BLAST in the Cloud.

Tom Madden, PhD

U.S. National Library of Medicine
National Center for Biotechnology Information

8600 Rockville Pike, Bethesda MD, 20894 USA

ncbi.nlm.nih.gov

Background

The Basic Local Alignment Search Tool (BLAST) [1, 2] is a very popular tool for searching and aligning sequences. BLAST workloads often come in bursts, with a researcher wanting to search a large number of sequences from a project and needing the results as soon as possible to enable further analysis. The cloud is an ideal solution to this problem. Users can order up a large number of servers very quickly and distribute their BLAST searches over those machines. Therefore, users are not constricted by the resources available on their local infrastructure or buying compute power that sits idle most of the time.

Also, users want to be able to run their bioinformatics pipeline regardless on the host environment, which is why NCBI is now providing a Docker© version of BLAST.

Common Workflow Language (CWL) [3] examples are also provided in order to encourage pipeline builders to use a formal approach that can improve the FAIRness [4] of the resulting pipeline.

Packaging

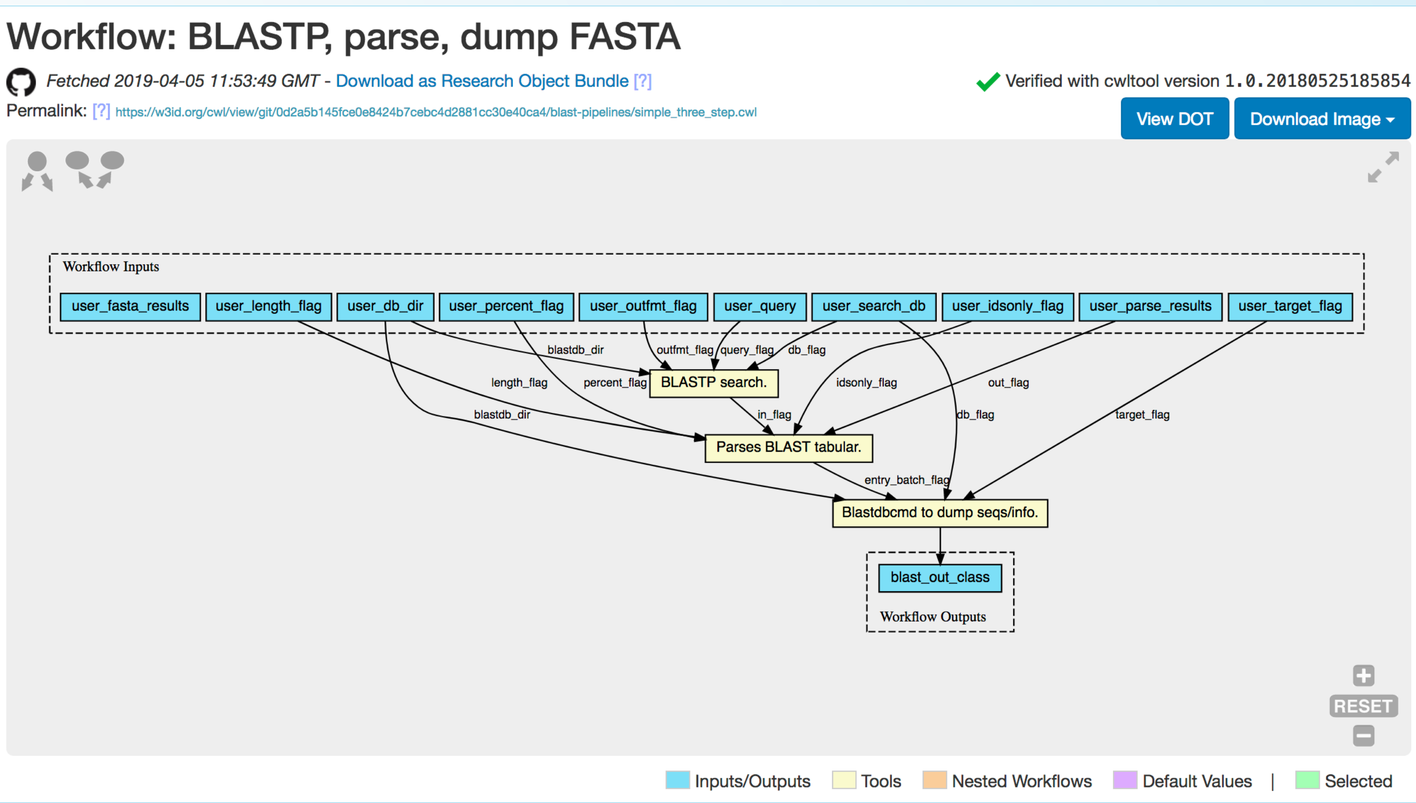
BLAST+ executables* and Edirect utilities in Docker.

Databases hosted and transparently downloaded from GCS (S3 coming).

* BLAST[NPX], TBLAST[NX],RPSBLAST, RPSTBLASTN, MAKEBLASTDB, BLASTDBCMD, Magic-BLAST

BLAST databases hosted on cloud provider

Database	Letters	Type
NR – non-redundant proteins	62 billion	protein
Refseq Protein	45 billion	protein
UniProtKB/Swiss-prot	177 million	protein
PDB – proteins with structures	26 million	protein
Landmark - Proteomes from 27 diverse organisms	234 million	protein
NT – nucleotide collection	198 billion	nucleotide
Refseq RNA	57 billion	nucleotide
16S ribosomal sequences from Archaea and Bacteria	30 million	nucleotide
Human genome	3.3 billion	nucleotide
Mouse genome	2.8 billion	nucleotide
Viral Refseq genomes	275 million	nucleotide



Visualization of a simple CWL [3] workflow. The workflow performs a BLASTP search, parses the output with a python script to find alignments longer than 100 residues and more than 40% identity, and produces FASTA output for those database sequences. The visualization was produced by CWLViewer [5]. The automated production of this graph demonstrates that a CWL workflow is machine readable, an important requirement for interoperability.

CWL Workflow

```
cwlVersion: v1.0
class: Workflow
label: BLASTP, parse, dump FASTA
inputs:
  user_db_dir: Directory?
  user_query: File
  user_search_db:   string
  user_parse_results: string
  user_outfmt_flag: string
  user_length_flag: int
  user_percent_flag: float
  user_idsonly_flag: int
  user_fasta_results: File
  user_target_flag: boolean?

outputs:
  blast_out_class:
    type: File
    outputSource: blastdbcmd_step/blastdbcmd_results

steps:
    blast_step:
      run: blastp_docker.cwl
      in:
         query_flag: user_query
         db_flag: user_search_db
         blastdb_dir: user_db_dir
         outfmt_flag: user_outfmt_flag
      out : [blast_results]
    parse_blast_step:
      run: parse_blast_report.cwl
      in:
        length_flag: user_length_flag
        percent_flag: user_percent_flag
        idsonly_flag: user_idsonly_flag
        in_flag: blast_step/blast_results
        out_flag: user_parse_results
      out: [parse_blast_results]
    blastdbcmd_step:
      run: blastdbcmd_docker.cwl
      in:
        entry_batch_flag: parse_blast_step/parse_blast_results
        db_flag: user_search_db
        blastdb_dir: user_db_dir
        target_flag: user_target_flag
      out: [blastdbcmd_results]
```

Excerpt from the visualized workflow. The workflow is described under “steps”. First, “blast_step” runs BLASTP. Second, “parse_blast_step” parses the tabular BLAST output with a python script. Finally, blastdbcmd produces FASTA output for the database sequences. Each step uses a reusable module that performs one task.

Availability & Resources

BLAST Docker: https://github.com/ncbi/docker/blob/master/blast/README.md

NCBI Workbench Docker: https://github.com/ncbi/docker/tree/master/ncbi-workbench

Sample CWL workflows: https://github.com/ncbi/cwl-demos

About BLAST and CWL: https://github.com/ncbi/cwl-demos/blob/master/blast-pipelines/README.md

BLAST home page: https://blast.ncbi.nlm.nih.gov/

CWL Resources: https://www.commonwl.org

CWL Viewer https://view.commonwl.org/

References

[1] SF Altschul et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997 Sep 1;25(17):3389-402.

[2] C Camacho et al. BLAST+: architecture and applications. BMC Bioinformatics. 2009 Dec 15;10:421. doi: 10.1186/1471-2105-10-421.

[3] Peter Amstutz et al. (2016): **Common Workflow Language, v1.0**. Specification, *Common Workflow Language working group*. https://w3id.org/cwl/v1.0/doi:10.6084/m9.figshare.3115156.v2

[4] MD Wilkinson et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016 Mar 15;3:160018. doi: 10.1038/sdata.2016.18.

[5] https://doi.org/10.7490/f1000research.1114375.1

Acknowledgements

Christiam Camacho, Dave Arndt, Amelia Fong, Yuri Merezhuk, Yan Raytselis, Eugene Yaschenko, Andrew Johnson, Valerie Schneider, Steve Sherry. This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

Need to reach us? Go to support.nlm.nih.gov.

ncbi.nlm @ncbi NCBI linkd.in/1gizCtx