step is to create a new App. You can generate the files from a template, write them yourself, or you can use the dx app creation wizard. For your first project, I recommend the last option.

Since the wizard is interactive, I have already performed this step and the example directory contains a copy of the output:

```
cp -R ../examples/findNi_original ./ex/findNi
```

Chapter 5

Modifying the App:

The app produced by the wizard is very minimal, it doesn't actually DO anything.

```
ls ./ex/findNi/
```

```
## dxapp.json
## findNi_original
## Readme.developer.md
## Readme.md
## resources
## src
## test
```

```
ls ./ex/findNi/src
```

```
## findNi.sh
```

The main files that you will need to modify are the dxapp.json file:

```
cat ./ex/findNi/dxapp.json
```

```
## {
##   "name": "findNi",
##   "title": "Find Ni",
##   "summary": "Finds words that begin with ni",
##   "dxapi": "1.0.0",
##   "version": "0.0.1",
##   "inputSpec": [
##     {
##       "name": "wordsFile",
##       "label": "File of all words",
##       "class": "file",
##       "optional": false,
##       "patterns": [
##         "*"
##       ],
##       "help": ""
##     }
##   ],
##   "outputSpec": [
##     {
##       "name": "outfile",
##       "label": "Output file of words",
##       "class": "file",
##       "patterns": [
##         "*"
##       ],
##       "help": ""
##     }
##   ],
##   "runSpec": {
##     "timeoutPolicy": {
##       "*": {
##         "hours": 48
##       }
##     },
##     "interpreter": "bash",
##     "release": "16.04",
##     "distribution": "Ubuntu",
##     "file": "src/findNi.sh",
##     "execDepends": [
##         {"name": "openjdk-8-jre-headless",
##          "package_manager": "apt"},
##
##         {"name": "tabix",
##          "package_manager": "apt"}
##     ]
##   },
##   "regionalOptions": {
##     "aws:us-east-1": {
##       "systemRequirements": {
##         "*": {
##           "instanceType": "mem1_ssd1_x4"
##         }
##       }
##     }
##   }
## }
```

and also the bash script in the src directory:

```
cat ./ex/findNi/src/findNi.sh
```

```
## #!/bin/bash
## # helloWorldTestApp 0.0.2
##
## main() {
##
##     #First, let's print out the input file, to make sure the inputs
##     # got set properly:
##     echo "Value of wordsFile: '$wordsFile'"
##     echo "Value of wordsFile_name: $wordsFile_name"
##
##     #Next let's download the words file from the cloud and into
##     # our virtual machine.
##     dx download "$wordsFile" -o $wordsFile_name
##
##     #This part is the program itself. It takes the input and grabs
##     # all the words that begin with Ni
##     cat $wordsFile_name | grep "^ni" > words.that.begin.with.ni.txt
##
##     #This part uploads the resultant file to the cloud
##     # If you don't do this, then all the files you made will be
##     # deleted when the virtual machine gets shut down.
##     #Note: dx upload --brief returns a file code that
##     # the next tool needs in order to register the output.
##     outfile_dx_code=$(dx upload words.that.begin.with.ni.txt --brief)
##
##     #Finally: this part registers the output and connects it to this job
##     # That way the output will be accessable from the job screen.
##     dx-jobutil-add-output outfile "$outfile_dx_code" --class=file
##
##
## }
```

I have already generated new versions of these files.

For the json file, I just added some dependencies just to show how one would do that. In this case I told the VM to install tabix and openJDK. Note: I don't actually need them in this example.

```
cp ../examples/files/dxapp.v2.json ./ex/findNi/dxapp.json
cat ./ex/findNi/dxapp.json
```

```
## {
##   "name": "findNi",
##   "title": "Find Ni",
##   "summary": "Finds words that begin with ni",
##   "dxapi": "1.0.0",
##   "version": "0.0.1",
##   "inputSpec": [
##     {
##       "name": "wordsFile",
##       "label": "File of all words",
##       "class": "file",
##       "optional": false,
##       "patterns": [
##         "*"
##       ],
##       "help": ""
##     }
##   ],
##   "outputSpec": [
##     {
##       "name": "outfile",
##       "label": "Output file of words",
##       "class": "file",
##       "patterns": [
##         "*"
##       ],
##       "help": ""
##     }
##   ],
##   "runSpec": {
##     "timeoutPolicy": {
##       "*": {
##         "hours": 48
##       }
##     },
##     "interpreter": "bash",
##     "release": "16.04",
##     "distribution": "Ubuntu",
##     "file": "src/findNi.sh",
##     "execDepends": [
##         {"name": "openjdk-8-jre-headless",
##          "package_manager": "apt"},
##
##         {"name": "tabix",
##          "package_manager": "apt"}
##     ]
##   },
##   "regionalOptions": {
##     "aws:us-east-1": {
##       "systemRequirements": {
##         "*": {
##           "instanceType": "mem1_ssd1_x4"
##         }
##       }
##     }
##   }
## }
```

Next we write the script itself. The main function gets executed once the virtual machine is spun up and everything is installed and prepped. Note that it sets a bunch of environment variables that link to the input files.

The program below runs through the provided words file and returns a file containing all the words that begin with “ni”.

```
cp ../examples/files/findNi.v2.bash ./ex/findNi/src/findNi.sh
cat ./ex/findNi/src/findNi.sh
```

```
## #!/bin/bash
## # helloWorldTestApp 0.0.2
##
## main() {
##
##     #First, let's print out the input file, to make sure the inputs
##     # got set properly:
##     echo "Value of wordsFile: '$wordsFile'"
##     echo "Value of wordsFile_name: $wordsFile_name"
##
##     #Next let's download the words file from the cloud and into
##     # our virtual machine.
##     dx download "$wordsFile" -o $wordsFile_name
##
##     #This part is the program itself. It takes the input and grabs
##     # all the words that begin with Ni
##     cat $wordsFile_name | grep "^ni" > words.that.begin.with.ni.txt
##
##     #This part uploads the resultant file to the cloud
##     # If you don't do this, then all the files you made will be
##     # deleted when the virtual machine gets shut down.
##     #Note: dx upload --brief returns a file code that
##     # the next tool needs in order to register the output.
##     outfile_dx_code=$(dx upload words.that.begin.with.ni.txt --brief)
##
##     #Finally: this part registers the output and connects it to this job
##     # That way the output will be accessable from the job screen.
##     dx-jobutil-add-output outfile "$outfile_dx_code" --class=file
##
##
## }
```

Chapter 6

Building the App:

Now we have to “build” the app, which packages up the app and uploads it to your current project

```
cd ./ex/
dx build -f findNi
```

```
## WARNING:dxpy:name "findNi" should be all lowercase
## INFO:dxpy:Deleting applet(s) applet-FgXgj5Q0pgPF3X4v4ZQv0j80
## {"id": "applet-FgXgjFj0pgP3yGXV15Q0bBY9"}
```

Chapter 7

Uploading Input Files

Before we can run our new app, first we need to upload an input file. For this we use the upload command:

```
dx upload ../examples/indata/linux.words.txt --brief --path /test/linux.words.txt
```

```
## file-FgXgjG00pgPBV2bJ8Z7ZzY7g
```

Chapter 8

Running the software

Before we can run our new app, first we need to upload an input file. For this we use the upload command:

```
dx run findNi -i wordsFile=/test/linux.words.txt --destination /test/
```

Chapter 9

Downloading results

We can now download the results using the dx download command:

```
dx download /test/words.that.begin.with.ni.txt
#Let's see the first 50 words:
cat words.that.begin.with.ni.txt | head -n50
```

```
## Error: path "/mnt/nfs/gigantor/ifs/Shared/hartleys/docs/DnaNexusTutorial/build
## /words.that.begin.with.ni.txt" already exists but -f/--overwrite was not set
## ni
## niacin
## niacinamide
## niacins
## niagara
## niagra
## niais
## niaiserie
## nialamide
## niata
## nib
## nibbana
## nibbed
## nibber
## nibbing
## nibble
## nibbled
## nibbler
## nibblers
## nibbles
## nibbling
## nibblingly
## nibby
## nibby-jibby
## nibelung
## niblic
## niblick
## niblicks
## niblike
## nibong
## nibs
## nibsome
## nibung
## nicad
## nicads
## nicaean
## nicaragua
## nicaraguan
## nicaraguans
## niccolic
## niccoliferous
## niccolite
## niccolo
## niccolous
## nice
## niceish
## niceling
## nicely
## nicene
## nice-Nellie
```

Chapter 10

Other useful commands:

You can use cat to quickly preview files:

```
dx cat /test/words.that.begin.with.ni.txt | head -n10
```

```
## ni
## niacin
## niacinamide
## niacins
## niagara
## niagra
## niais
## niaiserie
## nialamide
## niata
```

```
set -x
dx describe /test/words.that.begin.with.ni.txt
```

```
## + dx describe /test/words.that.begin.with.ni.txt
## Result 1:
## ID                    file-FgXgbfj0gzgbqx5KJ97V6J7j
## Class                 file
## Project               project-FbvPXyQ0pgP1PQbkJG1vfQpz
## Folder                /test
## Name                  words.that.begin.with.ni.txt
## State                 closed
## Visibility            visible
## Types                 -
## Properties            -
## Tags                  -
## Outgoing links        -
## Created               Mon Dec  2 15:58:03 2019
## Created by            hartleys
##  via the job          job-FgXgbK80pgPJJ54XF2qK4zBp
## Last modified         Mon Dec  2 15:58:05 2019
## archivalState         "live"
## Media type            text/plain
## Size                  10.33 KB
```

```
dx watch <jobid>
```

Chapter 11

Legal

This document and all relevant documentation is “United States Government Work” under he terms of the United States Copyright Act. It was written as part of the authors’ official duties for the United States Government and thus vArmyKnife cannot be copyrighted. This software is freely available to the public for use without a copyright notice. Restrictions cannot be placed on its present or future use.

Although all reasonable efforts have been taken to ensure the accuracy and reliability of the software, documentation, and data, the National Human Genome Research Institute (NHGRI), the National Cancer Institute (NCI) and the U.S. Government does not and cannot warrant the performance or results that may be obtained by using this software or data. NHGRI, NCI and the U.S. Government disclaims all warranties as to performance, merchantability or fitness for any particular purpose.

In any work or product derived from this material, proper attribution of the authors as the source of the software or data may be made, using “NCI Division of Cancer Epidemiology and Genetics, Human Genetics Program” as the citation.