Frederick National Laboratory for Cancer Research

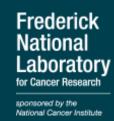
sponsored by the National Cancer Institute



Wrapper Scripts Overview for CANDLE Team

Andrew Weisman

Dec 16, 2020



Overview

- The wrapper scripts are a <u>GitHub repository</u> that helps to:
 - 1. **Setup:** Aids in setting up its **usage** alongside clones of the Supervisor and Benchmarks repositories on HPC systems
 - Documentation found <u>here</u>
 - 2. **Usage:** Adds various features to CANDLE while leaving the Supervisor/Benchmarks repos as untouched as possible
 - Full documentation now posted <u>here</u>



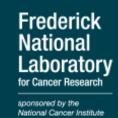
Overview of wrapper scripts functionality

- Run CANDLE as a central installation
- Edit a single input file for all settings
- Minimally modify a bare model script
- Run model scripts written in other languages
- Perform a consistent workflow for testing and production jobs
- Modify a single file if/when the CANDLE-compliance procedure changes



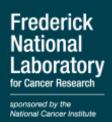
Loading the candle module

- Not yet on Summit: module load candle/tfl
- Currently: source /gpfs/alpine/med106/world-shared/candle/env_for_lmod-tf1.sh
- What this does:
 - Sets \$CANDLE to /gpfs/alpine/med106/world-shared/candle/tf1
 - Appends \$CANDLE/wrappers/bin to \$PATH
 - Sets \$SITE to summit-tf1
 - Appends \$CANDLE/Benchmarks/common to \$PYTHONPATH



Sample usage on Summit

- Enter an arbitrary directory on the Alpine filesystem
- source /gpfs/alpine/med106/world-shared/candle/env_for_lmod-tfl.sh
- candle import-template upf
- candle submit-job upf_example.in



Types of models

- Bare: Model downloaded from the Internet
- Canonically CANDLE-compliant: What we normally think of as CANDLE-compliant
- Minimally CANDLE-compliant: Minimal modification to bare models to get them to work with the wrapper scripts

Frederick National Laboratory for Cancer Research

How to modify a canonically CANDLE-compliant model for use with the wrapper scripts, by example

```
def initialize_parameters(default_model='nt3_default_model.txt'):
 import os # ADD THIS LINE
nt3Bmk = bmk.BenchmarkNT3(
     bmk.file path,
     # default model, # ORIGINAL LINE
     os.getenv('CANDLE_DEFAULT_MODEL_FILE'), # NEW LINE
     'keras',
     prog='nt3 baseline',
     desc='1D CNN to classify RNA sequence data in normal or tumor classes')
gParameters = candle.finalize_parameters(nt3Bmk)
return gParameters
```

How to modify a bare model for use with the wrapper scripts ("minimal CANDLE-compliance")



1. Set the hyperparameters in the model script using a dictionary called candle_params

2. Either:

- a. Define a metric of performance in the variable candle_value_to_return
- b. Define the history object as usual for Keras



Input file format

- &control section: Contains keywords specifying all settings except below
- &default_model section: Contains contents of, or points to, a default model file
- ¶m_space section: Contains contents of, or points to, a hyperparameter space file
- Example 1: all content in single input file
- <u>Example 2</u>: links to default model and parameter space files
- Example 3: mlrMBO parameter space section
- Example 4: running a benchmark

Recommended workflow: Run a benchmark using UPF or mlrMBO



- Enter a directory on the Alpine FS
- Load the CANDLE module
- Import one of the templates for running canonically CANDLE-compliant models: candle import-template {upf|mlrmbo}, keeping just the input file
- Rename and modify the input file
- Modify the initialize_parameters() function
- From an interactive node, set run_workflow=0 in the input file, and run candle submitjob <INPUT-FILE>
- From a login node, set run_workflow=1 (default setting), and run candle submit-job
 <INPUT-FILE>

Recommended workflow: Run UPF or mlrMBO on a brand new model



- Enter a directory on the Alpine FS
- Load the CANDLE module
- Create a bare model as usual
- Make the model minimally CANDLE-compliant
- Import one of the templates for running minimally CANDLE-compliant models: candle import-template {grid|bayesian}, keeping just the input file
- Rename and modify the input file
- From an interactive node, set run_workflow=0 in the input file, and run candle submitjob <INPUT-FILE>
- From a login node, set run_workflow=1 (default setting), and run candle submit-job
 <INPUT-FILE>



Miscellaneous

- See <u>documentation</u> for details on other recommended workflows, such as:
 - Running a model written in another language such as R or bash
 - Pulling your Supervisor/Benchmarks updates into the central CANDLE installation
 - Committing your changes to the wrappers/Supervisor/Benchmarks repos from the central CANDLE installation
- The <u>documentation</u> online also has some contribution ideas
- Contact me for help or discussion at <u>andrew.weisman@nih.gov</u> or on the ECP-CANDLE Slack at @Andrew Weisman

Questions?