DNAnexus Development Environment
Bioinformaticists & Developers
Today Agenda

- Introduction dx-toolkit
- Supported languages - bash, python, docker
- Supported resources
- App/Applet building experience
  Peter FitzGerald (bash)
  Carl McIntosh (Python)
  Skyler Kuhn (Docker)
- Open Discussion
DNAnexus provides a simplified, structured and managed access to Amazon Web Services (AWS) and Microsoft's (Azure)
DNAnexus Applet System Requirements

Default: mem1_ssd1_x4

Common AWS instance types:

<table>
<thead>
<tr>
<th>Name</th>
<th>Memory_GB</th>
<th>Storage_GB</th>
<th>CPU_Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem1_ssd1_x2</td>
<td>3.8</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>mem1_ssd1_x4</td>
<td>7.5</td>
<td>80</td>
<td>4</td>
</tr>
<tr>
<td>mem1_ssd1_x8</td>
<td>15.0</td>
<td>160</td>
<td>8</td>
</tr>
<tr>
<td>mem1_ssd1_x16</td>
<td>30.0</td>
<td>320</td>
<td>16</td>
</tr>
<tr>
<td>mem1_ssd1_x32</td>
<td>60.0</td>
<td>640</td>
<td>32</td>
</tr>
<tr>
<td>mem2_ssd1_x2</td>
<td>7.5</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>mem2_ssd1_x4</td>
<td>15.0</td>
<td>80</td>
<td>4</td>
</tr>
<tr>
<td>mem2_ssd1_x8</td>
<td>30.0</td>
<td>160</td>
<td>8</td>
</tr>
<tr>
<td>mem3_ssd1_x2</td>
<td>15.0</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>mem3_ssd1_x4</td>
<td>30.5</td>
<td>80</td>
<td>4</td>
</tr>
<tr>
<td>mem3_ssd1_x8</td>
<td>61.0</td>
<td>160</td>
<td>8</td>
</tr>
<tr>
<td>mem3_ssd1_x16</td>
<td>122.0</td>
<td>320</td>
<td>16</td>
</tr>
<tr>
<td>mem3_ssd1_x32</td>
<td>244.0</td>
<td>640</td>
<td>32</td>
</tr>
<tr>
<td>mem1_ssd2_x2</td>
<td>3.8</td>
<td>160</td>
<td>2</td>
</tr>
<tr>
<td>mem1_ssd2_x4</td>
<td>7.5</td>
<td>320</td>
<td>4</td>
</tr>
<tr>
<td>mem1_ssd2_x8</td>
<td>15</td>
<td>640</td>
<td>8</td>
</tr>
<tr>
<td>mem1_ssd2_x16</td>
<td>30</td>
<td>1280</td>
<td>16</td>
</tr>
<tr>
<td>mem1_ssd2_x36</td>
<td>60</td>
<td>2880</td>
<td>36</td>
</tr>
</tbody>
</table>

- **Memory:**
  - AWS: 3.8 - 244 GB
  - Azure: 3.9 - 448 GB

- **Storage:**
  - AWS: 32 - 2,880 GB
  - Azure: 32 - 1,024 GB

- **Harddrive:**
  - Standard Drive
  - Solid-State Drive

- **Number of Cores:**
  - AWS: 2-36
  - Azure: 2-32

Common Azure instance types:

<table>
<thead>
<tr>
<th>Name</th>
<th>Memory_GB</th>
<th>Storage_GB</th>
<th>CPU_Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>azure:mem1_ssd1_x2</td>
<td>3.9</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>azure:mem1_ssd1_x4</td>
<td>7.8</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>azure:mem1_ssd1_x8</td>
<td>15.7</td>
<td>128</td>
<td>8</td>
</tr>
<tr>
<td>azure:mem1_ssd1_x16</td>
<td>31.4</td>
<td>256</td>
<td>16</td>
</tr>
<tr>
<td>azure:mem2_ssd1_x1</td>
<td>3.5</td>
<td>128</td>
<td>1</td>
</tr>
<tr>
<td>azure:mem2_ssd1_x2</td>
<td>7.0</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>azure:mem2_ssd1_x4</td>
<td>14.0</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>azure:mem2_ssd1_x8</td>
<td>28.0</td>
<td>256</td>
<td>8</td>
</tr>
<tr>
<td>azure:mem2_ssd1_x16</td>
<td>56.0</td>
<td>512</td>
<td>16</td>
</tr>
<tr>
<td>azure:mem3_ssd1_x2</td>
<td>14.0</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>azure:mem3_ssd1_x4</td>
<td>28.0</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>azure:mem3_ssd1_x8</td>
<td>56.0</td>
<td>256</td>
<td>8</td>
</tr>
<tr>
<td>azure:mem3_ssd1_x16</td>
<td>112.0</td>
<td>512</td>
<td>16</td>
</tr>
<tr>
<td>azure:mem4_ssd1_x2</td>
<td>140.0</td>
<td>640</td>
<td>20</td>
</tr>
<tr>
<td>azure:mem4_ssd1_x4</td>
<td>28.0</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>azure:mem4_ssd1_x8</td>
<td>56.0</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>azure:mem4_ssd1_x16</td>
<td>112.0</td>
<td>256</td>
<td>8</td>
</tr>
<tr>
<td>azure:mem4_ssd1_x32</td>
<td>224</td>
<td>512</td>
<td>16</td>
</tr>
<tr>
<td>azure:mem4_ssd1_x32</td>
<td>448</td>
<td>1024</td>
<td>32</td>
</tr>
</tbody>
</table>

https://wiki.dnanexus.com/API-Specification-v1.0.0/Instance-Types
The DNAnexus SDK (dx-toolkit) helps users utilize the DNAnexus platform to its full potential. It provides command-line tools to run Apps/applets from a remote command-line/script. Additionally, it provides the environment for App/applet development.

- The dx-toolkit is installed on Helix/Biowulf in the module system and can be run with the following command (Note: *can only be run on bioiwdulf interactive nodes*):
  - module load DNAnexus
Development Environment

The main controlling module of an App/applet can be written in either of the following:

- Bash
- Python
- Docker

Helpful Web Pages

DNAnexus External Resources

- Package Managers
  - Advanced Packaging Tool (APT)
    - Libraries, Samtools, Bedtools, etc.
  - Python Package Index (PyPI)
  - Comprehensive Perl Archive Network (Perl)
  - Ruby Gems (gem)
  - Comprehensive R Archive Network (CRAN)

DNAnexus How-to https://wiki.dnanexus.com/Execution-Environment-Reference
Applet Design Process

• Sketch out Workflow (*OMNIGraffle for example*)

• Required Resources (Asset Bundle vs Applet Resource) -> Detailed web link

• Applet Model (standard, parallelize, SPG) -> Detailed web link

• Input Elements -> Detailed web link

• Output
  • Properties and Tags
  • Directory Structure

• Final Touches
  • Documentation (MacDown) -> Detailed web link
  • Script.sh
  • Versions
DNAnexus dx-app-wizard

- From *Terminal* on Biowulf/Helix
  - module load DNAnexus
  - dx-app-wizard

- From *Terminal* on Local Computer
  - dx-app-wizard
dx-app-wizard Applet Parameters

• Timeout policy[48h] (m | h | d)

• Applet Programming language: (Python | bash), but supports other languages

• Applet Access to Internet [N]:

• Applet Access to Parent Project: [N]

• Ubuntu 14.04

• Compute Nodes
Input Specification

You will now be prompted for each input parameter to your app. Each parameter should have a unique name that uses only the underscore "_" and alphanumeric characters, and does not start with a number.

1st input name (<ENTER> to finish): parameter

Label (optional human-readable name) []:

Your input parameter must be of one of the following classes:

applet       array:file    array:record    file       int
array:applet  array:float  array:string    float      record
array:boolean array:int    boolean        hash       string

Output Specification

The same
**DNAnexus dx-app-wizard**

Basic  Parallelized  Scatter-Process-Gather

```
dx-app-wizard --template  (basic | parallelized | scatter-process-gather)
```

Cloud Workstation

DNAnexus features a Cloud Workstation App that sets up a cloud workstation as an interactive computer node.

**What are typical use cases for this app?**

This app can be used as a workstation inside of the DNAnexus cloud platform. By running the app with `--ssh` or `--allow-ssh`, users can login to a machine inside of the DNAnexus cloud platform. From there, users can upload/download data to/from the project in which the app is run, perform data analysis, and install additional packages from sources such as apt, cran, pip, github, etc.

It’s good for debugging, since you can do so interactively on the node as its running.

```bash
dx run app-cloud_workstation --ssh
unset DX_WORKSPACE_ID
dx cd $DX_PROJECT_CONTEXT_ID:
dx download file.txt ## download to workstation from parent project
dx upload file.txt ## upload to parent project from workstation
dx terminate $DX_JOB_ID #terminate the session
```

Genome Analysis Unit (GAU)
DNAnexus Applet Development

Custom Work Flows developed by
Carl McIntosh and Peter FitzGerald (GAU)
Using DNAnexus to make the ADAP program readily available to a naive audience. The program was originally written, many years ago, and has had several interface iterations (Web App, Standalone Mac/PC program). The program takes a DNA fasta file and recodes the sequence using alternate AA codons, to generate a new sequence **As Different As Possible** from the original, yet codes the same protein. This approach is useful in the overexpression of proteins.

A Collaboration with Christopher Buck & Diana Pastrana (Laboratory of Cellular Oncology, NCI/CCR)

The simplest of applets - simple bash script (5 lines!), and a single binary from C code compiled on biowulf

#!/bin/bash
# The following line causes bash to exit at any point if there is any error
# and to output each line as it is executed -- useful for debugging
set -e -x

# Inputs
dx download "$input" -o input.fasta —no-progress
# make a directory for the output
mkdir -p out/results

diana -f input.fasta -c /usr/lib/diana.codes > out/results/adap.log

dx-upload-all-outputs
IGV Session Maker

- Designed to be a helper applet that allows easy visualization of large files (bam, vcf, big-wig) by a locally running copy of IGV, without the need to download the entire files. It can be run standalone or incorporated into a workflow. By launching from a custom built HTML page it provides a stable record of what is represented in the view.

- Applet consists of a single bash script, and uses dx commands and variables.

---

IGV_Session_Maker generated output
Mon Apr 8 21:00:54 UTC 2019
Ecoli Genome

This page contains a link to an IGV session file. This file will allow the specified data to be streamed to IGV without the need to explicitly download the data. IMPORTANT - IGV must be running on your local machine before you click the link.

Description: This session contain the file type sample 04/09/2019

The following files are included in the Session file:

- /PAUSING/1096-no-chase_S1_L001_R1_001_aligner/1096-no-chase_S1_L001_R1_001_m14_M30_uniq.bam
- /PAUSING/TSS/1096-no-chase_S1_L001_R1_001_m14_M30_finder/1096-no-chase_S1_L001_R1_001_m14_M30_50_S1_100_MG1655_median.bw
- /PAUSING/TSS/1096-no-chase_S1_L001_R1_001_m14_M30_finder/1096-no-chase_S1_L001_R1_001_m14_M30_50_S1_100_MG1655_median_peaks.bw

Click on the button below:
Launch IGV with relevant BAM files

This link will work for 100 days from its date of generation.
Pausing Peak Tools
Pausing Peak Tool

**Biological system:** Various microbial organisms
Primary goal was to identify RNA pausing sites, from netSeq data, and correlate with:
- genome position
- gene expression
- transcription start sites (TSS)
- specific sequence motifs
- protected read length
Additionally, we needed the ability to compare the effect of different gene deletions.
Pausing Peak Tool

Pausing_Peak_Aligner

- Remove sequencing primers/adapters using cutadapt (java)

- Use molecular bar code to identify and remove duplicate molecules with BBmap (bin)

- Remove molecular bar code with cutadapt (java)

- Align vs genome with bowtie (bin)

- Get gene expression read count with Salmon (bin)
Pausing Peak Tool

Pausing_Peak_Finder

- Identify pause peaks from bam file modified samtools (bin)
- Generate big-wig files for location of 3’ ends of reads
- Annotate peaks with info relative to genes or TSS from sqlite DB
- Generate Interactive web pages using DataTable and Plotly (javascript)
Pausing Peak Tool

Copy/export selected subsets of data

Data filtering and sorting
  • Text
  • Value Ranges

Expanding Annotation

Sequence Logos

Regenerate sequence logo on subset of data (remote web site)

Link to gene annotation (remote web sites)

Dynamic graph (based on selected row)
Using tags to surface information about files and processes (*.bam)

<table>
<thead>
<tr>
<th>TAGS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Escherchia</td>
<td>x</td>
</tr>
<tr>
<td>coli</td>
<td>x</td>
</tr>
<tr>
<td>NC_000913.2</td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Build</th>
<th>NC_000913.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genus</td>
<td>Escherchia</td>
</tr>
<tr>
<td>Species</td>
<td>coli</td>
</tr>
<tr>
<td>stats01 Aligner</td>
<td>Bowtie 1.2.2</td>
</tr>
<tr>
<td>stats02 Reads in this bam</td>
<td>3375586</td>
</tr>
<tr>
<td>stats03 FASTQ Reads Input</td>
<td>3585806</td>
</tr>
<tr>
<td>stats04 Cutadapt Reads Output</td>
<td>3528248</td>
</tr>
<tr>
<td>stats05 Clumpify Reads Output</td>
<td>3398154</td>
</tr>
<tr>
<td>stats06 Reads Input</td>
<td>3398154</td>
</tr>
<tr>
<td>stats07 Reads Unmapped</td>
<td>22568 (0.66%)</td>
</tr>
<tr>
<td>stats09 Mapped</td>
<td>3375586 (99.34%)</td>
</tr>
</tbody>
</table>
Tumor Mutational Burden

Haobin Chen, M.D., Ph.D.

Assistant Clinical Investigator
Thoracic Surgery Branch

Dr. Chen's research focuses on developing novel epigenetic therapies for small cell lung cancer. He is board certified in internal medicine and board certified in medical oncology.

Areas of Expertise
1) lung cancer 2) epigenetics 3) molecular biology
Tumor Mutation Burden Workflow

**SENTIEON TNsnv FASTQ TO VCF**

- PE/SE Blood FASTQ file
- PE/SE Tumor FASTQ file
- Ref FastQ & Index (BWA)

**SENTIEON TNSEQ FASTQ TO VCF**

- Cosmic Database
- GATK Bundle File
- Reads Groups CSV
- Agilent Sure Select v5 Target Region BED File

**Output:**
- combo_recal.bam
- TNsnv VCF
Tumor Mutational Burden Workflow

DNAlexus
SENTIEON TNSEQ FASTQ TO VCF

combo recal.bam

TNsnv VCF

BAM-Matcher

samtools split on Read Groups

Blood Bam

Tumor Bam

Strelka Applet

Streika VCF

SnpEff & VCFTools Annotation

SnpEff & VCFTools Annotation

Filtering

Reports

HTML Reports

HTML Report

Reports
Tumor Mutational Burden

• Annotation with snpEff and snpSift runs on **TNsnv** VCF and **Strelka2** VCF
  
  • **Cosmic67** - Catalogue of Mutations In Cancer, Welcome Sanger Institute
  
  • **ExACv03** - Exome Aggregation Consortium, Broad Institute
  
  • **1000 Genomes** - The International Genome Sample Resource
  
  • **dbSNPv138** - Single Nucleotide Polymorphism Database

• Filter Process

  • Missense mutations AND Cosmic67, OR

  • Missense mutations AND NOT in 3 dbs (ExACv03, 1000 Genomes, dbSNPv138)

• Filter Process Count represents Tumor Mutational Burden Value
Overview
Salmon is a tool for quantifying the expression of transcripts using RNA-seq data. Salmon uses new algorithms (specifically, coupling the concept of quasi-mapping with a two-phase inference procedure) to provide accurate expression estimates very quickly (i.e. wicked-fast) and while using little memory. Salmon performs its inference using an expressive and realistic model of RNA-seq data that takes into account experimental attributes and biases commonly observed in real RNA-seq data.

Workflow - Three Separate Applets

- Use Salmon to “align” to Gencode transcriptome and generate both quant files (Read Count and TPM per transcript) for both transcripts and genes. Additionally, gather data from optional bootstrap for subsequent use in Sleuth DEG program. This stage designed to run with separate node for each sample.

- Given a set of “read count files” (quant.sf) generate a combined count matrix or both transcripts and genes.

- Generate interactive HTML pages for each sample or combined samples for gene count, with graphic representation of bootstrap distribution.

- Hand off DEG and other tertiary analyses to Shiny Apps iDEP and/or Biojupies.
Salmon RNAseq Elements

- Two Assets
  - salmon_0.12.0_asset
  - r_base_3.5.2_asset
- Cloud Workstation
## Gene Expression Analysis

### Input Transcript Quant File Summary
1. `brain_rep2_quant.sf`
2. `brain_rep3_quant.sf`
3. `brain_rep1_quant.sf`
4. `muscle_rep2_quant.sf`
5. `muscle_rep3_quant.sf`
6. `muscle_rep1_quant.sf`

### Input Gene Quant File Summary
1. `brain_rep2_quant_genes.sf`
2. `brain_rep3_quant_genes.sf`
3. `brain_rep1_quant_genes.sf`
4. `muscle_rep2_quant_genes.sf`
5. `muscle_rep3_quant_genes.sf`
6. `muscle_rep1_quant_genes.sf`

### Instructions
1. Download Expression Tables from DNAnexus
   - Download RAW Counts Table for Transcripts
   - Download TPM (Transcripts Per Million) Counts Table for Transcripts
   - Download RAW Counts Table for Genes
   - Download TPM (Transcripts Per Million) Counts Table for Genes
2. Download and Edit Design Table in Excel or Text Editor for use in IDEP
3. Select Analysis Site and Upload Expression Table
   - Upload an Expression Table File to BioJudes
   - Upload an Expression Table File to IDEP

Created by: Genome Analysis Unit
Get Help for an Applet `salmon_spg_wf`

**Usage:**
```
dx run /SalmonWF/Applications/salmon_spg_wf [-INPUT_NAME=VALUE ...]
```

**Applet:** `salmon_spg_wf`

**Inputs:**
- `fastq_gz_list`: `-ifastq_gz_list=(file) [-ifastq_gz_list=... ...]`
- `salmon_idx_file`: `-isalmon_idx_file=(file)`
- `Bootstrap Value`: `[-ibootstrap_value=(int, default=0)]`
  Number of bootstraps to perform

**Outputs:**
- **Salmon Results Directories (tar.gz):** `quant_sf_files` (array:file)
- **Batch of quant_sf files:** `quant_sf_s` (array:file)
- **Salmon quant.sf files.**
- **Batch of quant_genes_sf files:** `quant_genes_sf_s` (array:file)
- **Salmon quant.genes.sf files.**
- **Batch of abundance_h5 files:** `abundance_h5_s` (array:file)
  Wasabi derived files need for Sleuth.
Upload Data for **salmon_spg_wf**

**Commands on Terminal to Upload Data**

```
dx mkdir /demo_data
dx upload *.fastq.gz --destination /demo_data
dx upload yeast_S288C_salmon_idx.tar.gz --destination /demo_data/yeast_S288C_salmon_idx.tar.gz
dx ls -l /demo_data
```
Run `salmon_spg_wf`

Commands on Terminal to run and see results for `salmon_spg_wf`

```
dx run /SalmonWF/Applications/salmon_spg_wf\ 
  -ifastq_gz_list=/demo_data/DST1_G418_B_R1.fastq.gz \ 
  -ifastq_gz_list=/demo_data/DST1_G418_B_R2.fastq.gz \ 
  -ifastq_gz_list=/demo_data/DST1_G418_C_R1.fastq.gz \ 
  -ifastq_gz_list=/demo_data/DST1_G418_C_R2.fastq.gz \ 
  -isalmon_idx_file=/demo_data/yeast_S288C_salmon_idx.tar.gz \ 
  -ibootstrap_value=0 \ 
  --destination /demo_result
```

dx ls -l /demo_result

Project: GAU_Development (project-FVqKF6j0v1xv6fxK15Bzj9B2)
Folder: /demo_result

<table>
<thead>
<tr>
<th>State</th>
<th>Last modified</th>
<th>Size</th>
<th>Name (ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>81.93 KB</td>
<td>DST1_G418_B_abundance.h5 (file-FXg33yj0bKxY49FpJvfKb0fX)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>218.73 KB</td>
<td>DST1_G418_B_quant.sf (file-FXg33y00bKxx4BbZ5X7k0k3V)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>179.09 KB</td>
<td>DST1_G418_B_quant_genes.sf (file-FXg33y00bKxx4BbZ5X7k0k3V)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>613.44 KB</td>
<td>DST1_G418_B_salmon.tar.gz (file-FXg33x80bKxxBpp8JvVX4qBF)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>82.36 KB</td>
<td>DST1_G418_C_abundance.h5 (file-FXg3400bKxxZVvY9VXJp6G)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>219.41 KB</td>
<td>DST1_G418_C_quant.sf (file-FXg33z80bKxxVfF3BF3gXY6K)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>179.46 KB</td>
<td>DST1_G418_C_quant_genes.sf (file-FXg33zj0bKvq47P9xyQgXY)</td>
</tr>
<tr>
<td>closed</td>
<td>2019-04-10 14:13:55</td>
<td>616.78 KB</td>
<td>DST1_G418_C_salmon.tar.gz (file-FXg33yj0bKxxG5PXJv13fBbJ)</td>
</tr>
</tbody>
</table>
# salmon_spg_wf

This application takes

[Created by GAU](https://gaucancer.gov)

## About Applet ...

Salmon Scatter-Process_Gather Workflow

This applet processes a batch of pair-end \_*fastq.gz* read files and runs Salmon

To use the developer's words:
> Salmon is a tool for wicked-fast transcript quantification from RNA-seq data. It requires a set of target transcripts (either from a reference or de novo assembly) to quantify. All you need to run Salmon is PASTA file containing your reference transcripts and a set of FASTA/PASTA files containing your reads. Optionally, Salmon can make use of pre-computed alignments (in the form of a SAM/BAM file) to the transcripts rather than the raw reads.

Developed by: [Fitzgerald, Peter (NHGRI) E] (fitzgerp@email.nih.gov)

Developed by: [McIntosh, Carl (NHGRI) E] (mcintosh@email.nih.gov)

Group: [Genome Analysis Unit](https://gaucancer.gov)

## Required Input Files

- \_*FASTQ Gzip Compressed Paired-end Files* - A batch sample PE read files with the form \_*R1.fastq.gz* and \_*R2.fastq.gz*.
- \_*Salmon Index tar.gz File* - A salmon Indexed genome files with the form \_*SalmonIndex.tar.gz*.

## Input Parameters

- \_*Output Folder* - Provide an output directory name for result files.
- \_*Instance type* - Asking for more compute resources will reduce run time and will cost more.

## COMMON Input Parameters
DNAnexus Developer Pages


Support Pages

• DNAnexus CCR Pilot  
  (https://gau.ccr.cancer.gov/dna-nexus-pilot-program/)

• Slack Channel for CCR_DNAnexus Pilot (dnaxpilot.slack.com)  
  (help, general, development)

• Creating Assets: https://gau.ccr.cancer.gov/about-dnanexus-asset/

• Building

• Example About Pages:
  • https://gau.ccr.cancer.gov/rnaseq_salmon/
  • https://gau.ccr.cancer.gov/salmon_spg_wf/
  • https://gau.ccr.cancer.gov/quant_sf2express_table/