Jun 30, 2025 Version 5

Annotation for Fungi V.5

Microbiology Resource Announcements

DOI
https://dx.doi.org/10.17504/protocols.io.e6nvw14nwlmk/v5

Sebastian Bassi[1], Virginia Gonzalez[1], Tristan Yang[2]

[1]Toyoko; [2]Keck Graduate Institute

Sebastian Bassi
Toyoko

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.e6nvw14nwlmk/v5

External link: https://doi.org/10.1128/mra.00936-24

Protocol Citation: Sebastian Bassi, Virginia Gonzalez, Tristan Yang 2025. Annotation for Fungi. **protocols.io** https://dx.doi.org/10.17504/protocols.io.e6nvw14nwlmk/v5Version created by **Sebastian Bassi**

Manuscript citation:
Yang T, Certano A, Lakshmanan V, Bassi S, Purushotham N, Nock T, Chaudhury A, Ray A Genome resource announcement of a Leptodophora sp. fungus isolated from roots of broadleaf plants in Wisconsin, USA. Microbiology Resource Announcements 14(8). doi: 10.1128/mra.00936-24

Protocol status: Working
We use this protocol and it's working

Created: June 29, 2025

Last Modified: June 30, 2025

Protocol Integer ID: 221248

Keywords: docker, bioinformatics, dnalinux, fungi, annotation for fungi protocol, fungi genome, fungi protocol, annotation, genome

Abstract

Protocol to annotate a fungi genome from long and short reads

Setup

1 Install Docker

If you don't have Docker already, install it. There are two versions, Docker Engine (also known as CE) and Docker Desktop. The Desktop version is more user friendly but since may require commercial license for large enterprise, this tutorial is based on the Docker engine. Both version will work in this protocol. Linux users can install both Docker CE and Desktop, while macOS and Windows users should install Docker Desktop.

Follow the installation instructions from **https://docs.docker.com/engine/install/**

2 Get your data ready

You will need fastq data (long reads), short reads, and the assembly data. In the following code, the assembly data file is called *assembly.fasta*. The long reads file is called *ID.fastq*. The short reads should be two files (*ID_R1.fastq.gz* and *ID_R2.fastq.gz*). If you have more files for short reads, you can concatenate them so you end up with 2 files. For example, if you have *ID_L001_R1.fastq.gz*, *ID_L002_R1.fastq.gz*, *ID_L001_R2.fastq.gz*, *ID_L002_R2.fastq.gz*, you can concatenate them with these commands:

```
cat ID_L001_R1.fastq.gz ID_L002_R1.fastq.gz > ID_R1.fastq.gz
cat ID_L001_R2.fastq.gz ID_L002_R2.fastq.gz > ID_R2.fastq.gz
```

All files should be inside a directory, for example: *your_dir*
Inside *your_dir* there should be three directories: *funannotate_prep*, *funannotate and funannotate/ipsout*.
You can create them with this command:

```
mkdir -p your_dir/funannotate/ipsout && mkdir
your_dir/funannotate_prep
```

3 Download FamDB HDF5 database, Interproscan database and GeneMark license

3.1 FamDB HDF5 database

FamDB HDF5 database is needed for the **RepeatMasker** step. This database is partitioned by taxonomic groups, the partition needed for Fungi is partition number 0, for more information about partitions read this file: README.txt 2.1KB
FamDB HDF5 database can be downloaded **from here**.
Bash commands to download, unzip and mv the database to */your_dir*:

```
wget https://www.dfam.org/releases/Dfam_3.8/families/FamDB/dfam38-
1_full.0.h5.gz
gunzip dfam38-1_full.0.h5.gz
mv dfam38-1_full.0.h5 /you_dir
ln -s dfam38-1_full.0.h5 dfam38_full.0.h5
```

3.2 Interproscan database

This DB is needed for the **Interproscan** step.
Download the Interproscan DB from **here** (this file is >5Gb).

Commands to download and untar:

```
cd your_dir
wget https://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.69-
101.0/interproscan-5.69-101.0-64-bit.tar.gz
tar -pxvzf interproscan-5.69-101.0-*-bit.tar.gz
```

3.3 If you don't have a **GeneMark** license, get it **from this page**. License key file should be named *gm_key* and located in */your_dir*. This license is need to run the **Funannotate Predict** step.

Run hifiasm

4 Run the following command (replace /your_dir for the base directory where you have your data

```
docker run -it -v /your_dir:/ftmp dnalinux/hifiasm hifiasm -o
/ftmp/output/ID-longread -t 36 /ftmp/hifi.ID.fastq.gz
```

hifiasm takes a collection of individual DNA sequencing reads (the .fastq.gz file) and reconstructs the DNA sequence of the organism (the assembled contigs/scaffolds in the output directory)

Run gfatools gfa2fa

5 Run the following command (replace /your_dir for the base directory where you have your data

```
docker run -it -v /your_dir:/ftmp dnalinux/gfatools sh -c
"gfatools-final-gt/gfatools gfa2fa /ftmp/output/ID-
longread.bp.p_ctg.gfa > /ftmp/ID-longread.bp.p_ctg.fasta"
```

gfa2fa takes the detailed graphical representation of the assembled genome (the .gfa file) and simplifies it into a more standard, sequence-only format (the .fasta file)

Run sspace_longread

6 Run the following command (replace /your_dir for the base directory where you have your data

```
docker run -it -v /your_dir:/ftmp dnalinux/sspace_longread perl
SSPACE-LongRead.pl -c /ftmp/ID-longread.bp.p_ctg.fasta -p
/ftmp/hifi.ID.fastq.gz -b /ftmp/outperlID
```

SSPACE-LongRead takes fragmented genome assembly (contigs) and uses the long, accurate HiFi reads to piece these fragments together into larger, more contiguous representations of the chromosomes (scaffolds), reducing the number of gaps and improving the overall completeness of the genome assembly.

Run Gapcloser

7 Run the following command (replace /your_dir for the base directory where you have your data

```
docker run -it -v /your_dir:/ftmp dnalinux/lr_gapcloser bash
/LR_Gapcloser/src/LR_Gapcloser.sh -i
/ftmp/outperlID/scaffolds.fasta -l /ftmp/hifi.ID.fastq.gz -o
/ftmp/ID_lr-gapcloser
```

LR_Gapcloser takes a scaffolded genome with remaining gaps (represented by 'N's) and uses the long, accurate HiFi reads to fill in these gaps, resulting in a more complete and contiguous genome assembly.

Run BWA Index

8 Run the following command (replace /your_dir for the base directory where you have your data

```
docker run -it -v /your_dir:/ftmp dnalinux/bwa:0.7.17-3-deb bwa
index /ftmp/ID_lr-gapcloser/iteration-1/gapclosed.fasta
```

bwa index takes a linear DNA sequence (the FASTA file) and transforms it into a highly efficient and searchable data structure (the index files). This allows subsequent bwa commands (e.g., bwa mem for alignment) to quickly map millions or billions of short sequencing reads back to this genome assembly

Run fastp

9 Run the following command (replace /your_dir for the base directory where you have your data)

```
docker run -it -v /your_dir:/ftmp dnalinux/fastp:0.23.4 fastp --
in1  /ftmp/ID_R1.fastq.gz --in2  /ftmp/ID_R2.fastq.gz --out1
/ftmp/ID_R1_trim.fastq.gz --out2  /ftmp/ID_R2_trim.fastq.gz
```

fastp is a very fast all-in-one preprocessor for FASTQ files. It's designed to perform quality control, filtering, and trimming of raw sequencing reads. Raw reads from sequencing machines often contain errors, adapter sequences, and low-quality bases at their ends. It takes raw, potentially problematic paired-end sequencing reads and transforms them into high-quality, "cleaned" paired-end reads by removing errors, adapters, and low-quality sequences.

Run BWA mem

10

Run the following command (replace /your_dir for the base directory where you have your data). Replace CPU for your CPU count.

```
docker run -it -v /your_dir:/ftmp dnalinux/bwa:0.7.17-3-deb
/bin/bash -c "bwa mem -t CPU /ftmp/ID_lr-gapcloser/iteration-
1/gapclosed.fasta /ftmp/ID_R1_trim.fastq.gz
/ftmp/ID_R2_trim.fastq.gz > /ftmp/ID_aligned_reads.sam"
```

bwa mem takes cleaned sequencing reads and a pre-indexed reference genome (the gapclosed.fasta file) and transforms them into a .sam file.

Run SAMTOOLS

11 Sort and Index

Run the following command (replace /your_dir for the base directory where you have your data).

```
docker run -it -v /your_dir:/ftmp dnalinux/samtools:1.20-3-deb
/bin/bash -c "samtools view -Sb /ftmp/ID_aligned_reads.sam >
/ftmp/ID_aligned_reads.bam"

docker run -it -v /your_dir:/ftmp dnalinux/samtools:1.20-3-deb
samtools sort /ftmp/ID_aligned_reads.bam -o
/ftmp/ID_sorted_aligned_reads.bam

docker run -it -v /your_dir:/ftmp dnalinux/samtools:1.20-3-deb
samtools index /ftmp/ID_sorted_aligned_reads.bam
```

samtools view command takes the human-readable (but large) text-based alignment file and converts it into a smaller, more efficient binary format suitable for further computational analysis.
samtools sort reorganizes the alignment data in the BAM file into a canonical, ordered sequence that is required for subsequent steps.

samtools index creates a lookup table that enables fast random access to alignment data within the sorted BAM file, significantly speeding up downstream analyses.

Pilon

12 Run the following command (replace /your_dir for the base directory where you have your data).

```
docker run -it -v /your_dir:/ftmp dnalinux/pilon:1.24-3-deb pilon
--genome /ftmp/ID_lr-gapcloser/iteration-1/gapclosed.fasta --frags
/ftmp/ID_sorted_aligned_reads.bam --output /ftmp/ID_polished
```

pilon takes a nearly complete genome assembly and refines it using high-accuracy, short-read sequencing data. It effectively "cleans up" the assembly, reducing the number of single-base errors and small indels, resulting in a more accurate and higher-quality final genome sequence. This is often one of the final steps in producing a finished de novo genome assembly.

Funannotate

13 Funannotate Clean and Sort

Run the following command (replace /your_dir for the base directory where you have your data).

```
docker run -it -v /your_dir:/ftmp dnalinux/funannotate:latest
funannotate clean -i /ftmp/ID_polished.fasta -o
/ftmp/funannotate_prep/ID_polished_clean.fasta

docker run -it -v /your_dir:/ftmp dnalinux/funannotate:latest
funannotate sort -i /ftmp/funannotate_prep/ID_polished_clean.fasta
-o /ftmp/funannotate_prep/ID_polished_clean_sort.fasta --minlen
1000
```

funannotate clean takes a high-quality genome assembly and performs a set of standardized quality control and formatting steps to make it optimal for gene prediction and annotation.
funannotate sort ensures that the genome assembly is organized and that only sufficiently long sequences, which are most amenable to accurate gene prediction, are

carried forward for the intensive annotation process.

RepeatMasker

14 Run the following command (replace /your_dir for the base directory where you have your data). Remember that is step requires the dfam38_full.0.h5 database installed in a directory that should be called /ftmp in the docker. If dfam38-1_full.0.h5 is available, a ln to dfam38_full.0.h5 must be done (check step 3.1 for details)

```
docker run -it -v /your_dir:/ftmp dnalinux/repeatmasker:latest
/usr/local/RepeatMasker/RepeatMasker:4.1.6-configured -s -species
Fungi /ftmp/funannotate_prep/ID_polished_clean_sort.fasta -xsmall
```

RepeatMasker takes a raw genome assembly and transforms it by marking all known repetitive DNA elements, typically by converting their nucleotides to lowercase. This "masked" genome is then much more suitable for accurate gene prediction and other sequence analyses that are sensitive to repetitive regions.

Fuannotate Predict

15 Run the following command (replace /your_dir for the base directory where you have your data). Replace CPU for your CPU count.

```
docker run -it -v /your_dir:/ftmp dnalinux/funannotate-gmes-
dikarya:latest funannotate predict -i
/ftmp/funannotate_prep/ID_polished_clean_sort.fasta.masked -s ID -
o /ftmp/funannotate --cpus CPU
```

funannotate predict takes a masked genome assembly and, by integrating various computational tools and biological evidence, transforms it into a set of predicted gene models (including their genomic locations, exon-intron structures, and corresponding protein/CDS sequences) represented in standard bioinformatics formats like GFF3.

Interproscan

16 Run the following command (replace /your_dir for the base directory where you have your data). Replace CPU for your CPU count.

```
docker run -it -v /your_dir:/ftmp -v /your_dir/interproscan-5.69-
101.0/data:/opt/interproscan/data -v /tmp:/temp
dnalinux/interproscan:5.69-101.0 --input
/ftmp/funannotate/predict_results/ID.proteins.fa --disable-precalc
--output-dir /ftmp/funannotate/ipsout --cpu CPU
```

InterProScan takes a set of predicted protein sequences and transforms them into a rich functional annotation by identifying known protein signatures, assigning higher-level InterPro classifications, and linking them to Gene Ontology terms.

Funannoate annotate

17 Run the following command (replace /your_dir for the base directory where you have your data)

```
docker run -it -v /your_dir:/ftmp dnalinux/funannotate-gmes-
dikarya funannotate annotate -i /ftmp/funannotate --fasta
/ftmp/funannotate/predict_results/ID.proteins.fa --species ID --
out /ftmp/FA_results --iprscan
/ftmp/funannotate/ipsout/ID.proteins.fa.xml
```

funannotate annotate takes the predicted gene structures and their corresponding protein sequences, enriches them with detailed functional information derived from databases like InterProScan, and compiles all this into a complete and standardized set of genome annotation files. This is the culmination of the entire annotation pipeline, providing the biological insights into the sequenced organism.