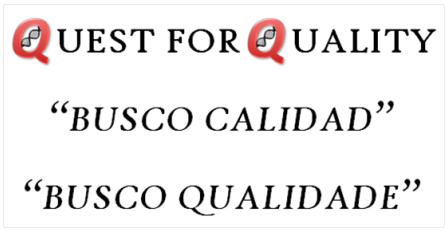

http://busco.ezlab.org
Version 1.1b; May 2015

BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs

Felipe A. Simão, Robert M. Waterhouse, Panagiotis Ioannidis,
Evgenia V. Kriventseva, & Evgeny M. Zdobnov

Zdobnov's Computational Evolutionary Genomics Group: http://cegg.unige.ch

Department of Genetic Medicine and Development, University of Geneva Medical School
and Swiss Institute of Bioinformatics, rue Michel-Servet 1, 1211 Geneva, Switzerland.

Introduction

BUSCO completeness assessment employs sets of Benchmarking Universal Single-Copy Orthologs from OrthoDB (www.orthodb.org) to provide quantitative measures of the completeness of genome assemblies, annotated gene sets, and transcriptomes in terms of expected gene content. Genes that make up the BUSCO sets for each major lineage are selected from orthologous groups with genes present as single-copy orthologs in at least 90% of the species. While allowing for rare gene duplications or losses, this establishes an evolutionary informed expectation that these genes should be found as single-copy orthologs in the genome of any newly-sequenced species.

Usage of the BUSCO software requires a working installation of Python 3, HMMER 3.1, Blast+, Augustus (genome assessment only) and EMBOSS transeq (transcriptome assessment only). BUSCO genome assembly assessment first identifies candidate regions from the genome to be assessed with tBLASTn searches using BUSCO consensus sequences. Gene structures are then predicted using Augustus with BUSCO block profiles. Finally, these predicted genes, or all genes from an annotated gene set or transcriptome, are assessed using HMMER and lineage-specific BUSCO profiles to classify matches as complete, duplicated, or fragmented, or when there are no matches, as missing.

BUSCO setup

The BUSCO distribution is released as a compressed archive file (BUSCO_v1.1b.tar.gz) for download. Extracting the files to your current directory **`tar -zxvf BUSCO_v1.1b.tar.gz`** will create the directory **`BUSCO`**, containing the required files.

Depending on the species you wish to assess, you should now download the appropriate lineage-specific profile libraries: Metazoa, Eukaryota, Arthropoda, Vertebrata, Fungi, or Bacteria from http://busco.ezlab.org to your **`BUSCO`** directory.

Before you begin, you will need to make sure that the following required software (some only required for genome or transcriptome assessments) are installed and accessible from the command-line, e.g. set environment variable PATH=$PATH:/path/to/software/bin

- Python 3

- NCBI BLAST+ http://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

- HMMER (HMMER 3.1b) http://hmmer.janelia.org/

- Augustus 3.0.x (genome only) http://bioinf.uni-greifswald.de/augustus/
Make sure that the environmental variable $AUGUSTUS_CONFIG_PATH was set during installation (e.g export AUGUSTUS_CONFIG_PATH=/path_to_augustus/config).
Write access to the Augustus installation directory is necessary for retraining the gene finder.

- EMBOSS tools 6.x.x (transcriptome only) ftp://emboss.open-bio.org/pub/EMBOSS/

BUSCO quick start

1- Genome assembly assessment:

```
python BUSCO_v1.1b.py -o NAME -in ASSEMBLY -l LINEAGE –m genome
```

NAME	name to use for the run and all temporary files
ASSEMBLY	genome assembly file in fasta format
LINEAGE	path to the BUSCO lineage data (i.e. /path/to/lineage)

2- Gene set assessment:

```
python BUSCO_v1.1b.py -o NAME -in GENE_SET -l LINEAGE -m OGS
```

NAME	name to use for the run and temporary files
GENE_SET	gene set protein sequence file in fasta format
LINEAGE	path to the BUSCO lineage data (i.e. /path/to/lineage)

3- Transcriptome assessment:

```
python BUSCO_v1.1b.py -o NAME -in TRANSCRIPTOME -l LINEAGE -m trans
```

NAME	name to use for the run and temporary files
TRANSCRIPTOME	transcript set sequence file in fasta format
LINEAGE	path to the BUSCO lineage data (i.e. /path/to/lineage)

BUSCO options

python BUSCO_v1.1b.py -in INPUT -o OUTPUT -l LINEAGE -m MODE [Options…]

1- Mandatory arguments

-o name	Name used for naming output files
-in input_file	Genome assembly / gene set / transcriptome in fasta format
-l lineage	Path to the BUSCO lineage data to be used Example: -l /path/to/metazoa
-m mode	Mode of analysis Valid options: genome, ogs, trans Default: genome

2- Optional arguments

-h –help	Print help
-c integer	Number of CPU threads to be used Default: 1
-sp species	Select from the pre-computed Augustus metaparameters Selecting a closely-related species usually produces better results Valid options: see Augustus help for list of options Default: generic
-e evalue	Use a custom blast e-value cutoff Default: 0.01
-f	Force overwriting of results files from a previous run with the same name
--flank N	Custom flanking genomic regions in base pairs (bp) Used when extending selected candidate regions before gene prediction Default: Automatically calculated flank sizes based on genome size
--long	Performs full optimization for Augustus gene finding training Default: Off

BUSCO Output

Successful execution of the BUSCO assessment pipeline will create a directory named **`name_OUTPUT`** where '**`name`**' is your assigned name for the assessment run. The directory will contain several files and directories:

1- Files

short_summary_	Contains summary results in BUSCO notation and a brief breakdown of the metrics
full_table_	Complete results in tabular format with coordinates, scores and lengths of BUSCO matches
training_set_	Set of complete BUSCO matches used for training Augustus Only created during genome assessment
_tblastn	Results in tabular format of tBLASTn searches with BUSCO consensus sequences

2- Directories

augustus_	Augustus-predicted genes Only created during genome assessment
augutus_proteins	Corresponding Augustus-predicted proteins Only created during genome assessment
Selected	Complete BUSCO matches, used for training Augustus
gb	Complete BUSCO matches, GenBank format
gffs	Complete BUSCO matches, GFF format
hmmer_output	Tabular format HMMER output of searches with BUSCO HMMs

BUSCO setup test with sample data

Sample data are provided to test your BUSCO setup. Execute the following commands and compare the final output '**run_SAMPLE**' with the provided files in '**run_TEST**'.

1. Change directory to 'sample_data'

```
cd sample_data/
```

2. Run BUSCO assessment on sequence file '**target.fa**' in genome mode.

```
python BUSCO_v1.1b.py -in target.fa -o SAMPLE -l example -m genome
```

3. Compare the final output '**run_SAMPLE**' with the provided files in '**run_TEST**'.

Example output: `short_summary_TEST`

```
#Summarized BUSCO assessment for file: target_sequence.fa
#BUSCO was run in mode: genome

Summary completeness assessment in BUSCO notation:
      C:80%[D:0.0%],F:0.0%,M:20%,n:10

Representing:
      8     Complete Single-Copy BUSCOs
      0     Complete Duplicated BUSCOs
      0     Fragmented BUSCOs
      2     Missing BUSCOs
      10    Total BUSCO groups searched
```

Example output: `full_table_TEST`

```
#BUSCO_group  Status    Scaffold  Start     End       Bitscore  Length
BUSCO_5       Complete  sample    66078     76647     475.7     287
BUSCO_7       Complete  sample    163394    174110    423.6     244
BUSCO_8       Complete  sample    228045    238915    238.1     189
BUSCO_1       Complete  sample    25227     35708     281.1     147
BUSCO_4       Complete  sample    64425     74970     420.3     419
BUSCO_6       Complete  sample    77357     91985     1259.1    688
BUSCO_2       Complete  sample    27021     51425     436.0     183
BUSCO_3       Complete  sample    62338     73243     237.5     144
BUSCO_9       Missing
BUSCO_10      Missing
```