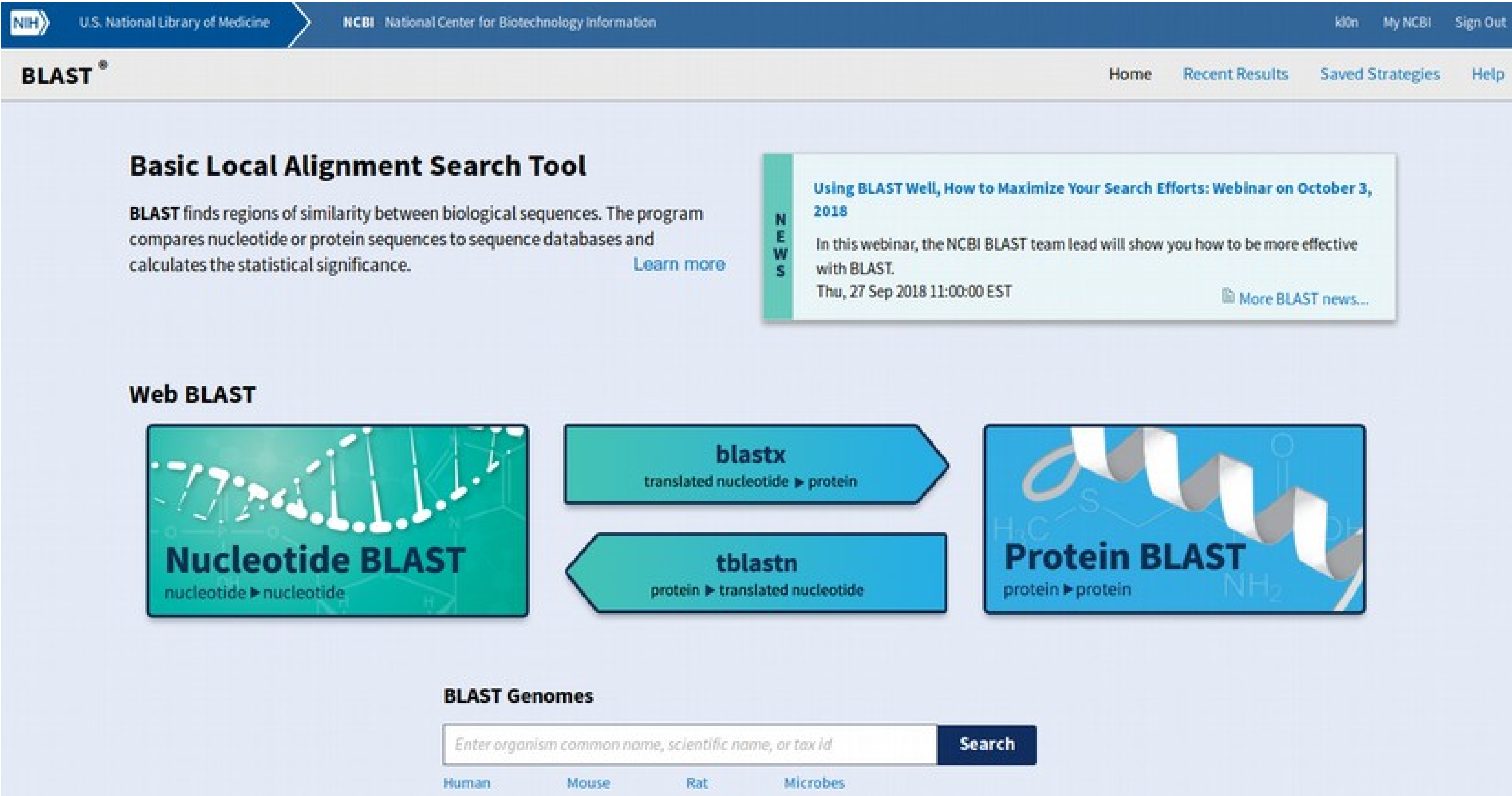
BLAST



NIH
U.S. National Library of Medicine
NCBI National Center for Biotechnology Information
kl0n
My NCBI
Sign Out
BLAST ®
Home
Recent Results
Saved Strategies
Help
NEWS
blastx
translated nucleotide ▶ protein
tblastn
protein ▶ translated nucleotide
Nucleotide BLAST
nucleotide ▶ nucleotide
Protein BLAST
protein ▶ protein
Enter organism common name, scientific name, or tax id
Search
Human
Mouse
Rat
Microbes

Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance. Learn more

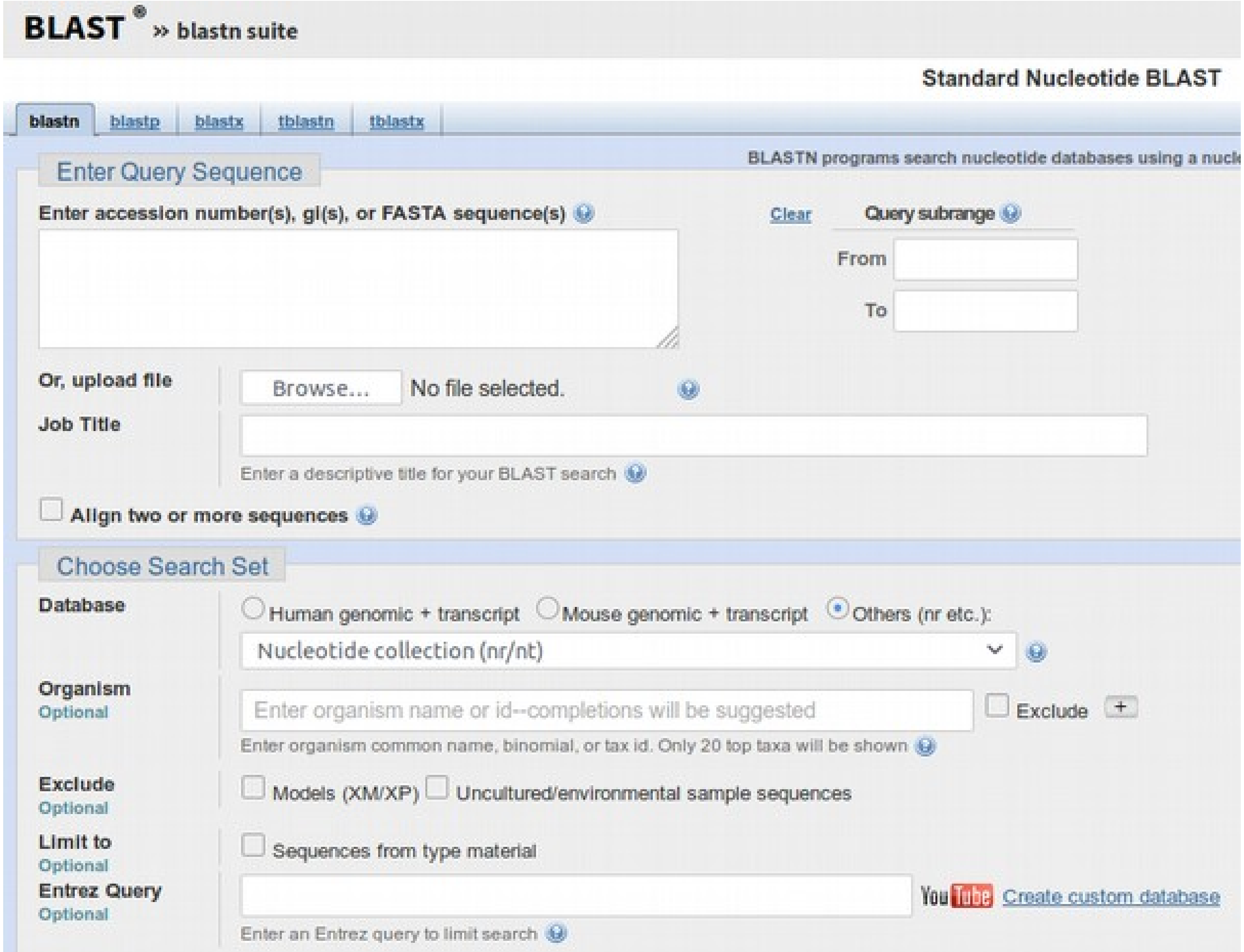
Using BLAST Well, How to Maximize Your Search Efforts: Webinar on October 3, 2018

In this webinar, the NCBI BLAST team lead will show you how to be more effective with BLAST.
Thu, 27 Sep 2018 11:00:00 EST

More BLAST news...

Web BLAST

BLAST Genomes



BLAST® » blastn suite

Standard Nucleotide BLAST

BLASTN programs search nucleotide databases using a nucle

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

Query subrange

From

To

Or, upload file No file selected.

Job Title

Enter a descriptive title for your BLAST search

□ Align two or more sequences

Choose Search Set

Database ○ Human genomic + transcript ○ Mouse genomic + transcript ● Others (nr etc.):

Nucleotide collection (nr/nt)

Organism
Optional

Enter organism name or id--completions will be suggested □ Exclude

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown

Exclude
Optional

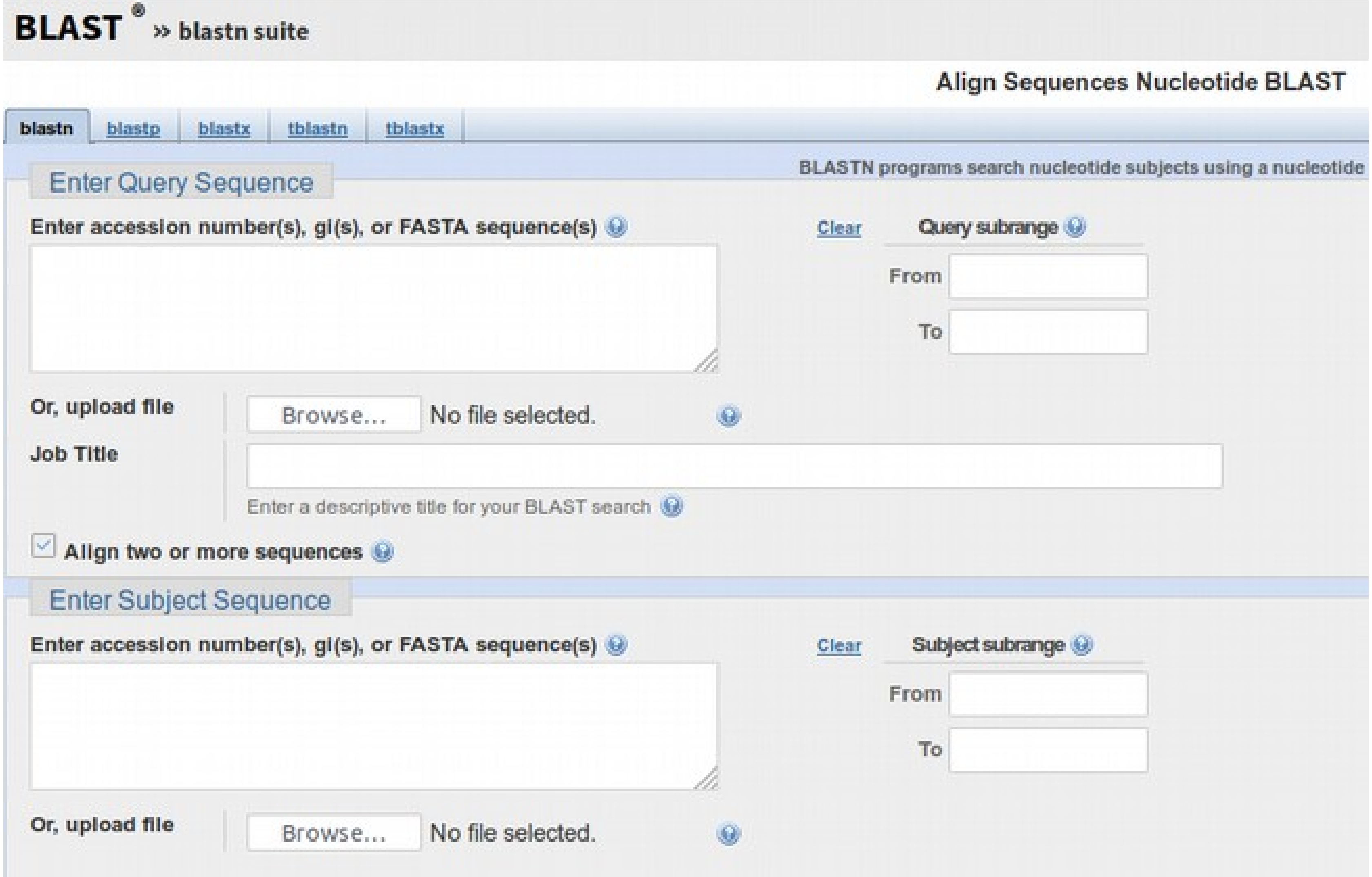
□ Models (XM/XP) □ Uncultured/environmental sample sequences

Limit to
Optional

□ Sequences from type material

Entrez Query
Optional

Enter an Entrez query to limit search



BLAST® » blastn suite

Align Sequences Nucleotide BLAST

BLASTN programs search nucleotide subjects using a nucleotide

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

Query subrange

From

To

Or, upload file No file selected.

Job Title

Enter a descriptive title for your BLAST search

☑ Align two or more sequences

Enter Subject Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

Subject subrange

From

To

Or, upload file No file selected.

Download BLAST

Get BLAST databases and executables

Use BLAST API

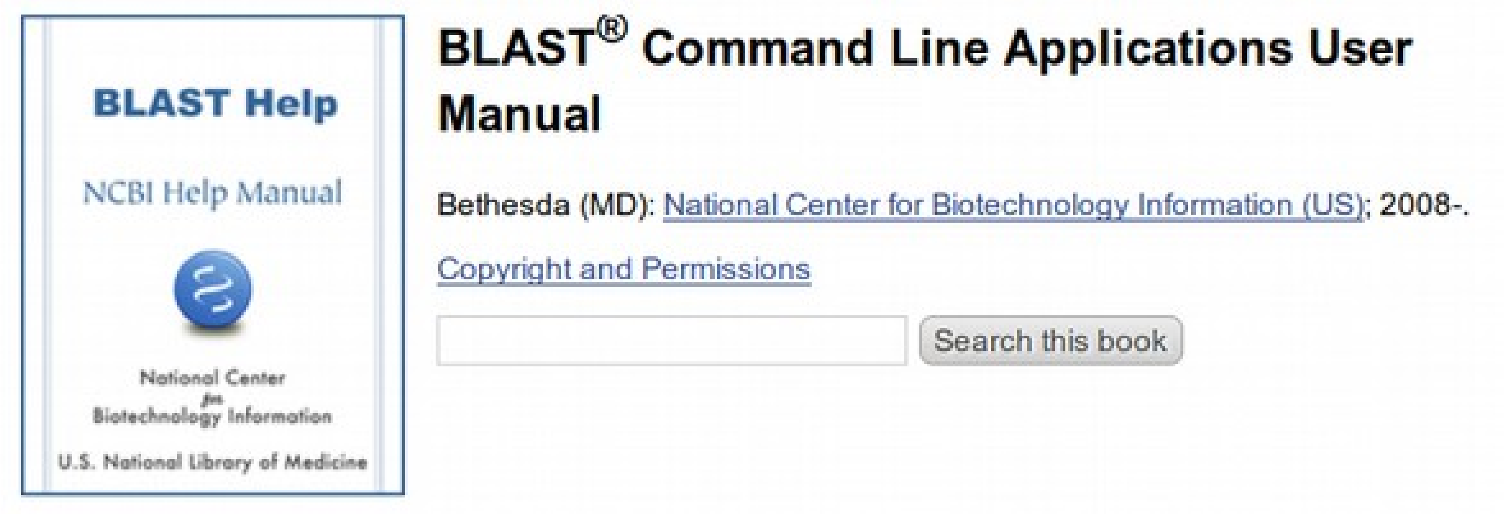
Call BLAST from your application

Use BLAST in the cloud

Start an instance at a cloud provider

(1) ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

(2)



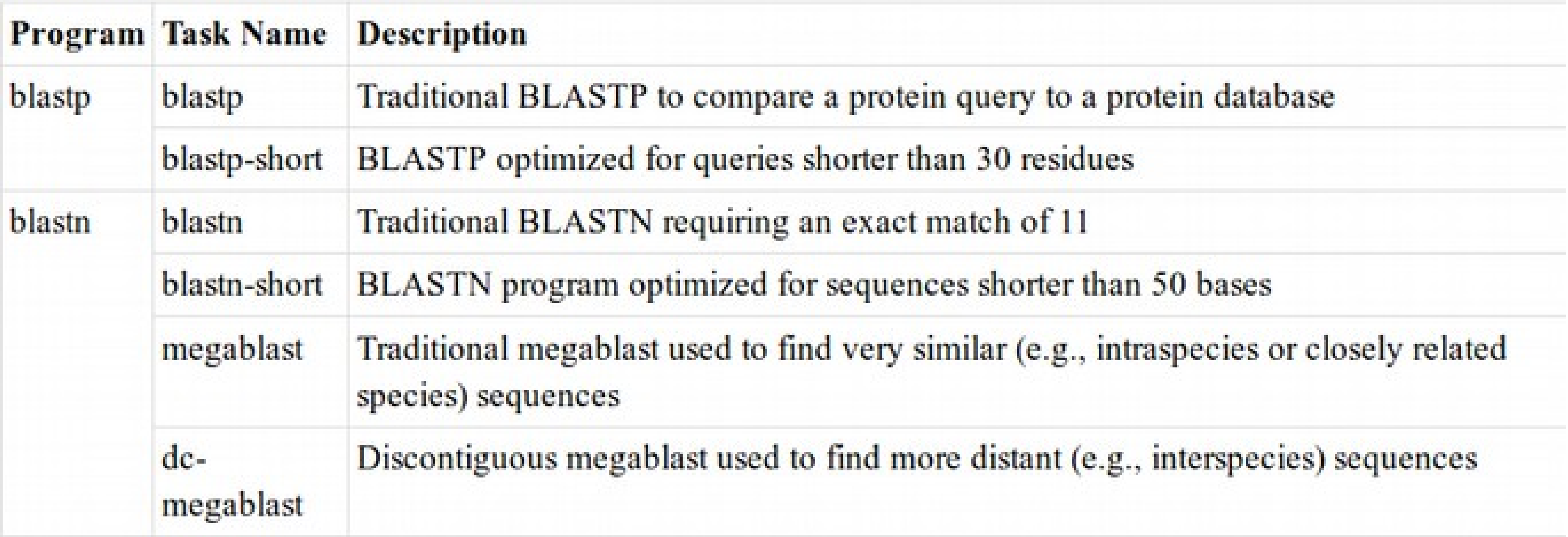
BLAST Help
NCBI Help Manual
National Center for Biotechnology Information
U.S. National Library of Medicine
Search this book

BLAST® Command Line Applications User Manual

Bethesda (MD): National Center for Biotechnology Information (US); 2008-.

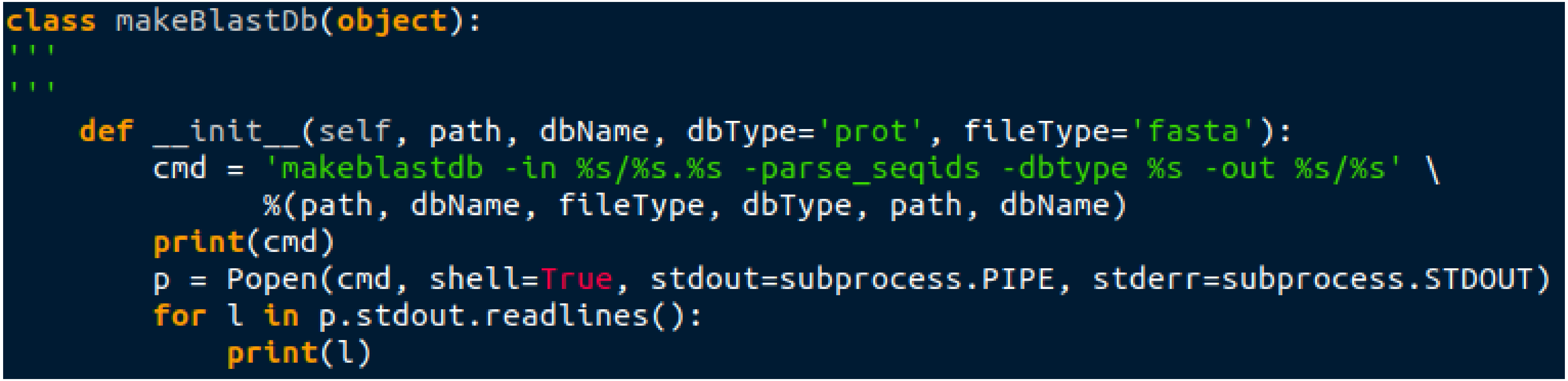
Copyright and Permissions

https://www.ncbi.nlm.nih.gov/books/NBK279668/#usermanual.Custom_output_formats_for_BLA

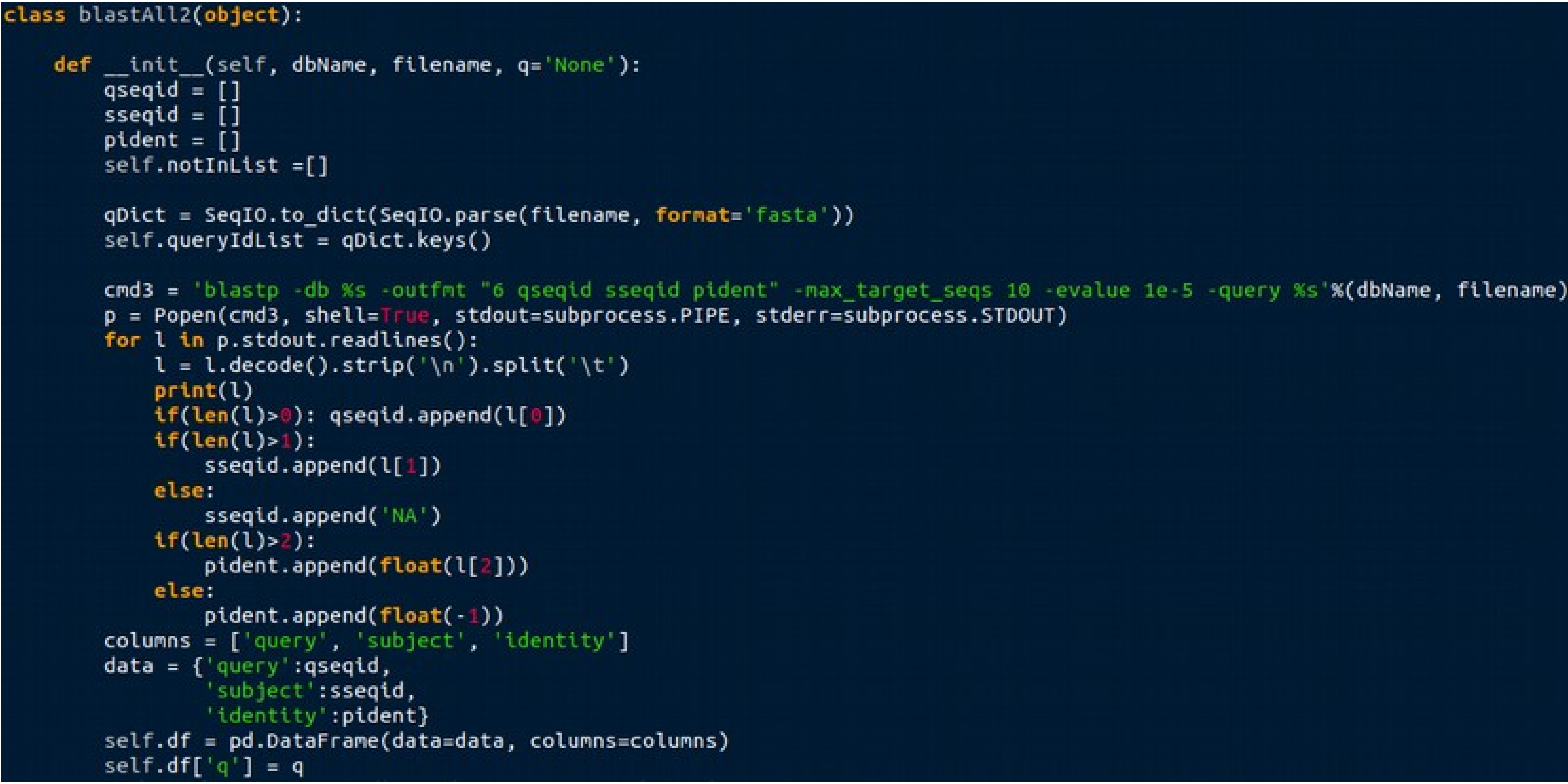


Program	Task Name	Description
blastp	blastp	Traditional BLASTP to compare a protein query to a protein database
	blastp-short	BLASTP optimized for queries shorter than 30 residues
blastn	blastn	Traditional BLASTN requiring an exact match of 11
	blastn-short	BLASTN program optimized for sequences shorter than 50 bases
	megablast	Traditional megablast used to find very similar (e.g., intraspecies or closely related species) sequences
	dc-megablast	Discontiguous megablast used to find more distant (e.g., interspecies) sequences

Bir grup FASTA sekansından yeni BLAST veritabanı yapmak için:



```
class makeBlastDb(object):
'''
'''
    def __init__(self, path, dbName, dbType='prot', fileType='fasta'):
        cmd = 'makeblastdb -in %s/%s.%s -parse_seqids -dbtype %s -out %s/%s' \
              %(path, dbName, fileType, dbType, path, dbName)
        print(cmd)
        p = Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
        for l in p.stdout.readlines():
            print(l)
```



```
class blastAll2(object):

    def __init__(self, dbName, filename, q='None'):
        qseqid = []
        sseqid = []
        pident = []
        self.notInList =[]

        qDict = SeqIO.to_dict(SeqIO.parse(filename, format='fasta'))
        self.queryIdList = qDict.keys()

        cmd3 = 'blastp -db %s -outfmt "6 qseqid sseqid pident" -max_target_seqs 10 -evalue 1e-5 -query %s'%(dbName, filename)
        p = Popen(cmd3, shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
        for l in p.stdout.readlines():
            l = l.decode().strip('\n').split('\t')
            print(l)
            if(len(l)>0): qseqid.append(l[0])
            if(len(l)>1):
                sseqid.append(l[1])
            else:
                sseqid.append('NA')
            if(len(l)>2):
                pident.append(float(l[2]))
            else:
                pident.append(float(-1))
        columns = ['query', 'subject', 'identity']
        data = {'query':qseqid,
                'subject':sseqid,
                'identity':pident}
        self.df = pd.DataFrame(data=data, columns=columns)
        self.df['q'] = q
```