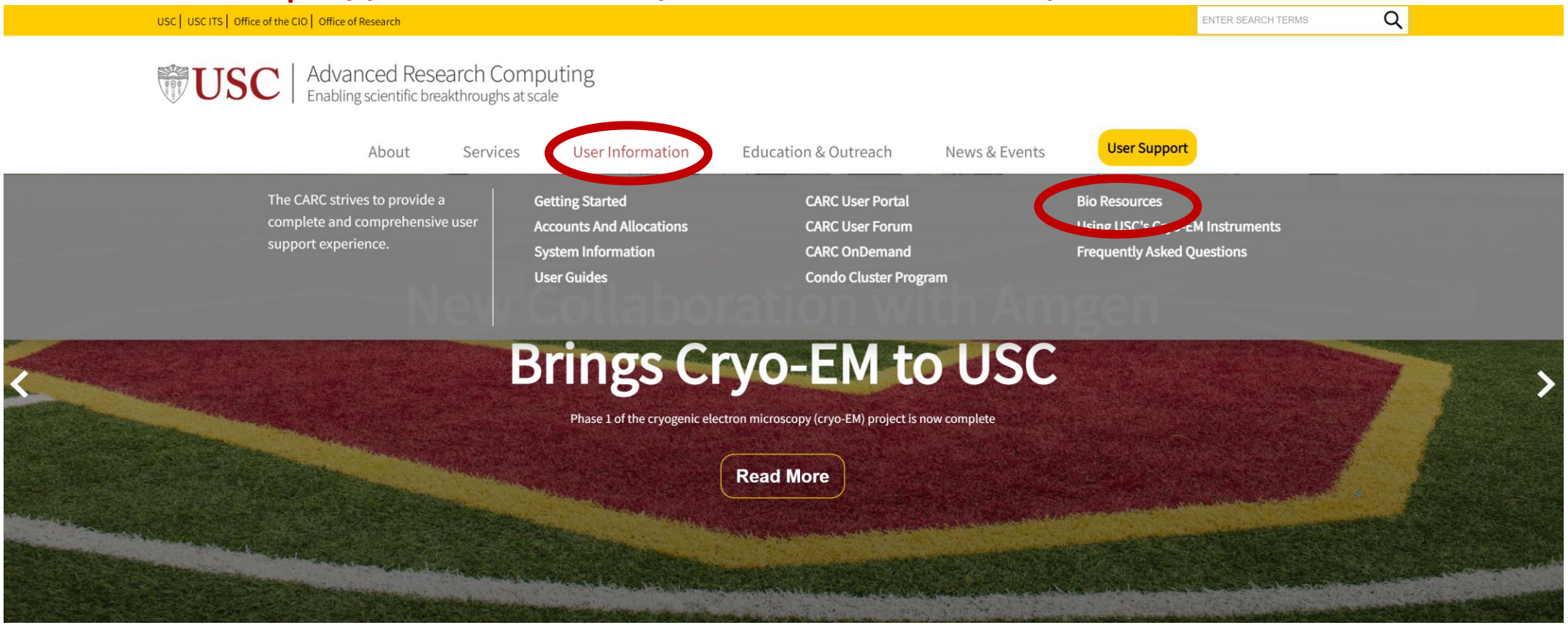
Computational Biology on CARC Systems

Center for Advanced Research Computing
University of Southern California
Tomasz Osinski, PhD
Research Facilitator in Life Science

USC | Advanced Research Computing
Enabling scientific breakthroughs at scale

Bio Resources (updated twice a year)

https://carc.usc.edu/user-information/bio-resources



Bio Resources (updated once a year)

User Information

Getting Started
Accounts and Allocations
System Information
User Guides
CARC User Portal
CARC User Forum
CARC OnDemand
Condo Cluster Program
Bio Resources
Using USC's Cryo-EM Instruments
Frequently Asked Questions

Bio Resources

Reference genomes, protein and nucleotide sequences databases, and other bio resources are now available on Discovery and Endeavour. If you need a specific release that is not currently included in the pages below, please submit a help ticket and we will try to make those resources available to you.

Genomes

A set of ready-to-use reference sequences and annotations for commonly analyzed organisms, sourced from iGenomes.

Genbank

The GenBank sequence database is an open access, annotated collection of all publicly available nucleotide sequences and their protein translations.

Genome Taxonomy Database (GTDB)

The Genome Taxonomy Database (GTDB) is an initiative to establish a standardized microbial taxonomy based on genome phylogeny.

Pfam Database

The Pfam database is a large collection of protein families, each represented by multiple sequence alignments and hidden Markov models (HMMs).

TIGRFAMs

TIGRFAMs is a resource consisting of curated multiple sequence alignments, Hidden Markov Models (HMMs) for protein sequence classification, and associated information designed to support automated annotation of (mostly prokaryotic) proteins.

UniProt

The Universal Protein Resource (UniProt), a collaboration between the European Bioinformatics Institute (EBI), the SIB Swiss Institute of Bioinformatics, and the Protein Information Resource (PIR)

Bio Resources (updated twice a year)

https://carc.usc.edu/user-information/bio-resources

- **Genomes** - reference sequences and annotations for commonly analyzed organisms
- **Genbank** - collection of all public nucleotide sequences and their protein translations
- **Genome Taxonomy Database (GTDB)** - an initiative to establish a standardized microbial taxonomy based on genome phylogeny
- **Pfam Database** - large collection of protein families, each represented by multiple sequence alignments and hidden Markov models (HMMs)
- **TIGRFAMs** - curated multiple sequence alignments, Hidden Markov Models (HMMs) for protein sequence classification
- **UniProt** – UniProtKB (curated protein information), UniRef (closely related sequences), UniParc (all protein sequences, consisting only of unique identifiers and sequences)

Bio Resources (command line access)

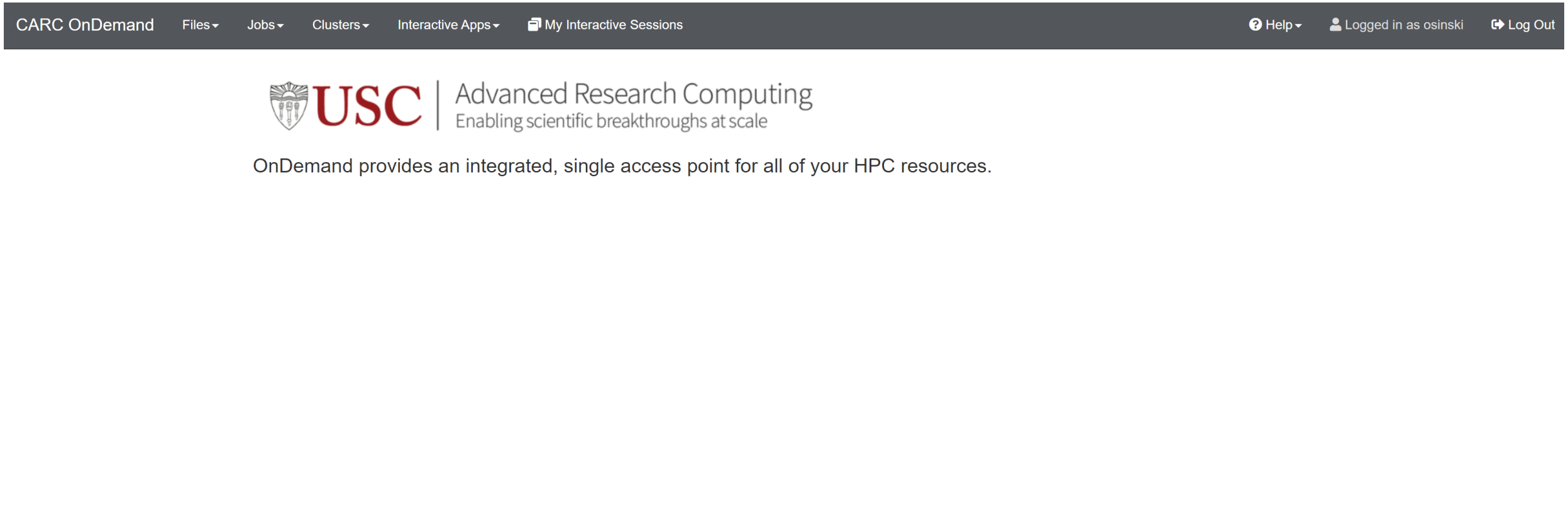
https://carc.usc.edu/user-information/bio-resources

- **Biogeotraces** - set of metagenomes, collected under the auspices of the bioGEOTRACES component of the international GEOTRACES program
  `/project/biodb/biogeotraces`
- **TaraOceans** - marine microbial metagenomes sampled across space and time
  `/project/biodb/taraoceans`
- **Variant Effect Predictor cache** - VEP can use a variety of annotation sources to retrieve the transcript models used to predict consequence types. Cache contains all transcript models, regulatory features and variant data for a species and allows for an offline use of VEP
  `/project/biodb/vep-cache`

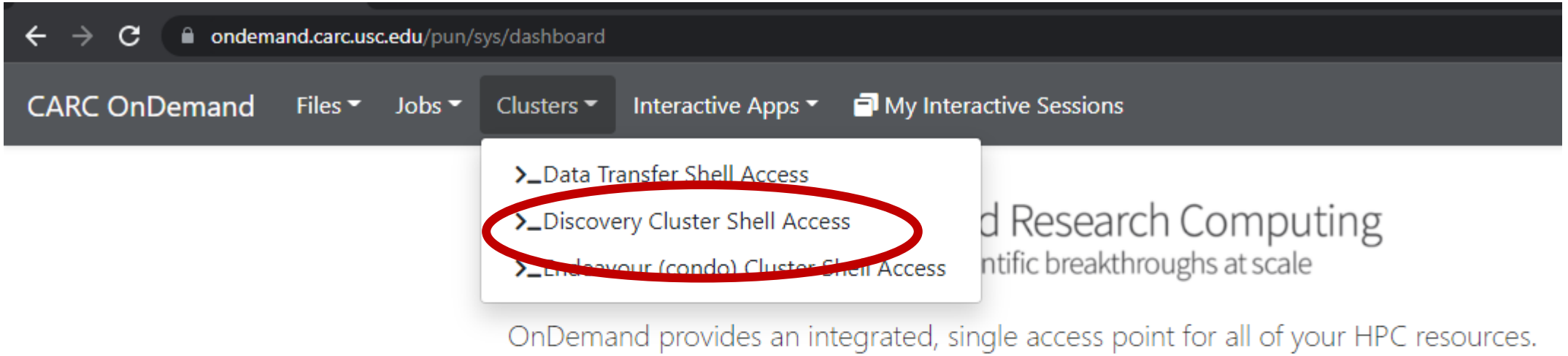
Log into CARC OnDemand

- Connect to the USC VPN (`connect.usc.edu`) or USC Secure Wifi
- Open the web browser and go to:
  - https://ondemand.carc.usc.edu
- Enter your username and password
- Choose an option in Duo-2FA, and confirm your access
- Done. You should see the CARC OnDemand dashboard

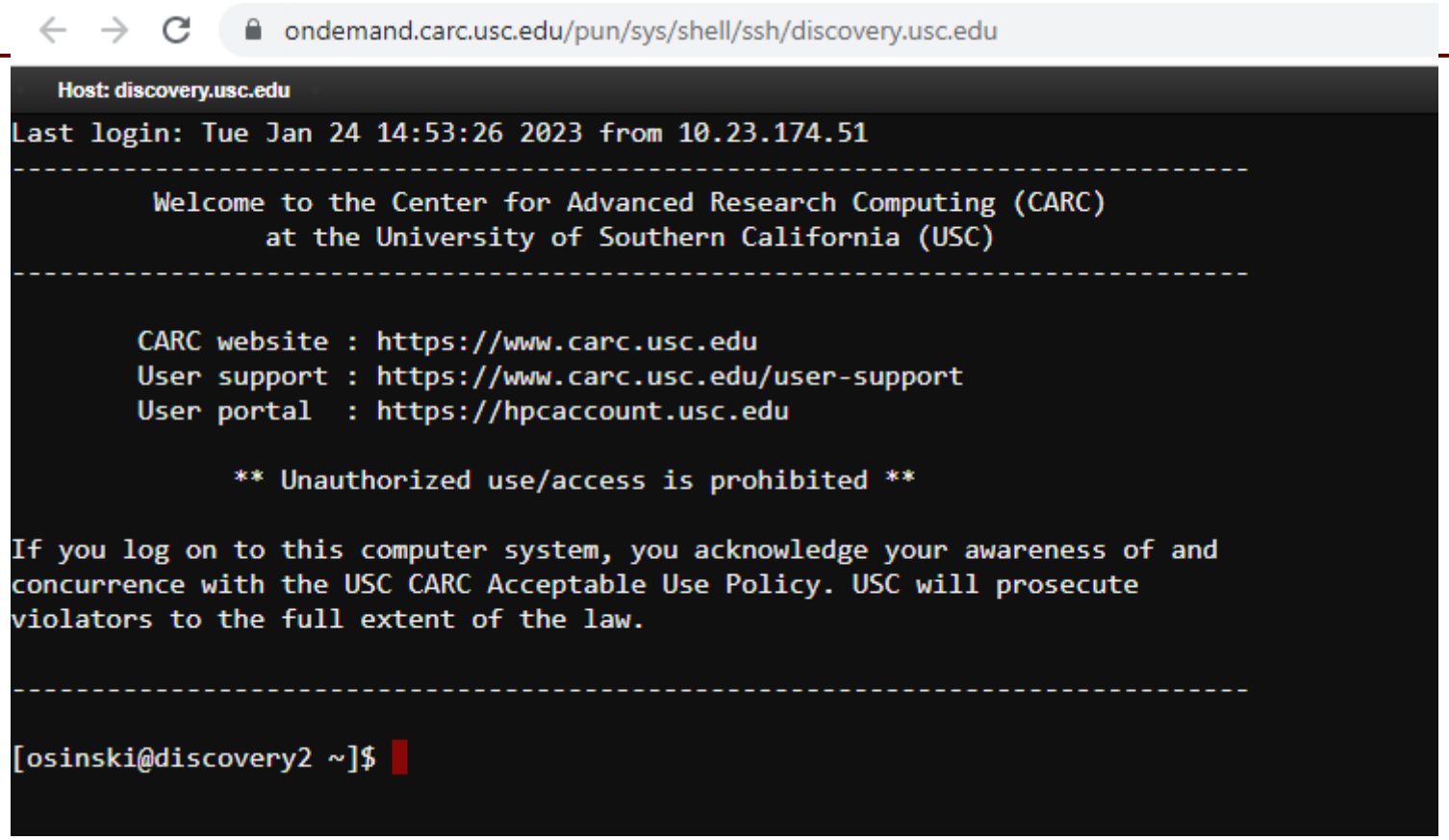
CARC OnDemand Dashboard



CARC OnDemand Dashboard

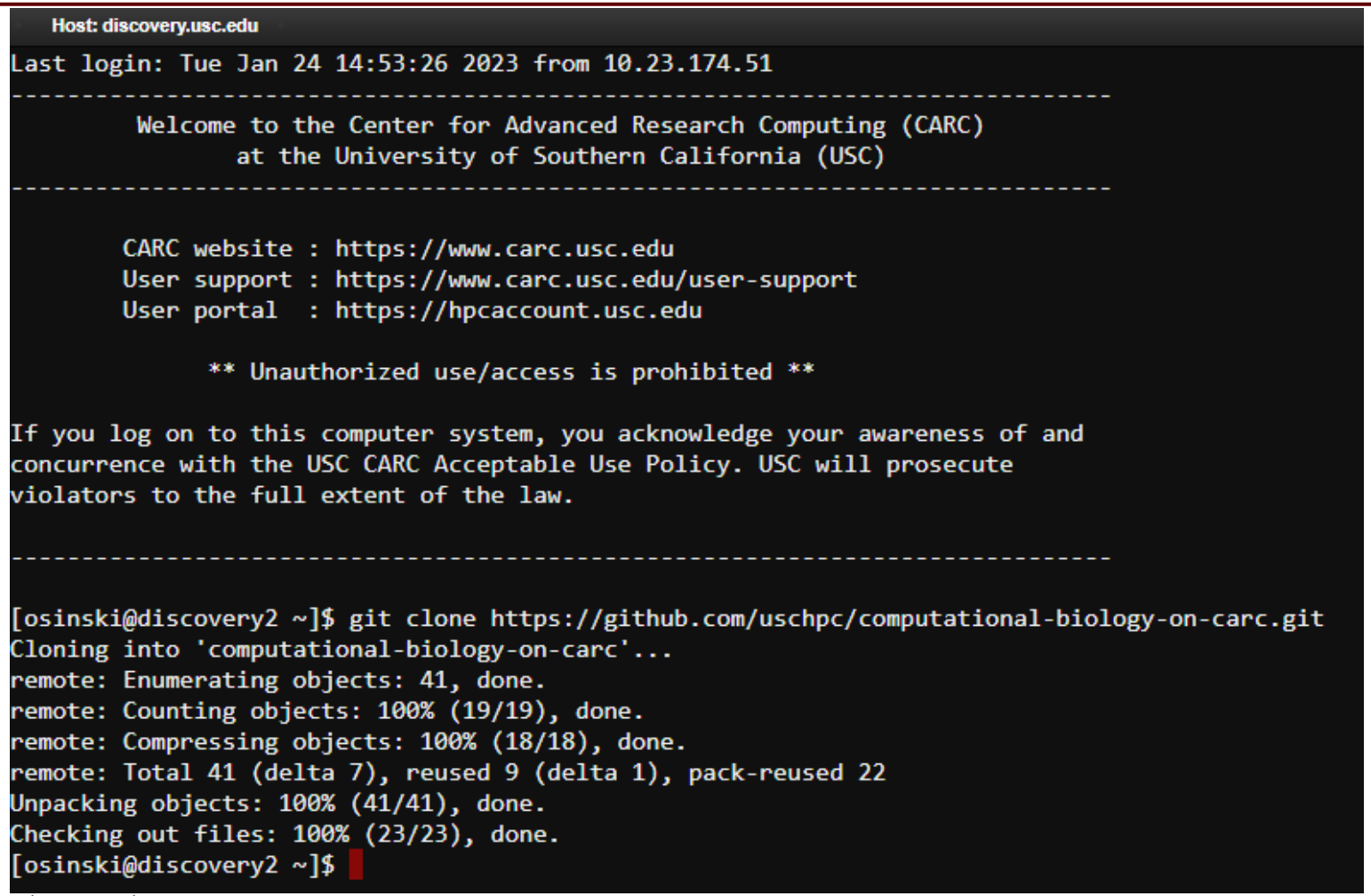


CARC OnDemand Dashboard

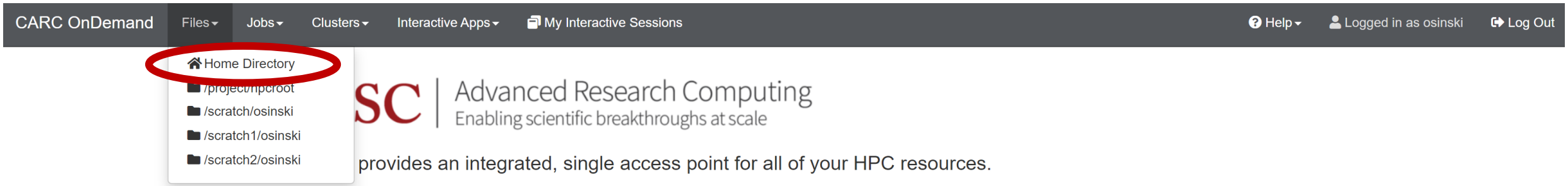


Copy the text below:
`git clone https://github.com/uschpc/computational-biology-on-carc.git`
and paste it in the terminal by pressing Shift+Ins keys simultaneously, then press Enter

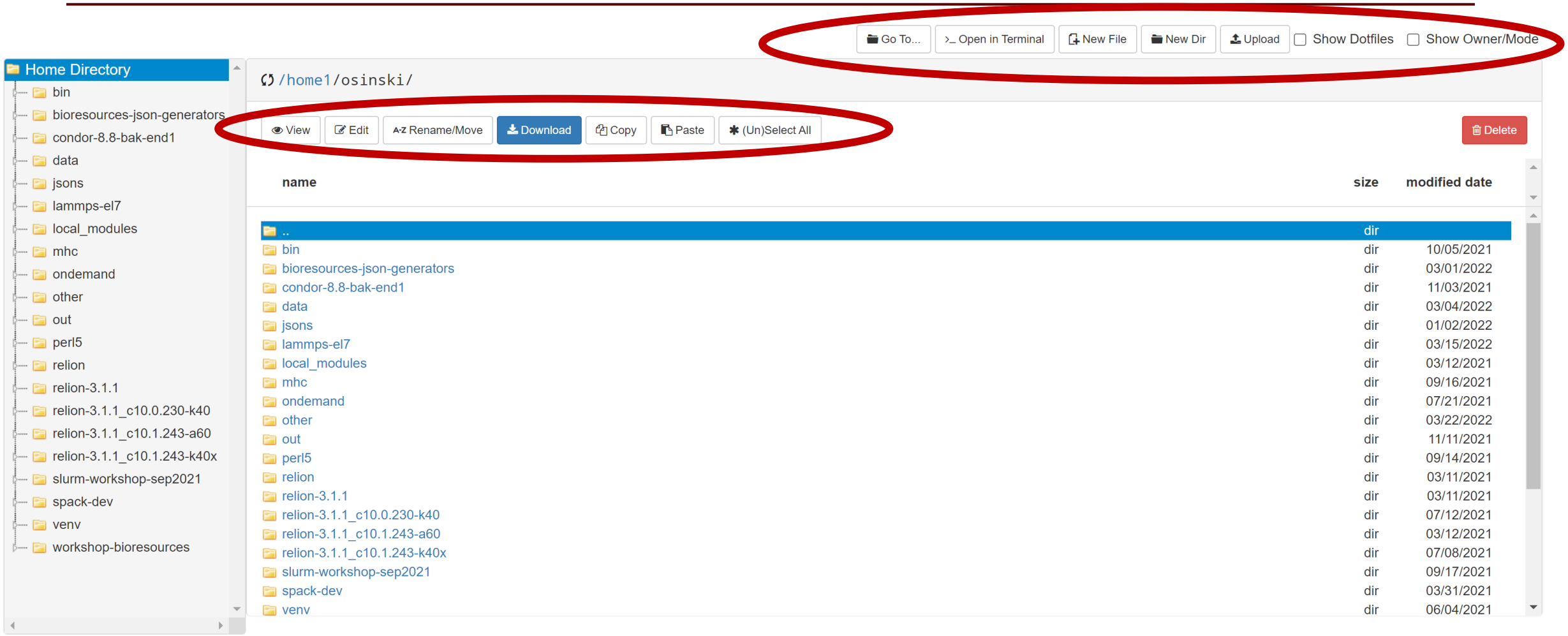
CARC OnDemand Dashboard



CARC OnDemand: File Management



CARC OnDemand: File Management

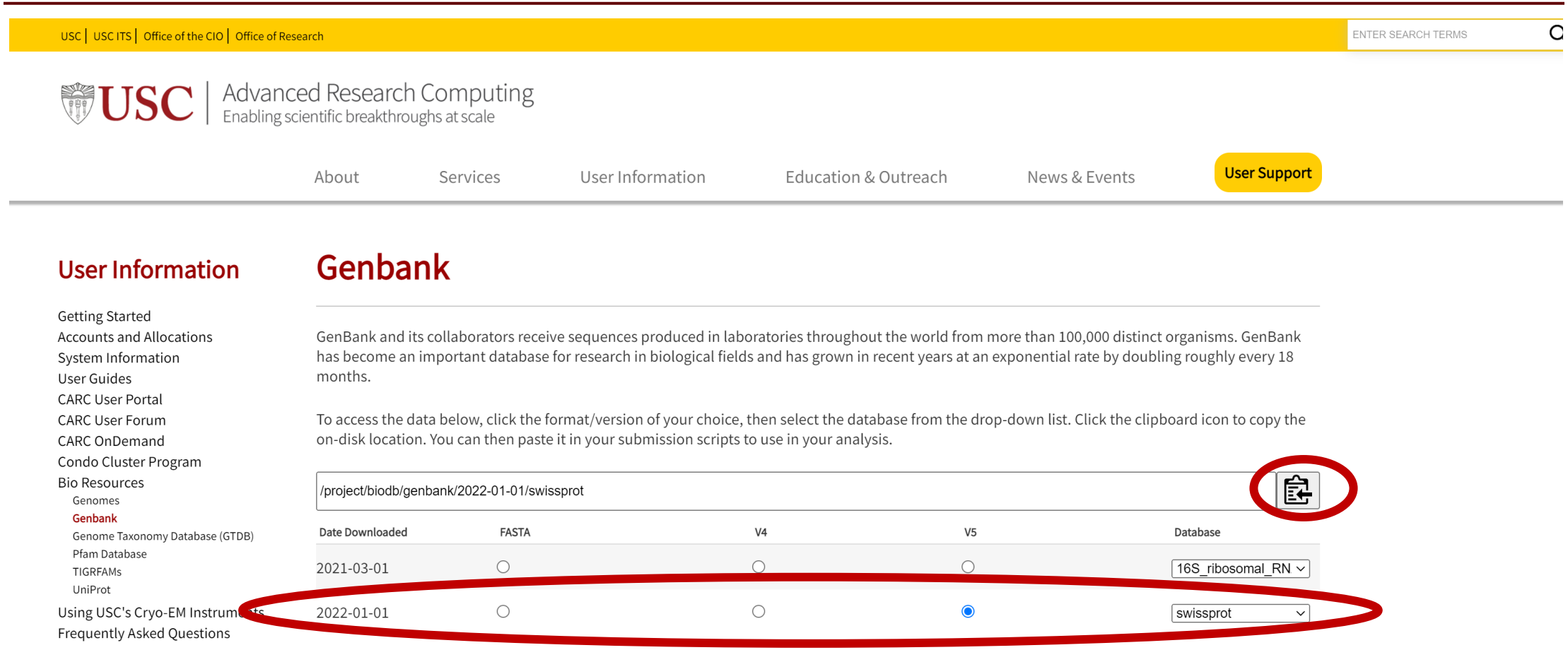


Create the BLAST Job Script

- Replace swissprot with the path to the v5 of swissprot db obtained from https://carc.usc.edu/user-information/bio-resources/genbank

```
#!/bin/bash
#SBATCH --nodes 1
#SBATCH --ntasks 10
#SBATCH --partition debug
#SBATCH --time 00:05:00
#SBATCH --account=ttrojan_001
#SBATCH --chdir /home1/ttrojan/computational-biology-on-carc
module purge
module load gcc
module load blast-plus
echo "Start BLAST Job"
blastp -db swissprot -query blast/query.txt -out results/blast/results.txt -num_threads
$SLURM_NTASKS
echo "Finish BLAST Job"
```

Create the BLAST Job Script



CARC OnDemand: Editing a Job

Save /home1/osinski/computational-biology-on-carc/jobs/blast1.sh

```
1  #!/bin/bash
2  #SBATCH --nodes 1
3  #SBATCH --ntasks 10
4  #SBATCH --partition debug
5  #SBATCH --time=00:05:00
6  #SBATCH --account=ttrojan_001
7  #SBATCH --chdir /home1/ttrojan/computational-biology-on-carc
8  module purge
9  module load gcc
10 module load blast-plus
11 echo “Example blast start”
12 sleep 20
13 blastp -db /project/biodb/genbank/2021-03-01/swissprot -query data/blast/query.txt -out results/blast/results.txt -num_threads $SLURM_NTASKS
14 echo “Example blast end”
15
```

Submit a Job – shell

- Submit the job

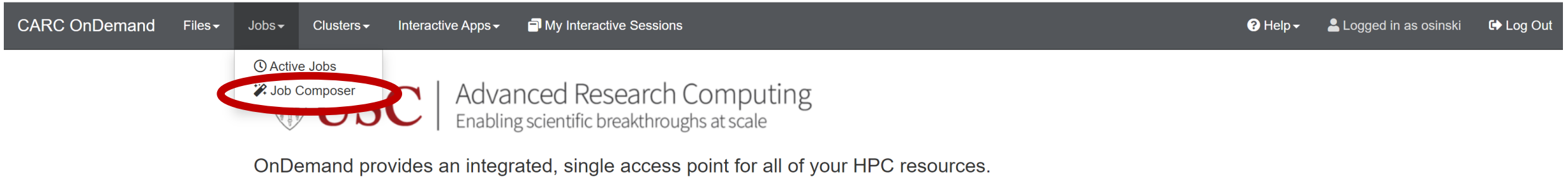
```
[ttrojan@discovery1 ~]$ sbatch blast1.sh

Submitted batch job 4773117
```

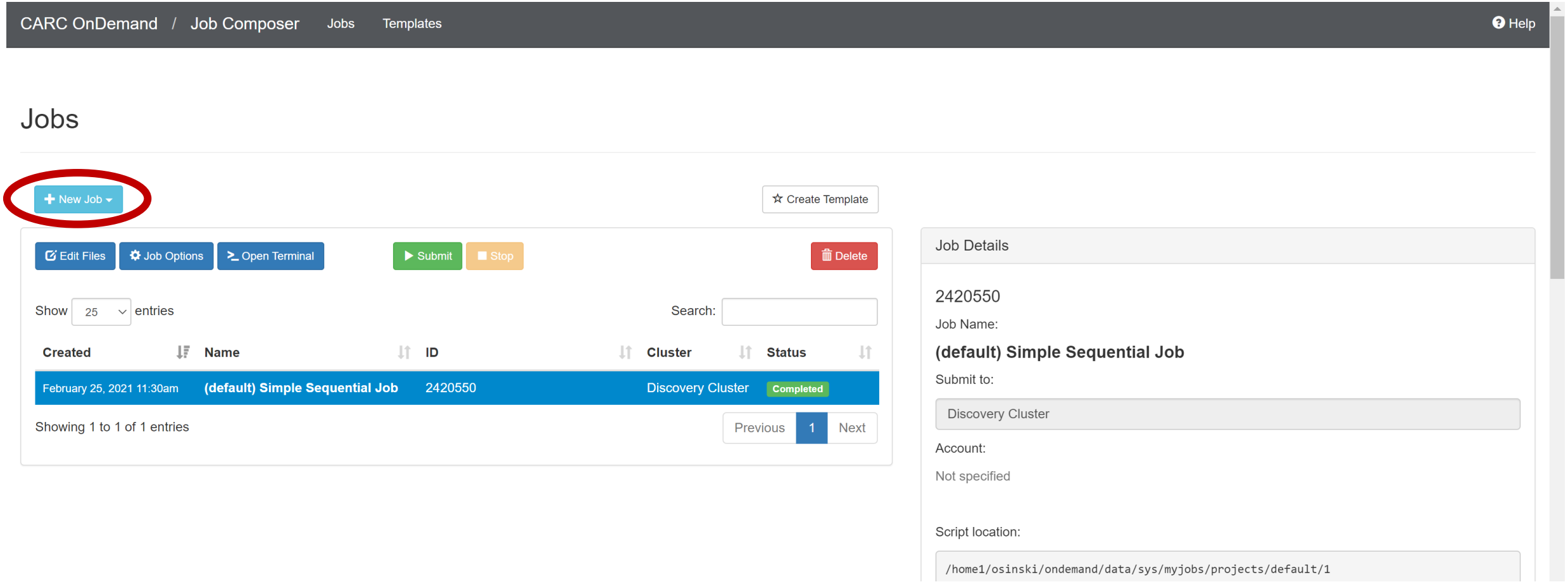
- Check the status of the job

```
[ttrojan@discovery1 ~]$ squeue –u ttrojan
JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
4773117      Main   blast1.j  ttrojan  R       0:02      1 a02-d11
```

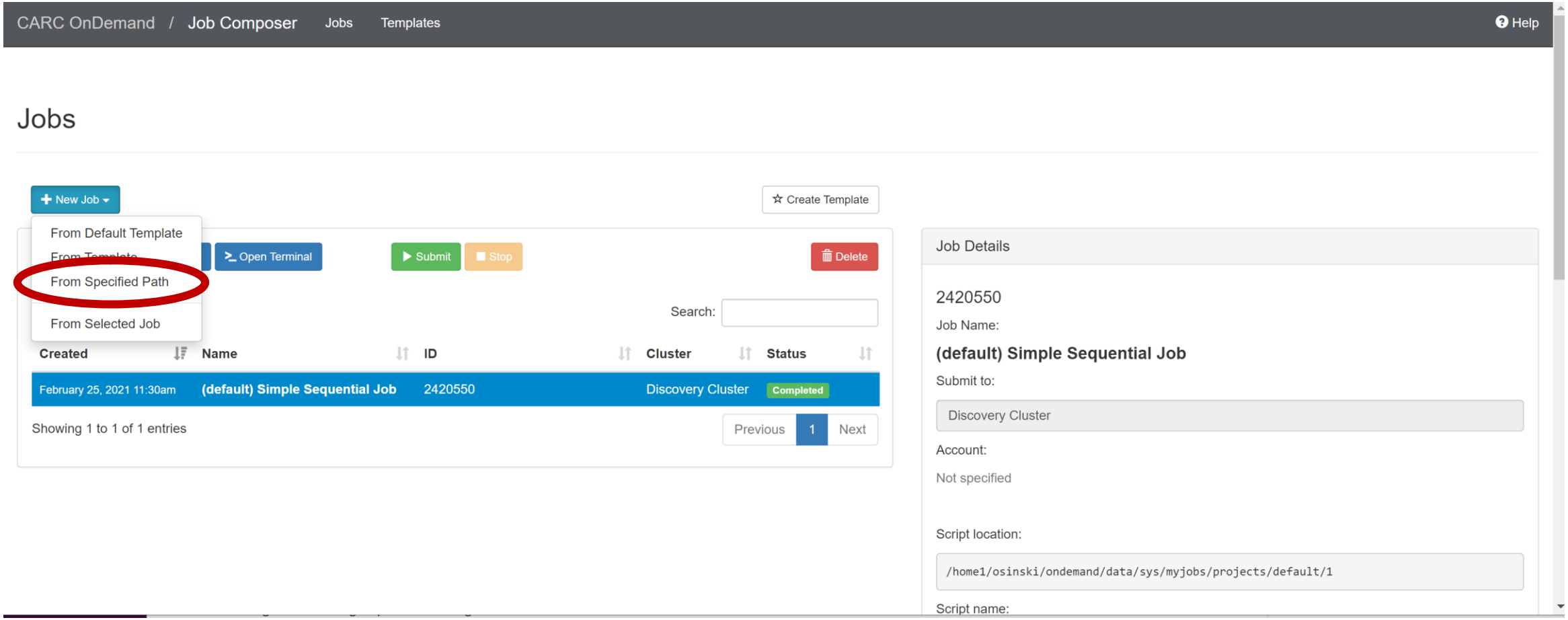
CARC OnDemand: Job Composer



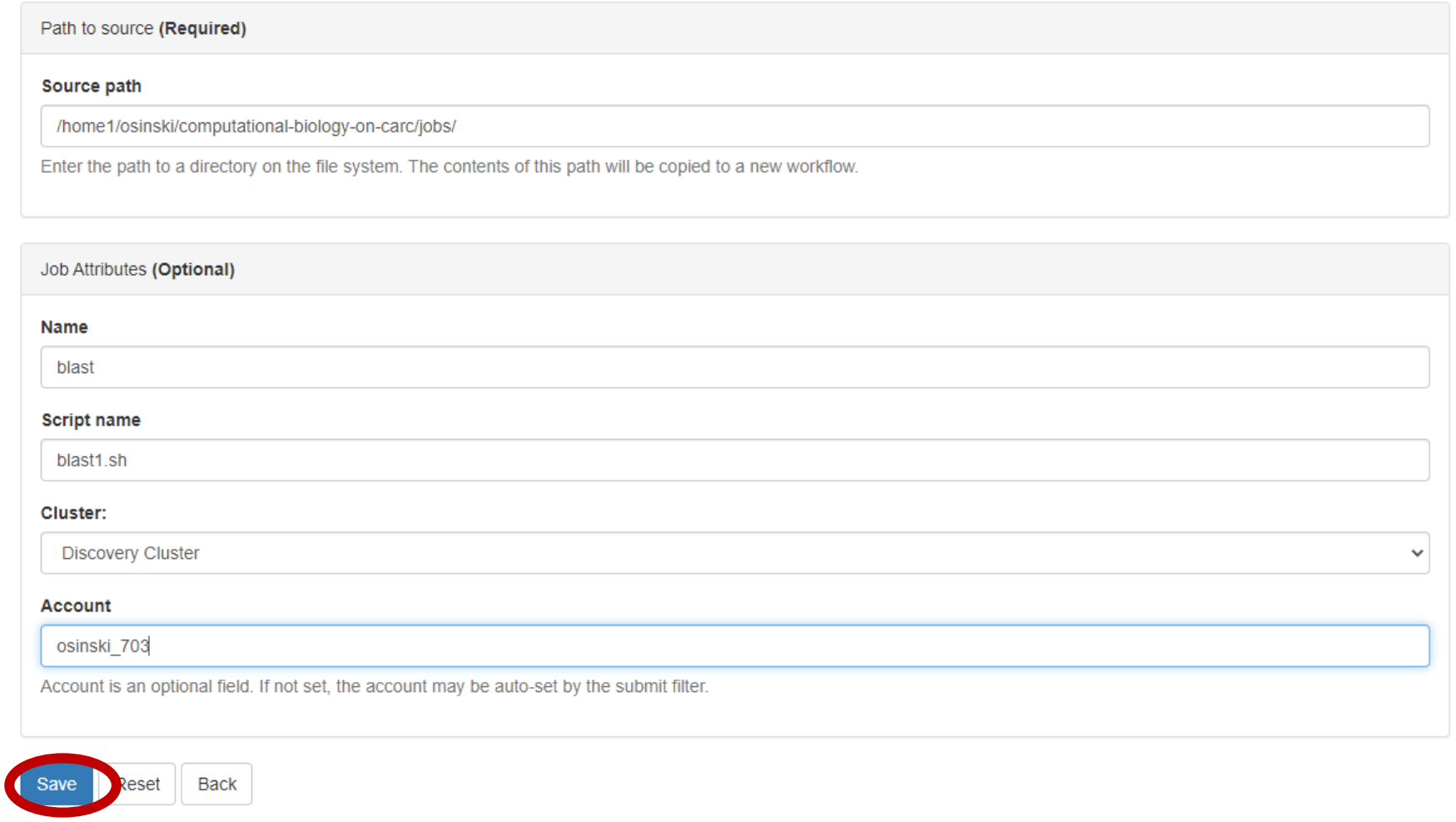
CARC OnDemand: Job Creation



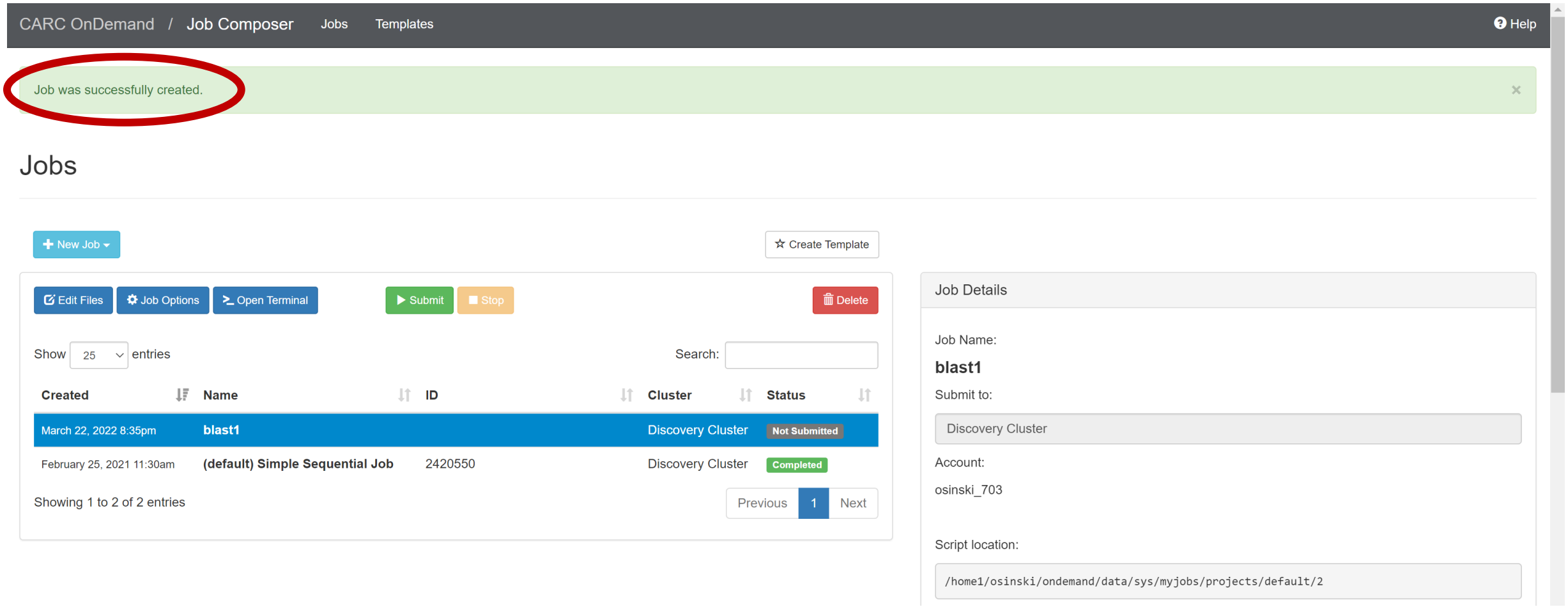
CARC OnDemand: Job Creation



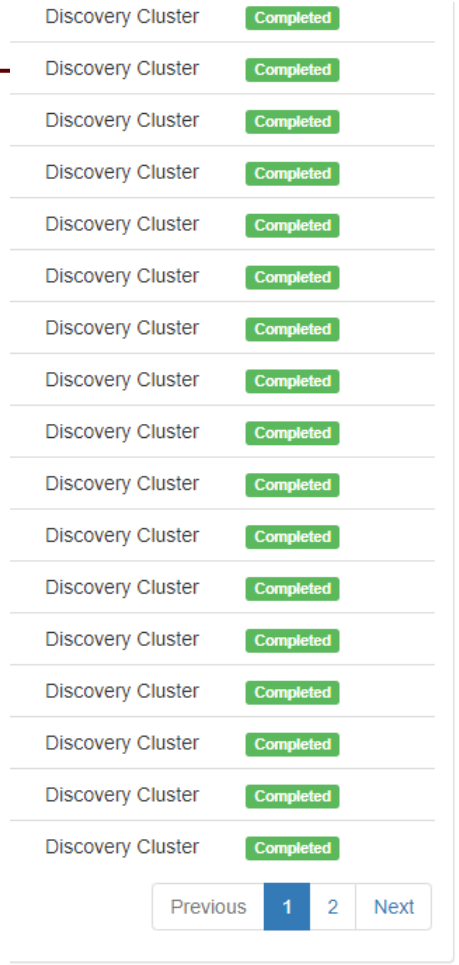
CARC OnDemand: Job Creation

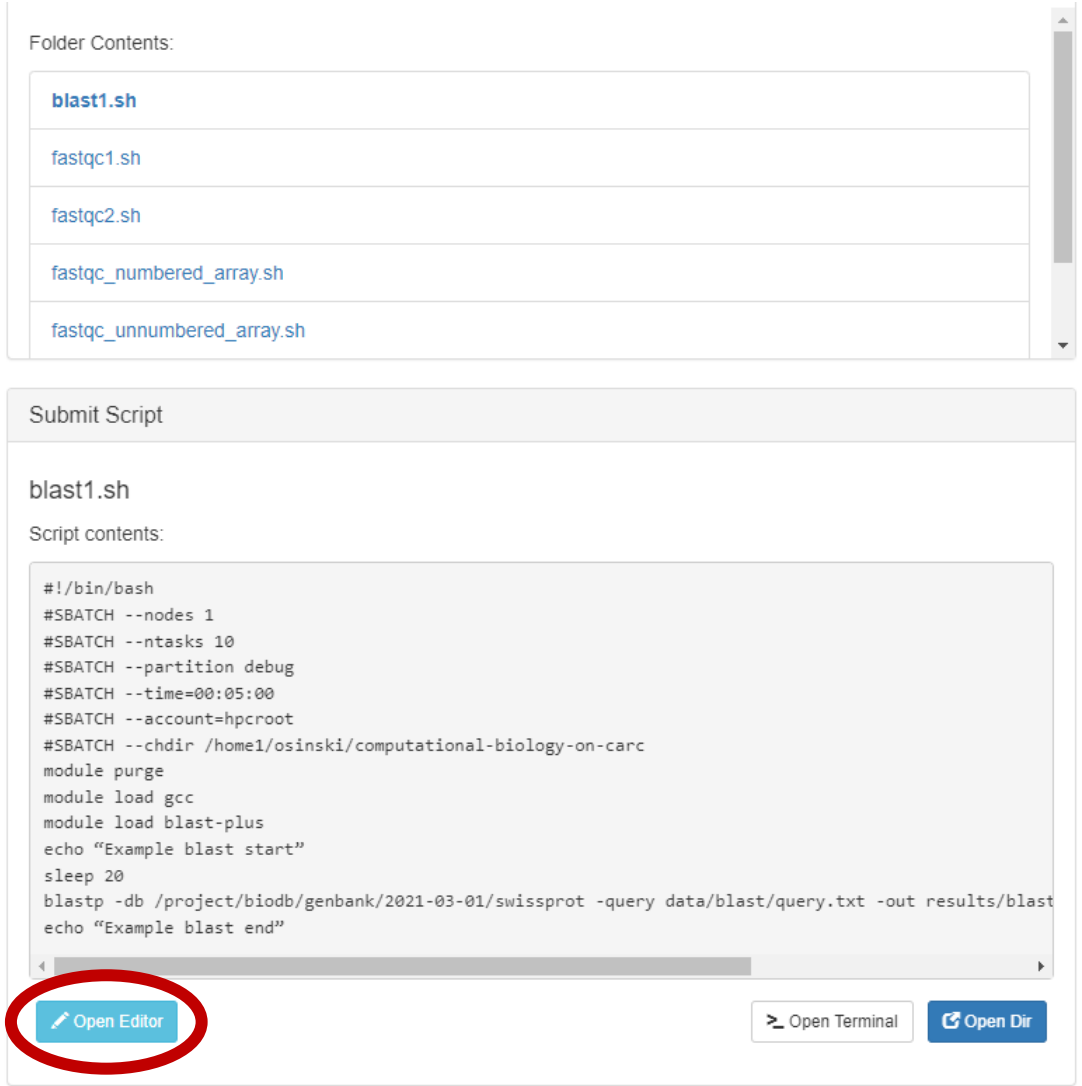


CARC OnDemand: Job Creation

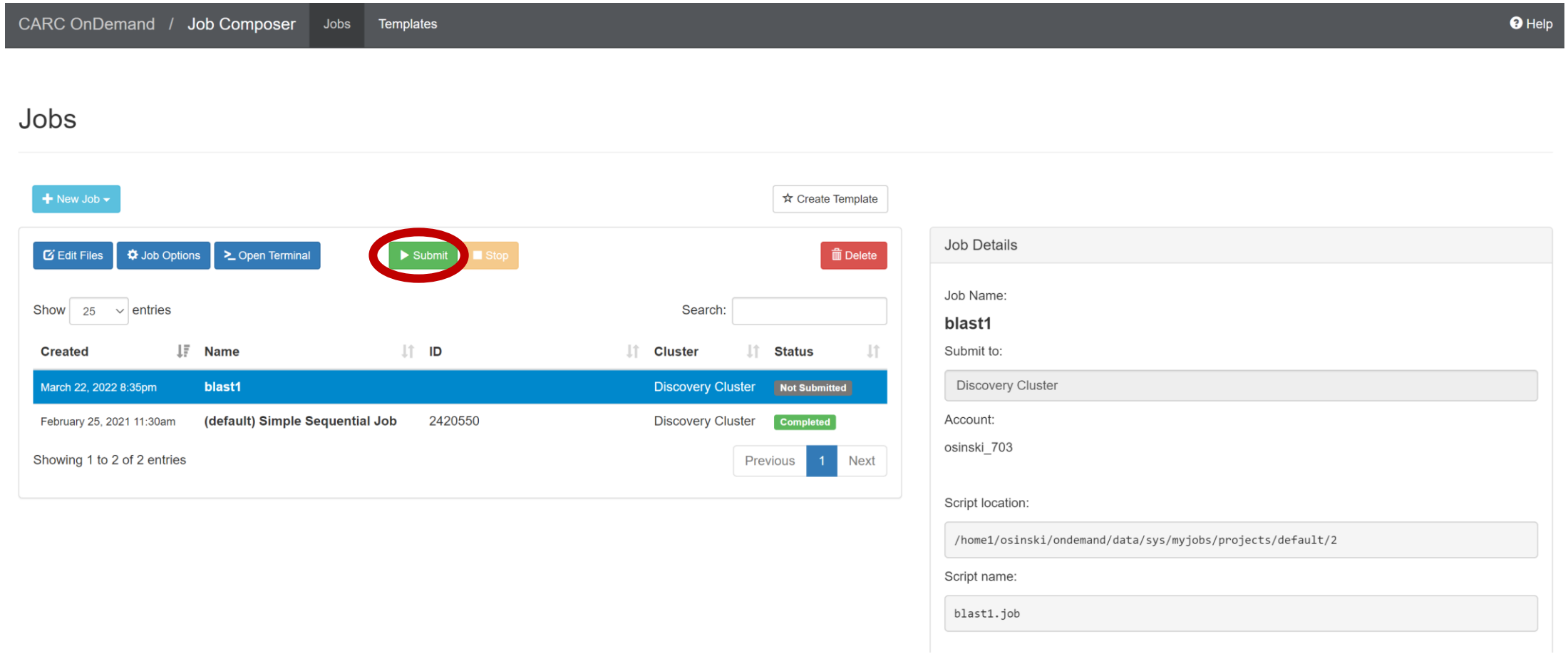


CARC OnDemand: Editing a Job

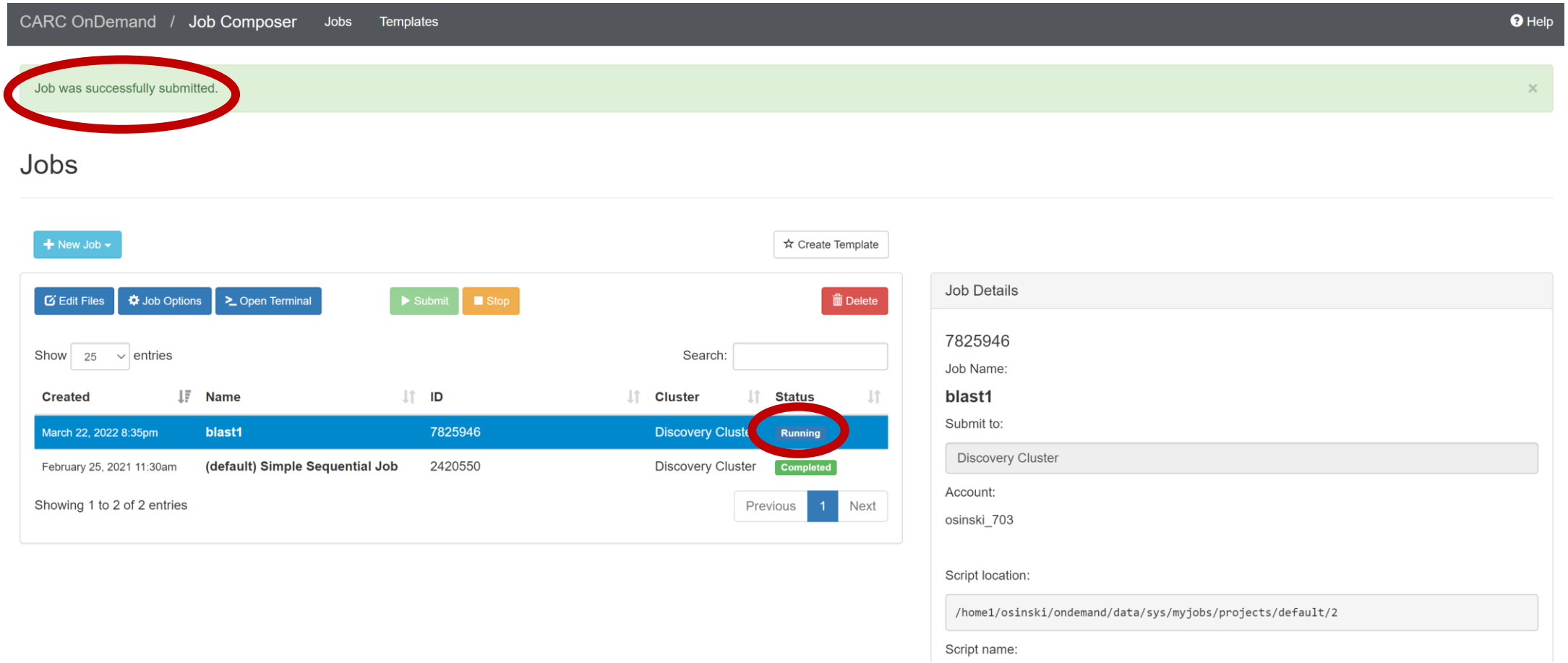




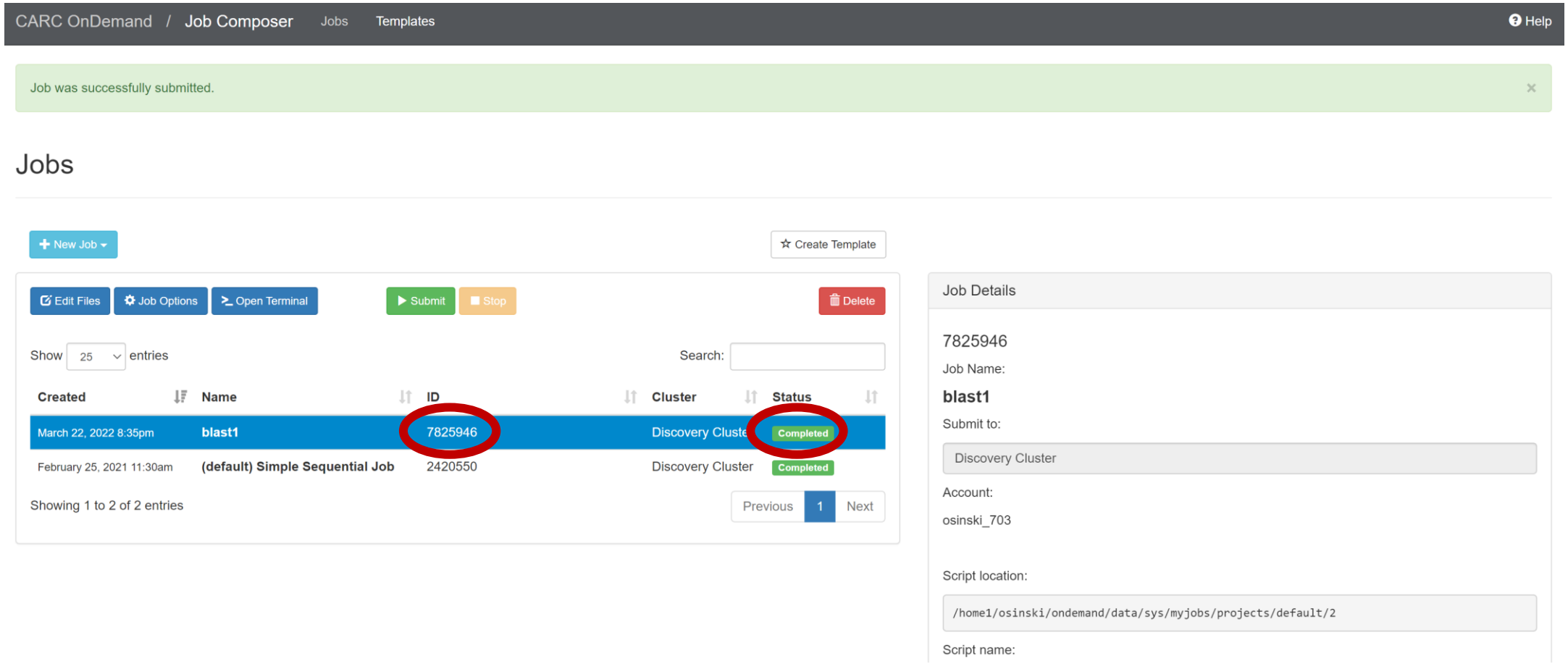
CARC OnDemand: Submitting a Job



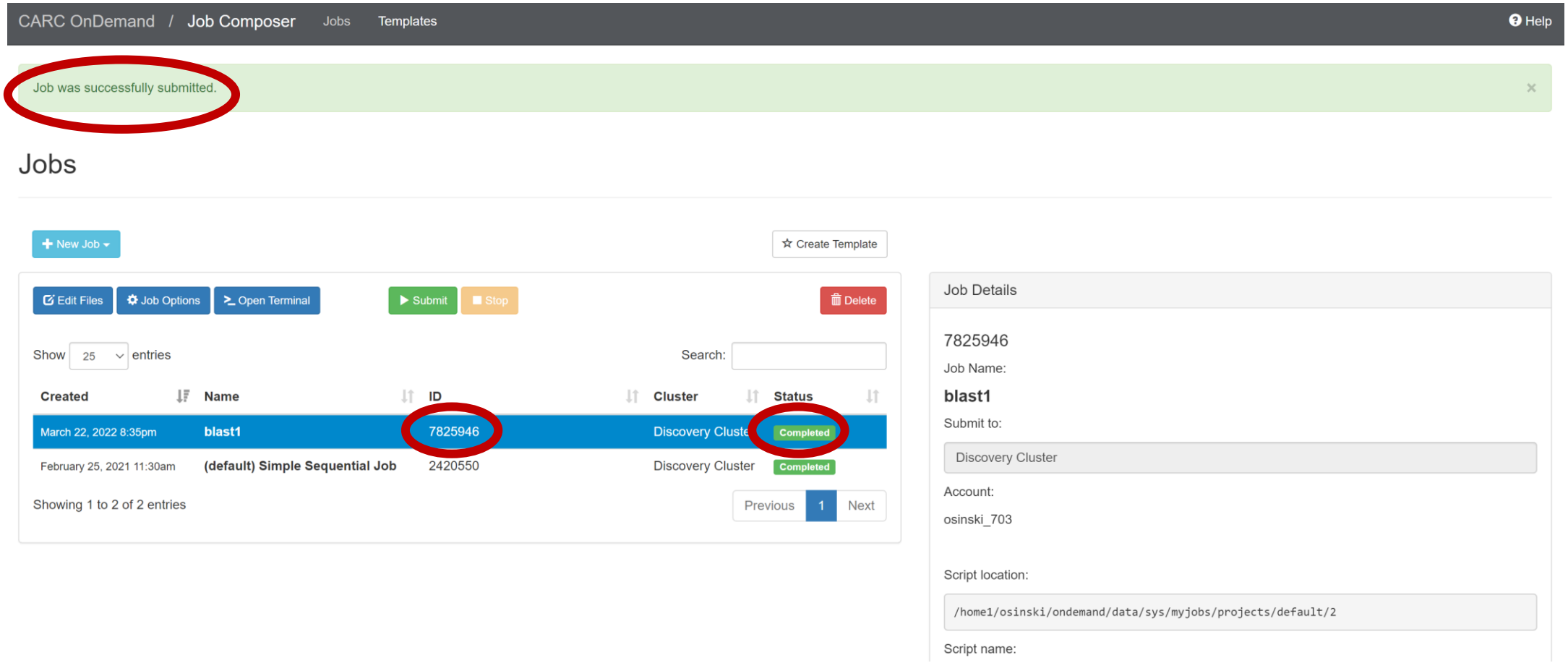
CARC OnDemand: Submitting a Job



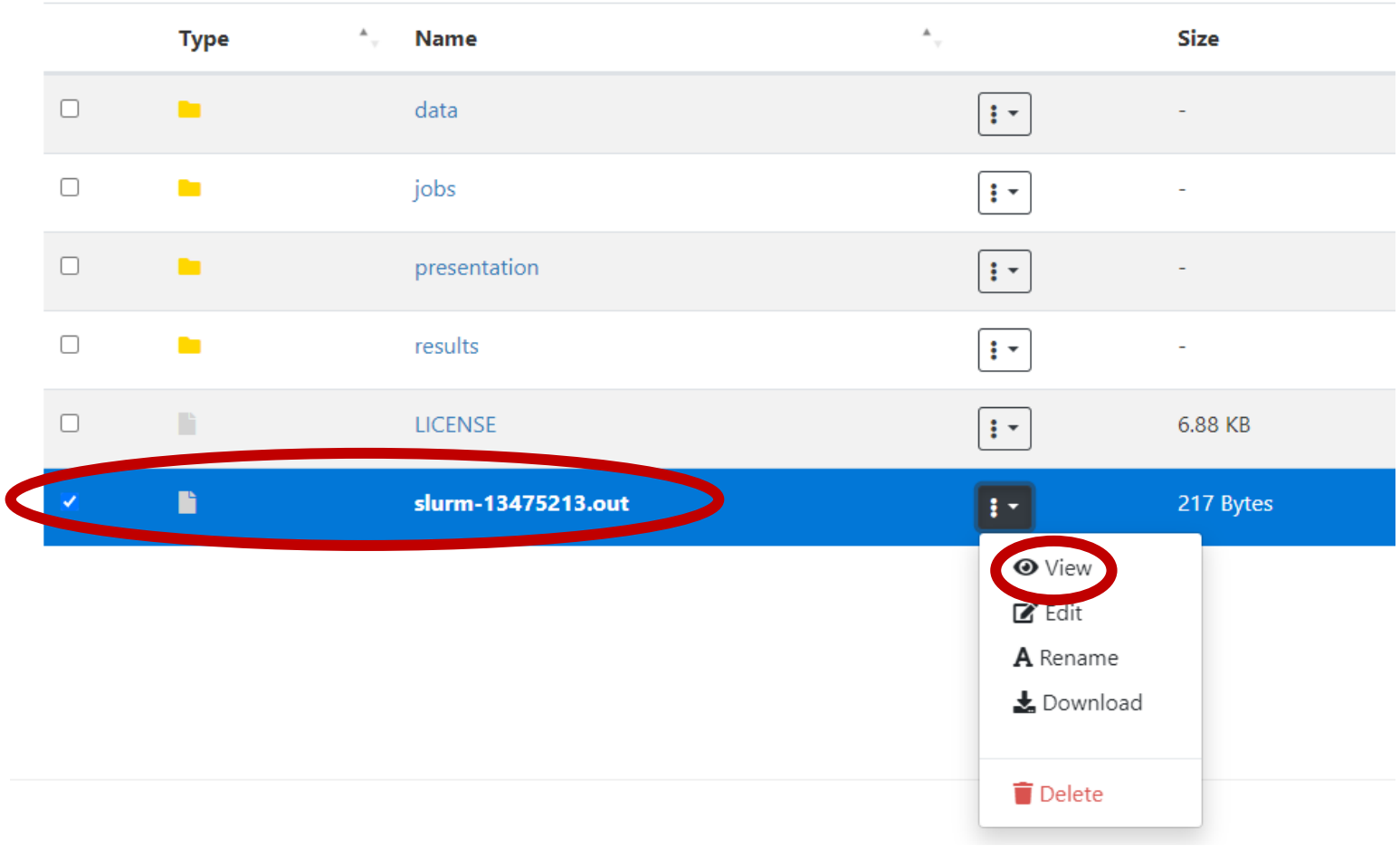
CARC OnDemand: Results



CARC OnDemand: Results



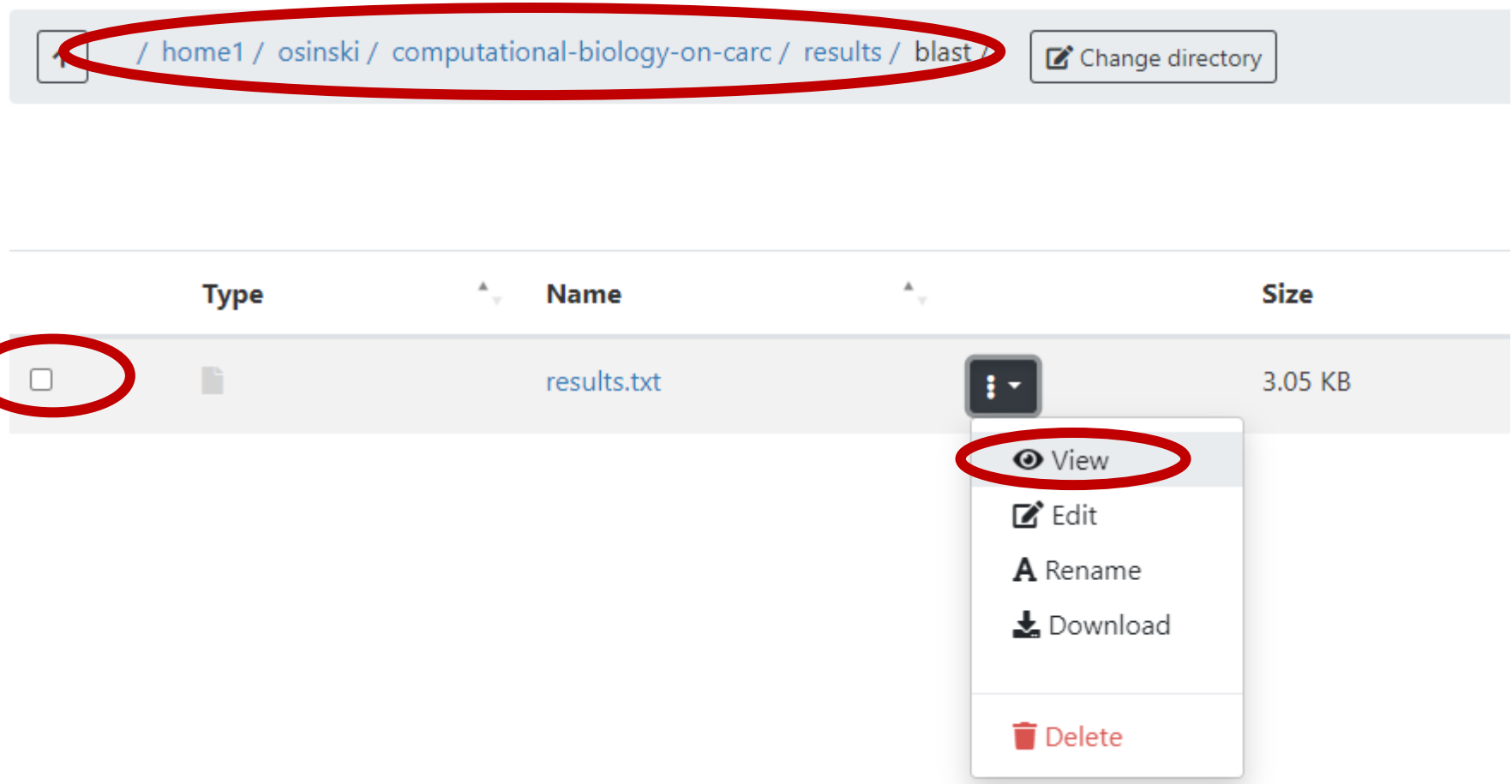
CARC OnDemand: Results



CARC OnDemand: Results

```
==========================================
SLURM_JOB_ID = 7825946
SLURM_JOB_NODELIST = e09-18
TMPDIR = /tmp/SLURM_7825946
==========================================
“Example blast start”
“Example blast end”
```

CARC OnDemand: Results



CARC OnDemand: Results

```
Query=
Length=15
                                                                      Score     E
Sequences producing significant alignments:                          (Bits)  Value

Q9JK11.1 RecName: Full=Reticulon-4; AltName: Full=Foocen; AltName...  35.0    0.001
Q99P72.2 RecName: Full=Reticulon-4; AltName: Full=Neurite outgrow...  35.0    0.001
Q9NQC3.2 RecName: Full=Reticulon-4; AltName: Full=Foocen; AltName...  33.9    0.004


>Q9JK11.1 RecName: Full=Reticulon-4; AltName: Full=Foocen; AltName: Full=Glut4
vesicle 20 kDa protein; AltName: Full=Neurite outgrowth
inhibitor; Short=Nogo protein [Rattus norvegicus]
Length=1163

 Score = 35.0 bits (79),  Expect = 0.001, Method: Composition-based stats.
 Identities = 15/15 (100%), Positives = 15/15 (100%), Gaps = 0/15 (0%)

Query  1     HYLGLANKSVKDAMA  15
             HYLGLANKSVKDAMA
Sbjct  1135  HYLGLANKSVKDAMA  1149


>Q99P72.2 RecName: Full=Reticulon-4; AltName: Full=Neurite outgrowth inhibitor;
Short=Nogo protein [Mus musculus]
Length=1162

 Score = 35.0 bits (79),  Expect = 0.001, Method: Composition-based stats.
```

Genome mapping and tools: Read mapping

- Aim: to find coordinates of reads in the reference genome.
- Challenges:
  - Millions of short sequences
  - Sequences are often paired
  - Errors are not randomly distributed
- Most popular programs are bowtie and bwa (both use Burrows-Wheeler Transform algorithm). Two-step approach:
  - Create an index for the reference genome (one time for one genome).
  - Map reads to the reference genome using this index

Genome mapping and tools – overview I

- FastQC
  - FastQC is a quality control application for high throughput sequence data
  - Checks the quality of their sequence data
  - Generates an HTML report

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Genome mapping and tools – overview II

- bowtie
  - The first version of bowtie [Langmead et al. 2009] is optimal for:
    - short reads (under 50 bp)
    - reads without indels (insertions/deletions)
- bowtie2
  - The second version of bowtie2 [Langmead & Salzberg 2012] is optimal for:
    - long reads (more than 50 bp)
    - reads with indels
    - various alignment options
- Each version has its own index file format (bowtie-build / bowtie2-build tools).
- A popular RNA-seq analysis toolset (tophat, cufflinks) is based on bowtie / bowtie2

http://bowtie-bio.sourceforge.net

Genome mapping and tools – overview III

- bwa
  - `bwa backtrack` [Li, Durbin 2009]:
    - for short reads (< 100bp)
  - `bwa bwasw` [Li, Durbin 2010]:
    - for long reads (70bp - 1Mbp)
    - short indels
  - `bwa mem` [Li 2013]:
    - for long reads (70bp - 1Mbp)
    - faster and more efficient

Genome mapping and tools – overview IV

- samtools package - A set of utilities for processing SAM/BAM files
- samtools view
  - convert a bam file into a sam file - `samtools view sample.bam > sample.sam`
  - Convert a sam file into a bam file - `samtools view -bS sample.sam > sample.bam`
  - Extract all the reads aligned to the range specified. An index of the input file is required

```
samtools view -h -b sample_sorted.bam "chr1:10-13" > tiny_sorted.bam
```

- samtools sort

```
samtools sort unsorted_in.bam sorted_out
```

- samtools index

```
samtools index sorted.bam (creates an index file, sorted.bam.bai)
```

http://samtools.sourceforge.net

Genome mapping and tools – overview IV

- samtools package - A set of utilities for processing SAM/BAM files
- samtools view
  - convert a bam file into a sam file - `samtools view sample.bam > sample.sam`
  - Convert a sam file into a bam file - `samtools view -bS sample.sam > sample.bam`
  - Extract all the reads aligned to the range specified. An index of the input file is required

```
samtools view -h -b sample_sorted.bam "chr1:10-13" > tiny_sorted.bam
```

- samtools sort

```
samtools sort unsorted_in.bam sorted_out
```

- samtools index

```
samtools index sorted.bam (creates an index file, sorted.bam.bai)
```

-b: output BAM
-S: read SAM

http://samtools.sourceforge.net

add a proper header

Genome mapping and tools – overview IV

- samtools flagstat – report basic statistics

```
samtools flagstat sample.bam
```

An example of output:

```
4198456 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
4022089 + 0 mapped (95.80%:-nan%)
4198456 + 0 paired in sequencing
2099228 + 0 read1
2099228 + 0 read2
3796446 + 0 properly paired (90.42%:-nan%)
4013692 + 0 with itself and mate mapped
8397 + 0 singletons (0.20%:-nan%)
167574 + 0 with mate mapped to a different chr
72008 + 0 with mate mapped to a different chr (mapQ>=5)
```

- samtools faidx – index a FASTA file

`samtools faidx ref.fasta` (creates an index file ref.fasta.fai)

- samtools merge – merge several BAM files into one

```
samtools merge out.bam in1.bam in2.bam
```

Genome mapping and tools – overview V

- BedTools package
  - `bamtobed` - Convert BAM alignments to BED (& other) formats
  - `bamtofastq` - Convert BAM records to FASTQ records
  - `bedtobam` - Convert intervals to BAM records
  - `closest` - Find the closest, potentially non-overlapping interval
  - `complement` - Extract intervals _not_ represented by an interval file
  - `coverage` - Compute the coverage over defined intervals
  - `genomecov` - Compute the coverage over an entire genome
  - `getfasta` - Use intervals to extract sequences from a FASTA file
  - `intersect` - Find overlapping intervals in various ways
  - `shuffle` - Randomly redistribute intervals in a genome
  - `sort` - Order the intervals in a file

Genome mapping and tools – overview V

`bedtools intersect`

- Report the intervals that represent overlaps between your two files:

```
bedtools intersect -a cpg.bed -b exons.bed
```

- Report the original feature in each file:

```
bedtools intersect -a cpg.bed -b exons.bed -wa -wb
```

- How many base pairs of overlap were there?

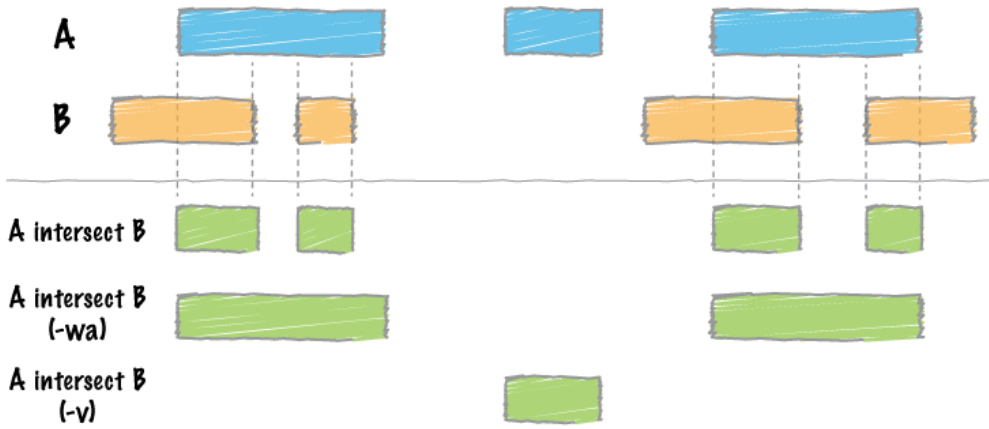
```
bedtools intersect -a cpg.bed -b exons.bed -wo
```

- Counting the number of overlapping features:

```
bedtools intersect -a cpg.bed -b exons.bed -c
```

- Find features that DO NOT overlap:

```
bedtools intersect -a cpg.bed -b exons.bed -v
```



Exercise

- There are paired reads of some DNA sequencing experiment of the human sample:
  `computational-biology-on-carc/data/R1.fastq.gz`
  `computational-biology-on-carc/data/R2.fastq.gz`
- You will study some particular region of the human genome
- Map reads to the human reference genome (version hg19 – find path on our Bio Resources)
- Extract reads that map to your region only
- Upload the reads to UCSC genome browser as a custom track
- count the number of insertions and deletions in SAM file

How To: Mapping

- load bowtie2 program:

```
module purge
module load gcc
module load bowtie2
```

- Copy sequence of a chromosome your region is located at as a FASTA file
  - find the path on our website in Homo sapiens > UCSC > hg19 > Chromosome 21
  - https://carc.usc.edu/user-information/bio-resources/reference-genomes
  - Add `chr*.fa` at the end of the path
  - `ln -s path_above ~/computational-biology-on-carc/results/read-mapping/`
- Map reads to this chromosome using `bowtie2` with the standard parameters

Don't forget to make an index (`bowtie-build2`) of the chromosome before mapping!

- You will get a SAM file as an output, convert to BAM (`samtools view`)
- Count the number of insertions and deletions in SAM file (use `cut` for field 6, and `grep`)

How To: Extracting reads

- load BedTools:

```
module purge
module load gcc
module load bedtools2
```

- Create a tab-delimited BED file with the coordinates of your region:

chr21 10000000 20000000

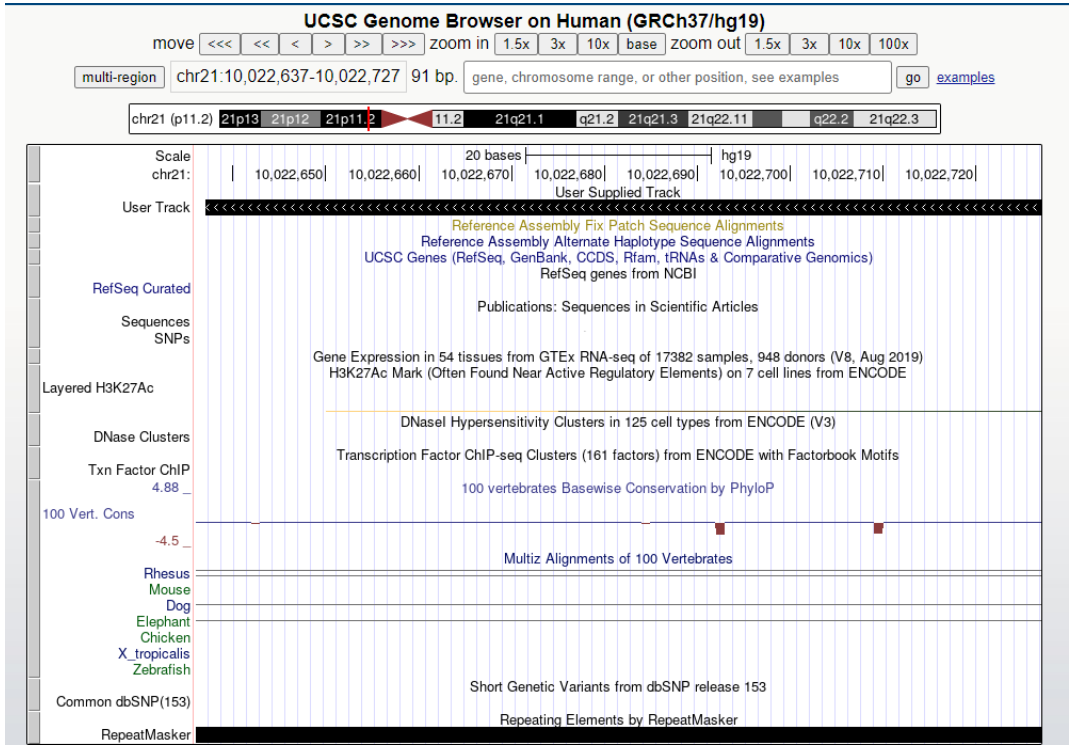
- Convert SAM file with mapped reads to BAM file using `samtools view`
- Use `bedtools intersect` to extract the reads from the BAM file

You'll need a BED file to upload the result to UCSC genome browser, so figure out how to make `bedtools intersect` to produce an output in BED format.

How To: UCSC custom track

- Upload the BED file to UCSC genome browser

‘Add custom track’ button → Choose file → Submit → Go



Resources

- CARC home page
  - https://carc.usc.edu
- Bio Resources at CARC
  - https://carc.usc.edu/user-information/bio-resources
- CARC User Forum
  - https://hpc-discourse.usc.edu/categories
- SLURM tutorials
  - https://slurm.schedmd.com/tutorials.html
- SLURM quick reference
  - https://slurm.schedmd.com/pdfs/summary.pdf

Reference material:

- HPCBio (Holmes J., Clark L., Drnevicch J., Valizadegan N.)
- CNRG (Davidson D., Leigh J.)
- Skoltech (Khrameeva E.)

Mapping exercise: Answer

```
#!/bin/bash
#SBATCH --partition main
#SBATCH --nodes 1
#SBATCH --ntasks 4
#SBATCH --time 01:00:00
#SBATCH --mem 4g
#SBATCH --account=ttrojan_001
#SBATCH --chdir /home1/ttrojan/computational-biology-on-carc
module purge
module load gcc
module load bowtie2
module load samtools
module load bedtools2
mkdir results/read-mapping
cp data/R*.gz results/read-mapping
gunzip results/read-mapping/R1.fastq.gz
gunzip results/read-mapping/R2.fastq.gz
ln -s /project/biodb/genomes/Homo_sapiens/UCSC/hg19/Sequence/Chromosomes/chr21.fa results/read-mapping/
bowtie2-build --threads $SLURM_NTASKS results/read-mapping/chr21.fa results/read-mapping/chr21index
bowtie2 --threads $SLURM_NTASKS -x results/read-mapping/chr21index -q results/read-mapping/R1.fastq > results/read-mapping/R1.sam
bowtie2 --threads $SLURM_NTASKS -x results/read-mapping/chr21index -q results/read-mapping/R2.fastq > results/read-mapping/R2.sam
samtools view results/read-mapping/R1.bam | cut -f 6 | grep -c 'D' > results/read-mapping/R1.no_of_deletions.txt
samtools view results/read-mapping/R1.bam | cut -f 6 | grep -c 'I' > results/read-mapping/R1.no_of_insertions.txt
samtools view results/read-mapping/R2.bam | cut -f 6 | grep -c 'D' > results/read-mapping/R2.no_of_deletions.txt
samtools view results/read-mapping/R2.bam | cut -f 6 | grep -c 'I' > results/read-mapping/R2.no_of_insertions.txt
samtools view -bS results/read-mapping/R1.sam > results/read-mapping/R1.bam
samtools view -bS results/read-mapping/R2.sam > results/read-mapping/R2.bam
samtools sort results/read-mapping/R1.bam > results/read-mapping/R1_sorted.bam
samtools sort results/read-mapping/R2.bam > results/read-mapping/R2_sorted.bam
samtools view -h -b results/read-mapping/R1_sorted.bam "chr21:10000000-20000000" > results/read-mapping/R1_sorted_region.bam
samtools view -h -b results/read-mapping/R2_sorted.bam "chr21:10000000-20000000" > results/read-mapping/R2_sorted_region.bam
bamToBed -i results/read-mapping/R1_sorted_region.bam > results/read-mapping/R1_sorted_region.bed
bamToBed -i results/read-mapping/R2_sorted_region.bam > results/read-mapping/R2_sorted_region.bed
bedtools intersect -a results/read-mapping/R1_sorted_region.bed -b results/read-mapping/R2_sorted_region.bed > results/read-mapping/reads.bed
```