How to Run NCBI BLAST on zcluser at GACRC

**BLAST:** Basic Local Alignment Search Tool

Georgia Advanced Computing Resource Center
University of Georgia
Suchitra Pakala
pakala@uga.edu
OVERVIEW

• What is BLAST? Why use it? How does it work?
• Different BLAST programs and Running BLAST
• Running on zcluster and other considerations
Part 1:

What is BLAST?

Why use it?

How does it work?
Introduction to BLAST

• **BLAST**: Basic Local Alignment Search Tool
• Rapid sequence comparison of a “query” sequence against a large set of sequences – the “reference database”
• “Local” means it searches and aligns sequence segments, rather than aligning the entire sequence
• *Defacto* standard for search and alignment
• Fast, accurate, open source, available on several platforms
BLAST - Uses

• Identify a species
• Identify unknown sequences
• Identify homologous sequences in a database
• Find domains and hence determine protein function
• Phylogenetic analysis
• Placing sequences on chromosome
• Genome annotation
• Find variants of genes
BLAST – How does it work?

• Initial short matches – “words”
• Build local alignments in regions with common words – “seeding”
  • Seed extensions
• Scoring matrix
• Alignment score; thresholds

Query sequence: PQGEFG

Database sequence: DP P E G V V

Exact match is scanned.

Score: -2 7 7 2 6 1 -1

HSP

Optimal accumulated score = 7 + 7 + 2 + 6 + 1 = 23
Part 2:

BLAST Programs

Reference Databases

Running BLAST
BLAST Programs

• blastn (nucleotide BLAST)
• blastp (protein BLAST)
• blastx (translated BLAST)
• tblastn (translated BLAST)
• tblastx (translated BLAST)
**blastn**  
AGTCCGATGA  
AGCGCAAGTCAAGTCCGATGAGCTAGCAGCT

**blastp**  
FLPQLLGNVL  
MFIPFIFLPQLLGNVLVCVLAHHFGKEFTP

**blastx**  
TGGCGAACCTGGCTGCTGGATG  
\[\text{TGG-CA-AC-TT-GCT-GCT-GGA-TG}\]
\[\text{T-GGC-GAA-CCT-TGG-CTG-CTG-GAT-G}\]
\[\text{T-GCG-AAC-CTT-GGC-TGC-TGG-ATG}\]
\[\text{MANLGWMLVLFLVATWSDLGLCCKRP}\]

WRTLAAG  
GEPWLLD  
ANLGCWM  
3 more frames from reverse

NOT shown here
### Number of Searches Performed

<table>
<thead>
<tr>
<th>Program</th>
<th>Input</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>blastn</td>
<td>DNA</td>
<td>DNA</td>
</tr>
<tr>
<td>blastp</td>
<td>protein</td>
<td>protein</td>
</tr>
<tr>
<td>blastx</td>
<td>DNA</td>
<td>protein</td>
</tr>
<tr>
<td>tblastn</td>
<td>protein</td>
<td>DNA</td>
</tr>
<tr>
<td>tblastx</td>
<td>DNA</td>
<td>DNA</td>
</tr>
</tbody>
</table>

**DNA** => 6-frame translation
BLAST Databases

• Publicly available collections of reference sequences and databases
  • nr/nt - non-redundant nucleotide collection
  • Refseq proteins
  • Species specific
    • Human
    • Mouse

• Custom databases
  • Build your own!
  • Example: Fungal ITS
Creating your own BLAST reference database

```
makeblastdb -in my_genomes.fasta -dbtype nucl -out my_genomes_db -title my_genomes_db
```

- Index files are created:

  - my_genomes_db.nhr
  - my_genomes_db.nin
  - my_genomes_db.nog
  - my_genomes_db.nsd
  - my_genomes_db.nii
  - my_genomes_db.nsq
Run a BLAST Search – Where?

- NCBI BLAST server – web based searches

- Local infrastructure – zcluster

- Personal computer – yes, you can do it on your laptop
  - Integrated bioinformatics workbenches (CLC Genomics Workbench, etc)
  - Command line
    - Download the executables that are readily available for your machine (windows / mac)
    - Installation instructions in [manual](#)
    - Obtain readily available reference databases AND/OR Build your own
Run a BLAST search – How?

• 4 “simple” steps:
  • Provide “Query” sequence
  • Select the BLAST program
  • Choose the database to search
  • Choose optional parameters

• “BLAST” away
STEP1: Choose the Sequence

- Sequence can be input in FASTA format or an accession number

Example of the FASTA format for a BLAST query:

```
>gi|4504349|ref|NP_000509.1| hemoglobin subunit beta [Homo sapiens]
MVHLTPKEKSAVTALGKVNVDEVGGEALGKLVVYPWTQRFFESFGDLSTPDVGMDPKAKHKVVLGAFSDGLAHDLNKSTFTATLSELHCDKLHDPFENFLLGHLVCLAHFFGKEFTPVQAAAYKVVAMVANLALKYTH
```
STEP 2: Choose the BLAST Program

BLAST Assembled Genomes
Choose a species genome to search, or list all genomic BLAST databases:
- Human
- Mouse
- Rat
- Arabidopsis thaliana
- Oryza sativa
- Bos taurus
- Danio rerio
- Drosophila melanogaster
- Gallus gallus
- Pan troglodytes
- Microbes
- Apis mellifera

Basic BLAST
Choose a BLAST program to run.

- **nucleotide blast**
- **protein blast**
- **blastx**
- **tblastn**
- **tblastx**

Search a nucleotide database using a nucleotide query
*Algorithms:* blastn, megablast, discontiguous megablast

Search protein database using a protein query
*Algorithms:* blastp, psi-blast, phi-blast

Search protein database using a translated nucleotide query

Search translated nucleotide database using a protein query

Search translated nucleotide database using a translated nucleotide query

Specialized BLAST
Choose a type of specialized search (or database name in parentheses.)
- Search trace archives
- Find conserved domains in your sequence (cds)
- Find sequences with similar conserved domain architecture (cdart)
- Search sequences that have gene expression profiles (GEO)
- Search immunoglobulins (IgBLAST)
- Search for SNPs (snp)
- Screen sequence for vector contamination (veccsreen)
- **Align** two sequences using BLAST (bl2seq)
Step 3: Choose the database

nr = non-redundant (most general database)
dbest = database of expressed sequence tags
dbsts = database of sequence tag sites
gss = genomic survey sequences
Step 4: Choose Optional Parameters

You can....

• Choose the organism to search
• Turn filtering on/off
• Change the substitution matrix
• Change the expect (e) value
• Change the word size
• Change the output format
Step 4: Select optional search parameters

- Organism
- Algorithm
Step 4: optional blastp search parameters

**Algorithm parameters**

**General Parameters**
- **Max target sequences**: 100
- **Short queries**: Automatically adjust parameters for short input sequences
- **Expect threshold**: 10
- **Word size**: 3
- **Max matches in a query range**: 0

**Scoring Parameters**
- **Matrix**: BLOSUM62
- **Gap Costs**: Existence: 11 Extension: 1
- **Compositional adjustments**: Conditional compositional score matrix adjustment

**Filters and Masking**
- **Filter**: Low complexity regions
- **Mask**: Mask for lookup table only, Mask lower case letters

**BLAST**
Search database Non-redundant protein sequences (nr) using Blastp
- **Show results in a new window**
### BLAST Output

#### Web interface – Summary of Results

<table>
<thead>
<tr>
<th>Description</th>
<th>Max score</th>
<th>Total score</th>
<th>Query cover</th>
<th>E value</th>
<th>Identi</th>
<th>Accession</th>
</tr>
</thead>
<tbody>
<tr>
<td>unnamed protein product [Homo sapiens]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>2e-25</td>
<td>100%</td>
<td>BAG58727.1</td>
</tr>
<tr>
<td>PREDICTED: major prion protein isoform X7 [Mandrillus leucophaeus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>2e-25</td>
<td>100%</td>
<td>XP_011830622.1</td>
</tr>
<tr>
<td>PREDICTED: major prion protein isoform X8 [Colobus angolensis palliatus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>2e-25</td>
<td>100%</td>
<td>XP_011796437.1</td>
</tr>
<tr>
<td>prion protein precursor [Microcebus murinus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>3e-25</td>
<td>96%</td>
<td>NP_001296849.1</td>
</tr>
<tr>
<td>PREDICTED: major prion protein isoform X5 [Mandrillus leucophaeus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>3e-25</td>
<td>100%</td>
<td>XP_011830615.1</td>
</tr>
<tr>
<td>PREDICTED: major prion protein isoform X6 [Colobus angolensis palliatus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>3e-25</td>
<td>100%</td>
<td>XP_011796435.1</td>
</tr>
<tr>
<td>PREDICTED: major prion protein isoform X4 [Colobus angolensis palliatus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>4e-25</td>
<td>100%</td>
<td>XP_011796433.1</td>
</tr>
<tr>
<td>PREDICTED: major prion protein isoform X3 [Mandrillus leucophaeus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>4e-25</td>
<td>100%</td>
<td>XP_011830601.1</td>
</tr>
<tr>
<td>prion protein [Galeopterus variegatus]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>5e-25</td>
<td>94%</td>
<td>AAN16488.1</td>
</tr>
<tr>
<td>prion protein (p27-30) (Creutzfeld-Jakob disease, Gerstmann-Strausler-Scheinker syndrome)</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>5e-25</td>
<td>100%</td>
<td>AAV38283.1</td>
</tr>
<tr>
<td>prion protein [Homo sapiens]</td>
<td>105</td>
<td>105</td>
<td>49%</td>
<td>5e-25</td>
<td>100%</td>
<td>CAA58442.1</td>
</tr>
</tbody>
</table>
### BLAST Output

#### Web interface – Alignments

#### Unnamed protein product [Homo sapiens]

- **Sequence ID:** dbj[BAG58727.1]
- **Length:** 229
- **Number of Matches:** 1

<table>
<thead>
<tr>
<th>Range</th>
<th>Score</th>
<th>Expect Method</th>
<th>Identities</th>
<th>Positives</th>
<th>Gaps</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 49</td>
<td>105 bits(263)</td>
<td>2e-25</td>
<td>Compositional matrix adjust.</td>
<td>49/49(100%)</td>
<td>0/49(0%)</td>
<td>+1</td>
</tr>
</tbody>
</table>

**Query**: MANL6CW/LVL/VTWSDLG/CCKXR/PKGwNTGSG/RYPQGSPGGNRY

**Subject**: MANL6CW/LVL/VTWSDLG/CCKXR/PKGwNTGSG/RYPQGSPGGNRY

#### Predicted major prion protein isoform X7 [Mandrillus leucophaeus]

- **Sequence ID:** refXP_011830622.1
- **Length:** 229
- **Number of Matches:** 1

<table>
<thead>
<tr>
<th>Range</th>
<th>Score</th>
<th>Expect Method</th>
<th>Identities</th>
<th>Positives</th>
<th>Gaps</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 49</td>
<td>105 bits(263)</td>
<td>2e-25</td>
<td>Compositional matrix adjust.</td>
<td>49/49(100%)</td>
<td>0/49(0%)</td>
<td>+1</td>
</tr>
</tbody>
</table>

**Query**: MANL6CW/LVL/VTWSDLG/CCKXR/PKGwNTGSG/RYPQGSPGGNRY

**Subject**: MANL6CW/LVL/VTWSDLG/CCKXR/PKGwNTGSG/RYPQGSPGGNRY
In Summary

- Key results
  - Score
  - Query Coverage
  - Expected value (evalue)
  - % Identities

- HSPs – High Scoring Segment Pairs
Part 3:

Running on zcluster

And other considerations
zcluster – General Information

- Batch-queueing System:
  - Jobs can be started (submitted), monitored, and controlled
  - Determine which compute node is the best place to run a job
  - Determine appropriate execution priority for a job to run

- On zcluster: Sun Grid Engine (SGE)
## zcluster – Computing Resources

<table>
<thead>
<tr>
<th>Queue Type</th>
<th>Queue Name</th>
<th>Nodes</th>
<th>Processor</th>
<th>Cores/Node</th>
<th>RAM(GB)/Node</th>
<th>Cores</th>
<th>NVIDIA GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>rcc-30d</td>
<td>45</td>
<td>Intel Xeon</td>
<td>12</td>
<td>48</td>
<td>540</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>150</td>
<td></td>
<td>8</td>
<td>16</td>
<td>1200</td>
<td></td>
</tr>
<tr>
<td>High Memory</td>
<td>rcc-m128-30d</td>
<td>1</td>
<td>Intel Xeon</td>
<td>8</td>
<td>128</td>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>8</td>
<td>192</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td>12</td>
<td>256</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rcc-m512-30d</td>
<td>2</td>
<td></td>
<td>32</td>
<td>512</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Multi Core</td>
<td>rcc-mc-30d</td>
<td>6</td>
<td>AMD Opteron</td>
<td>32</td>
<td>64</td>
<td>192</td>
<td>N/A</td>
</tr>
<tr>
<td>Interactive</td>
<td>interq</td>
<td>2</td>
<td>AMD Opteron</td>
<td>48</td>
<td>132</td>
<td>96</td>
<td>N/A</td>
</tr>
<tr>
<td>GPU</td>
<td>rcc-sgpu-30d</td>
<td>2</td>
<td>Intel Xeon</td>
<td>8</td>
<td>48</td>
<td>16</td>
<td>4 Tesla S1070 cards</td>
</tr>
<tr>
<td></td>
<td>rcc-mgpu-30d</td>
<td>2</td>
<td></td>
<td>12</td>
<td>48</td>
<td>24</td>
<td>9 Tesla (Fermi) M2070 cards</td>
</tr>
<tr>
<td></td>
<td>rcc-kgpu-30d</td>
<td>4</td>
<td></td>
<td>12</td>
<td>96</td>
<td>24</td>
<td>32 Tesla (Kepler) K20Xm cards</td>
</tr>
</tbody>
</table>

**Total peak performance:** 23 Tflops
So, Where to run the BLAST job?

• Directly on login node of zcluster – **NO. Please don’t.**

• After login to zcluster, Submit to the queue. **rcc-30d** is recommended
How does the command look like?

```
blastx -query my_input.fasta
        -db nt
        -out my_input_blastx_nr.out
        -num_threads 4
        -outfmt 6
        -evalue 1e-5
        -max_target_seqs=2
```
• Write a shell script [blast_AF293.sh]

```bash
#!/bin/bash
cd /home/gacrc-instruction/pakala/Blast/AF293
time /usr/local/ncbiblast+/latest/bin/blastn -num_threads 4 \
egquery AFU293.fna -db /db/ncbiblast/latest/nt -out blast_afu.out \
egoutfmt 6 -evalue 1e-5 -max_target_seqs=2
```

• Submit to queue

```bash
qsub -q rcc-30d -pe thread 4 ./blast_AF293.sh
```

• Monitor job

```bash
qstat, qacct, qsj, qdel
```
To check the status of all queued and running jobs: `qstat`

- `qstat` shows your job in the pool
- `qstat -u "*"` shows all the jobs in the pool
- `qstat -j 12345` shows detailed information, e.g., `maxvmem`, about the job with JOBID 12345
- `qstat -g t` lists all nodes used by your jobs

To cancel a queued or running job: `qdel`

- `qdel -u pakala` deletes all your jobs
- `qdel 12345` deletes your job with JOBID 12345

To list detailed information about a job: `qsj`, `qacct`

- `qsj 12345` shows information, e.g., `maxvmem`, about the RUNNING job with JOBID 12345
- `qacct -j 12345` shows information, e.g., `maxvmem`, about the ENDED job with JOBID 12345
### The best part: The Results!!!

<table>
<thead>
<tr>
<th>Qseqid</th>
<th>Sseqid</th>
<th>Pident</th>
<th>Length</th>
<th>Mismatch</th>
<th>Gapopen</th>
<th>Qstart</th>
<th>Qend</th>
<th>Sstart</th>
<th>Send</th>
<th>Evalue</th>
<th>Bitscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch334</td>
<td>gi</td>
<td>56</td>
<td>ref</td>
<td>NP_757.1</td>
<td>17.42</td>
<td>597</td>
<td>364</td>
<td>26</td>
<td>33168</td>
<td>33694</td>
<td>741  1278</td>
</tr>
<tr>
<td>ch423</td>
<td>gi</td>
<td>60</td>
<td>ref</td>
<td>WP_500.1</td>
<td>50.85</td>
<td>118</td>
<td>25</td>
<td>2</td>
<td>11856</td>
<td>11973</td>
<td>580  664</td>
</tr>
<tr>
<td>ch425</td>
<td>gi</td>
<td>60</td>
<td>ref</td>
<td>WP_500.1</td>
<td>55.05</td>
<td>109</td>
<td>28</td>
<td>3</td>
<td>11863</td>
<td>11971</td>
<td>809  896</td>
</tr>
</tbody>
</table>

### Alignment view options

- `-outfmt <String>`
  - 0: pairwise
  - 1: query-anchored showing identities
  - 2: query-anchored no identities
  - 3: flat query-anchored, show identities
  - 4: flat query-anchored, no identities
  - 5: XML Blast output
  - 6: **tabular**
  - 7: tabular with comment lines
  - 8: Text ASN.1
  - 9: Binary ASN.1
  - 10: Comma-separated values
  - 11: BLAST archive format (ASN.1)
Other parameters

• There are a LOT of parameters that can be adjusted

• Beyond the scope of this introduction

• Example:

*** Restrict search or results
-gilist <String>
  Restrict search of database to list of GI's
  * Incompatible with: negative_gilist, seqidlist, remote, subject, subject_loc
-seqidlist <String>
  Restrict search of database to list of SeqId's
  * Incompatible with: gilist, negative_gilist, remote, subject, subject_loc
-negative_gilist <String>
  Restrict search of database to everything except the listed GIs
  * Incompatible with: gilist, seqidlist, remote, subject, subject_loc

-max_target_seqs <Integer, >=1>
  Maximum number of aligned sequences to keep
  Not applicable for outfmt <= 4
What about “LARGE” jobs?

RCCBatchBlastPlus

• rccbatchblastplus
  • Splits big query file into smaller units
  • Submits each unit to queue to BLAST

  qlogin
cd working_directory
rccbatchblastplus blastn –query inputfile –out outputfile -db targetdatabase
–size size-of-split-unit –queue queue-name

• rccbatchblastplus-check
  • Run after rccbatchblastplus finishes
  • Checks the results and merges them into one final output file

  qlogin
rccbatchblastplus-check

• https://wiki.gacrc.uga.edu/wiki/RCCBatchBlastPlus
Another option: **Mpiblast**

- Open-source, parallel implementation of NCBI BLAST
- Takes advantage of distributed computational resources, i.e., a cluster, through explicit MPI communication
- Utilizes all available resources at cluster

```
mpifformatdb
mpirun
mpiblast
```

- [https://wiki.gacrc.uga.edu/wiki/Mpiblast](https://wiki.gacrc.uga.edu/wiki/Mpiblast)
Quick links

• NCBI Blast+
  https://wiki.gacrc.uga.edu/wiki/NCBI_Blast_%2B

• Legacy Blast
  https://wiki.gacrc.uga.edu/wiki/NCBI_Blast

• Batch Blast
  https://wiki.gacrc.uga.edu/wiki/RCCBatchBlastPlus

• MPI Blast
  https://wiki.gacrc.uga.edu/wiki/Mpiblast

• Common Blast databases
  https://wiki.gacrc.uga.edu/wiki/BLAST_Databases