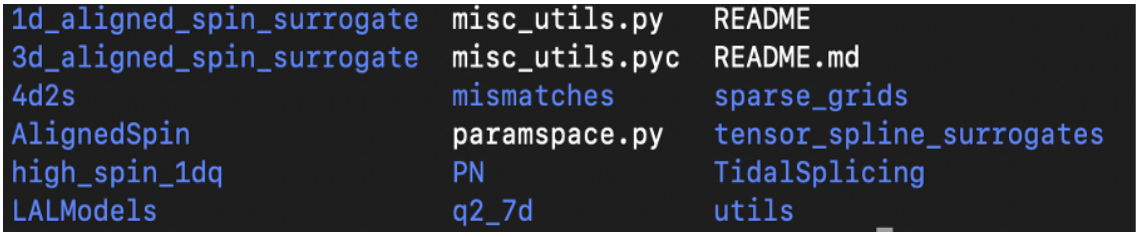
Quick Intro to Surrogate Modeling repo

Last Edited: Feb. 8th 2021

Short introduction to get started on building a surrogate model using the `surrogate_modeling` repo. There is README in every sub-directories with brief summary and instruction, so check them out.



This introduction gives a brief explanation of each step in the process and includes sample inputs for the python scripts needed to be run during the process. Highly recommend doing "python [name of the script].py -h" for each of the script. It will give summary of all possible inputs for that particular script.

Non-precessing Case

For the non-precessing surrogate waveform model (example Ref. Varma et al. [2019]), proceed to AlignedSpin directory within the `surrogate_modeling` repo. Within this directory, there many sub-directories. Only need to care about following four if building Hybrid NR surrogate.

(1) ErrorChecking - includes scripts for testing the surrogate once it's built
(2) Hybridization - inlcudes scripts for hybridization process
(3) NRSurroagate - includes script for generating the NR surrogate
(4) Scripts - includes scripts used in building NR surrogate

First, proceed to Hybridization directory:

Step 1: Run the 'setup.py script', which creates directories that will contain the waveform data that will be used in building the surrogate.

python setup.py -o [name of directory]

e.g. python setup.py -o ../data/q15Sur

```
[jy884@wheeler:[/panfs/.../data/q15Sur]$ pwd
/home/jy884/work/surrogate_modeling/AlignedSpin/data/q15Sur
[jy884@wheeler:[/panfs/.../data/q15Sur]$ ls
HiRes  MedRes  setup.py_opts.json
```

Step 2: Run '`get_annex_data.py`', which retrieve the NR data and prepare them for surrogate building. The preparation includes aligning with time shift and rotation and interpolating the waveforms.

python `get_annex_data.py` -d [same directory name used with setup.py] -a [path to SimAnnex] `--no_drop --project` [name of the BFI project] -n [number of the digits for the case numbers] `--tAlign` [alignment time relative to the merger; default -4500] `--save_h_as_dict` -ov 0 `--include_close_cases`

e.g. python `get_annex_data.py` -d ../data/q15Sur -a /home/jy884/work/SimAnnex `--no_drop --project q15Sur` -n 3 `--tAlign` -4500 `--save_h_as_dict` -ov 0 `--include_close_cases`

```
--no_drop : do not 'git annex drop' files afterwards
--save_h_as_dict : save strain as dictionary rather than a list
--include_close_cases : also include cases very similar to ones we already have
-ov : overwrite settings, 0 = skip, 1 = overwrite, -1 = raise exception
```

The waveforms get stored in [name of directory]/HiRes/NR/`raw_data`.

```
jy884@wheeler:[/panfs/.../NR/raw_data]$ pwd
/home/jy884/work/surrogate_modeling/AlignedSpin/data/q15Sur/HiRes/NR/raw_data
jy884@wheeler:[/panfs/.../NR/raw_data]$ ls
annex_params.txt  Case_0002.h5  Case_0004.h5  Case_0006.h5  Case_0008.h5  Case_0011.h5  Case_0013
Case_0001.h5      Case_0003.h5  Case_0005.h5  Case_0007.h5  Case_0010.h5  Case_0012.h5  Case_0014
```

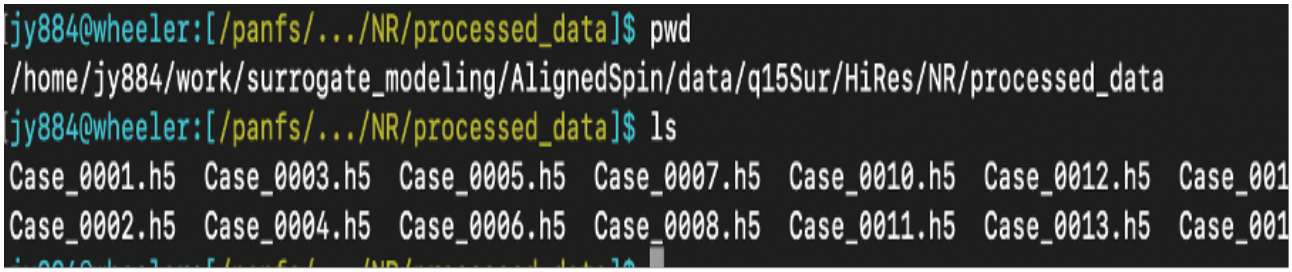
Step 3: Run '`process_NR_data.py`', which process the waveform (truncate the junk) and saves into processed_data directory. This directory will be used for the subsequent hybridization scripts. Note it also saves 'extra-processed' data into `data_for_sur` directory which is for building NR surrogate instead of Hyb waveform surrogate.

python `process_NR_data.py` -d [same directory named used with setup.py] -ov 1 `--customModes --tStart -4500.00 --tRing 120.0`

e.g. python `process_NR_data.py` -d ../data/q15Sur/ -ov 1 `--customModes --tStart` -4500.0 `--tRing` 120.0

```
--customModes : only keep a specific subset of mode define within the script
--tJunk : length of initial data to be truncated (default 500)
--tRing: length of post merger to keep wrt to peak (default 75)
--tStart : start time of common time array
```

After running above three scripts, we now have processed NR waveforms, ready for hybridization.



Step 4: Run the '`run_hybridization.py`'. Should run on compute node.

python `run_hybridization` -d [same directory name used with setup.py] -omega0 0.01 `--pts_per_orbits 5 --n_orbits 3 -np 24 -ov 0 --tAlign -1000 --tStart_NR 500 --domain_end` 120.0

e.g. `run_hybridization` -d ../data/q15sur -omega0 0.01 `--pts_per_orbits 5 --n_orbits 3` -np 24 -ov 0 `--tAlign -1000 --tStart_NR 500 --domain_end` 120.0

```
-omega0 : starting orbital frequency for PN data
--n_orbit : number of orbits in the matching region for hybridization
--pts_per_orbits : number of points to keep per orbit
-np : number of processors to use
--tAlign : phase alignment time (default -1000) with respect to peak time (set to 0)
--domain_end: end time for common time domain
```

This goes through cases within the processed_data directory, makes case list. It generates the PN waveform for each corresponding cases, hybridizes the generated PN waveform with the processed NR data. Process the hybrid waveform. Finally, end up with hybridized waveform data ready for surrogate in: [name of directory]/HiRes/Hyb/`data_for_sur`

```
[jy884@wheeler:[/panfs/.../HiRes/Hyb]$ pwd
/home/jy884/work/surrogate_modeling/AlignedSpin/data/q15Sur/HiRes/Hyb
[jy884@wheeler:[/panfs/.../HiRes/Hyb]$ ls
data_for_sur  Hybridize.py_opts.json  process_hybrids.py_opts.json  raw_data  run_hybridization.py_opts.json
```

```
/home/jy884/work/surrogate_modeling/AlignedSpin/data/q15Sur/HiRes/Hyb/data_for_sur
[jy884@wheeler:[/panfs/.../Hyb/data_for_sur]$ ls
Case_0001.h5  Case_0010.h5  Case_0101.h5  Case_0109.h5  Case_0117.h5  Case_0125.h5          Paramspace.pdf
Case_0002.h5  Case_0011.h5  Case_0102.h5  Case_0110.h5  Case_0118.h5  Case_0126.h5          Paramspace.png
Case_0003.h5  Case_0012.h5  Case_0103.h5  Case_0111.h5  Case_0119.h5  Case_0127.h5          Readme_validity
Case_0004.h5  Case_0013.h5  Case_0104.h5  Case_0112.h5  Case_0120.h5  Case_0128.h5
Case_0005.h5  Case_0014.h5  Case_0105.h5  Case_0113.h5  Case_0121.h5  Case_0129.h5
Case_0006.h5  Case_0015.h5  Case_0106.h5  Case_0114.h5  Case_0122.h5  Case_0130.h5
Case_0007.h5  Case_0016.h5  Case_0107.h5  Case_0115.h5  Case_0123.h5  domain.npy
Case_0008.h5  Case_0100.h5  Case_0108.h5  Case_0116.h5  Case_0124.h5  domain_truncation.png
```

Now, we can start building Hybridized NR surrogate.
Go to NRSurrogate directory.

Step 5: Run 'RunValidation.py'

python RunValidation.py -r [directory name to store the surrogate model] -d [path to `data_for_sur` of Hyb data] `--BuildSurOnNodes --CornerCasesFile` CornerCases.txt -k [number of cases to leave out]

e.g. python RunValidation.py -r ../Results/`q15Sur_test` -d ../data/q15Sur/HiRes/Hyb/`data_for_sur`/ `--BuildSurOnNodes --CornerCasesFile` CornerCases.txt -k 1

```
--BuildSurOnNodes : this way it will just create directories for surrogate data
and the run scripts instead of running now
--CornerCasesFile : if we have some cases that we do not want to be used for
validation, then we include those case numbers in the CornerCases.txt
-k : LeavekOut basically means we are leaving "k" number of cases out when
training the surrogate so that we can use those "k" cases for validation later on.
```

Now, we have created new directory with some run scripts and LeavekOutSur directory within. For example, assuming we have total of 20 cases and we set k to 1, we will have Leave1OutSur and within this directory, Set 0 through Set 19 (with each set corresponding to a different surrogate model with a different single case left out during the training).

Each submission script is meant for building the surrogate for corresponding set and subsequently testing with the validation case (the single case left out, in the case of Leave1Out)

```
[jy884@wheeler:[/panfs/.../Results/q15Sur_test]$ pwd
/home/jy884/work/surrogate_modeling/AlignedSpin/Results/q15Sur_test
[jy884@wheeler:[/panfs/.../Results/q15Sur_test]$ ls
Leave0Out_set0_old.stderr  Leave1Out_set13.stdout  Leave1Out_set6.stdout         Submit_Leave1Out_set13.input
Leave0Out_set0_old.stdout  Leave1Out_set14.stderr  Leave1Out_set7.stderr         Submit_Leave1Out_set14.input
Leave0Out_set0.stderr      Leave1Out_set14.stdout  Leave1Out_set7.stdout         Submit_Leave1Out_set1.input
Leave0Out_set0.stdout      Leave1Out_set1.stderr   Leave1Out_set8.stderr         Submit_Leave1Out_set2.input
Leave0OutSur               Leave1Out_set1.stdout   Leave1Out_set8.stdout         Submit_Leave1Out_set3.input
Leave1Out_set0.stderr      Leave1Out_set2.stderr   Leave1Out_set9.stderr         Submit_Leave1Out_set4.input
Leave1Out_set0.stdout      Leave1Out_set2.stdout   Leave1Out_set9.stdout         Submit_Leave1Out_set5.input
Leave1Out_set10.stderr     Leave1Out_set3.stderr   Leave1OutSur                  Submit_Leave1Out_set6.input
Leave1Out_set10.stdout     Leave1Out_set3.stdout   suball.sh                     Submit_Leave1Out_set7.input
Leave1Out_set11.stderr     Leave1Out_set4.stderr   Submit_Leave0Out_set0.input   Submit_Leave1Out_set8.input
Leave1Out_set11.stdout     Leave1Out_set4.stdout   Submit_Leave1Out_set0.input   Submit_Leave1Out_set9.input
Leave1Out_set12.stderr     Leave1Out_set5.stderr   Submit_Leave1Out_set10.input
Leave1Out_set12.stdout     Leave1Out_set5.stdout   Submit_Leave1Out_set11.input
Leave1Out_set13.stderr     Leave1Out_set6.stderr   Submit_Leave1Out_set12.input
```

Each submission script runs 'NRSurrogate.py' and 'TestSurrogate.py'.

```
this_dir=$(pwd)
cd /panfs/ds08/sxs/jy884/surrogate_modeling/AlignedSpin/NRSurrogate
python NRSurrogate.py -r $this_dir/Leave1OutSur/Set0 -d ../data/q15Sur/HiRes/Hyb/data_for_sur/ --nprocs 1


cd /panfs/ds08/sxs/jy884/surrogate_modeling/AlignedSpin/ErrorChecking
mpirun -np 1 python TestSurrogate.py -r $this_dir/Leave1OutSur/Set0 -d ../data/q15Sur/HiRes/Hyb/data_for_sur/  --onlyValidati
on
```

'NRSurrogate.py' is a symbolic link to whichever surrogate script we are trying to build. 'TestSurrogate.py' test the surrogate against validation cases. (by altering the input one could also test it against the training cases as well).

References

Vijay Varma, Scott E. Field, Mark A. Scheel, Jonathan Blackman, Lawrence E. Kidder, and Harald P. Pfeiffer. Surrogate model of hybridized numerical relativity binary black hole waveforms. *Phys. Rev.*, D99(6):064045, 2019. doi: 10.1103/PhysRevD.99.064045.