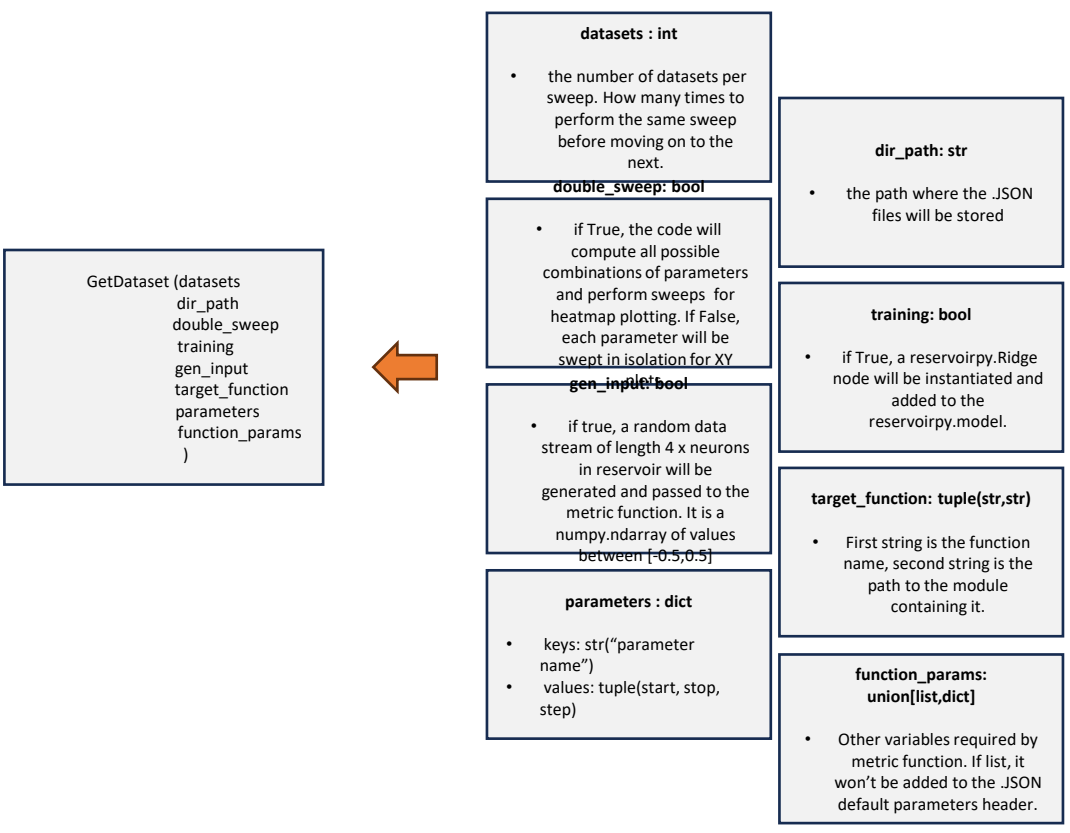
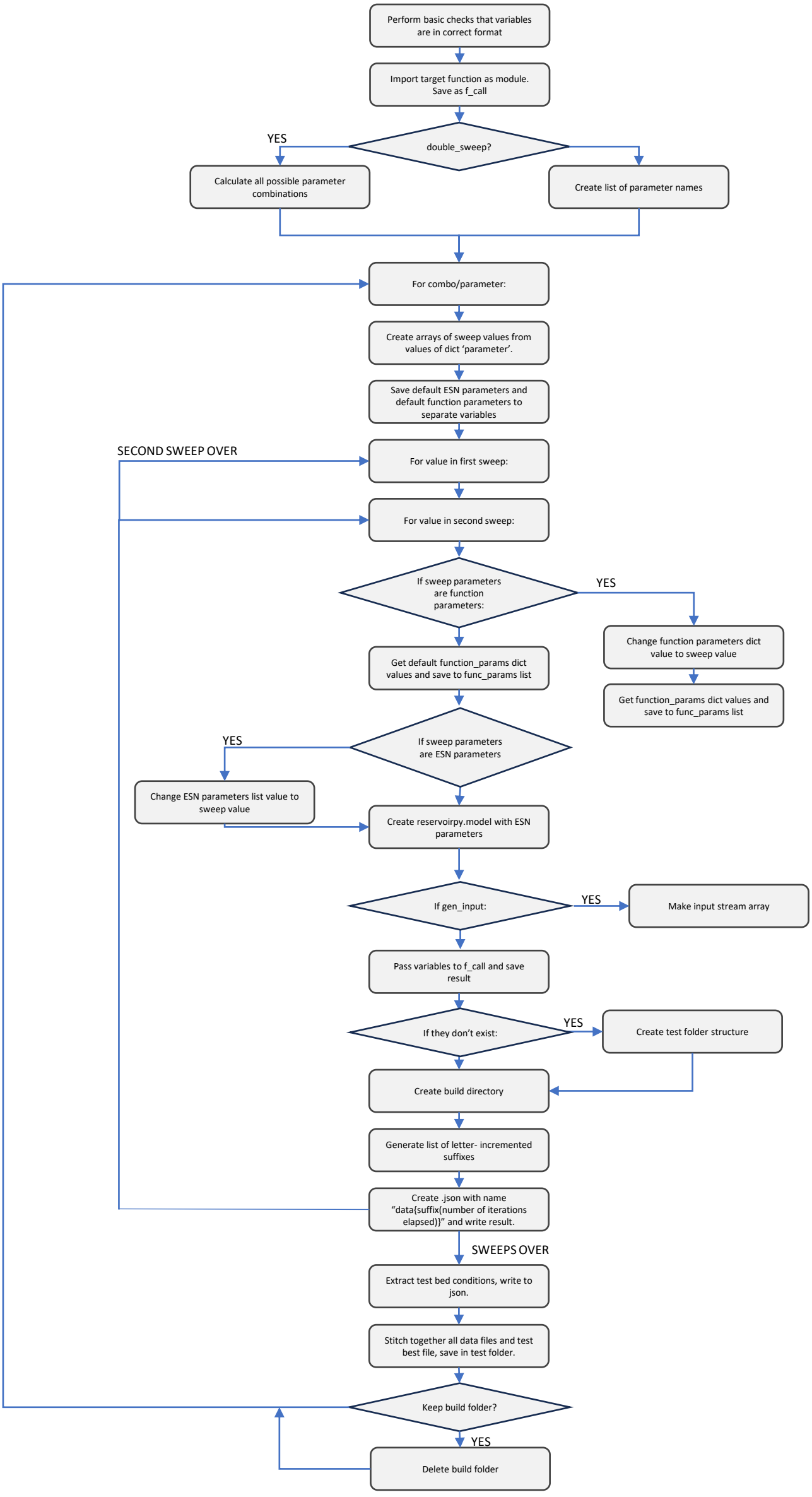

GetDataset (datasets
dir_path
double_sweep
training
gen_input
target_function
parameters
function_params
)
datasets : int
• the number of datasets per sweep. How many times to perform the same sweep before moving on to the next.
double_sweep: bool
• if True, the code will compute all possible combinations of parameters and perform sweeps for heatmap plotting. If False, each parameter will be swept in isolation for XY plots
gen_input: bool
• if true, a random data stream of length 4 x neurons in reservoir will be generated and passed to the metric function. It is a numpy.ndarray of values between [-0.5,0.5]
parameters : dict
• keys: str(“parameter name”)
• values: tuple(start, stop, step)
dir_path: str
• the path where the .JSON files will be stored
training: bool
• if True, a reservoirpy.Ridge node will be instantiated and added to the reservoirpy.model.
target_function: tuple(str,str)
• First string is the function name, second string is the path to the module containing it.
function_params: union[list,dict]
• Other variables required by metric function. If list, it won’t be added to the .JSON default parameters header.


Perform basic checks that variables are in correct format
Import target function as module. Save as f_call
double_sweep?
YES
Calculate all possible parameter combinations
Create list of parameter names
For combo/parameter:
Create arrays of sweep values from values of dict 'parameter'.
Save default ESN parameters and default function parameters to separate variables
SECOND SWEEP OVER
For value in first sweep:
For value in second sweep:
If sweep parameters are function parameters:
YES
Change function parameters dict value to sweep value
Get function_params dict values and save to func_params list
Get default function_params dict values and save to func_params list
If sweep parameters are ESN parameters
YES
Change ESN parameters list value to sweep value
Create reservoirpy.model with ESN parameters
If gen_input:
YES
Make input stream array
Pass variables to f_call and save result
If they don't exist:
YES
Create test folder structure
Create build directory
Generate list of letter- incremented suffixes
Create .json with name "data{suffix(number of iterations elapsed)}" and write result.
SWEEPS OVER
Extract test bed conditions, write to json.
Stitch together all data files and test best file, save in test folder.
Keep build folder?
YES
Delete build folder