Domain

"Function_name" (location)

A brief description of the function
Note: Any strange notes?
TODO: Any todo's for this operation?
Details:

- "keyword" (type): (Default: default_value) Description of the detail
  - Further details
- "keyword" (type): Description and must be one of:
  - "valid_value_1": description
  - "valid_value_2": description
  - ...

Generic Functions (all of them)

All steps in a queue document contain these keys. The agent, (container), (details), operation, and (time_est) parameters are the only ones which may need to be defined at the time of the queue's creation.
"**agent**" (str): Abbreviated name of (sub)system
"completed" (str): (Default 'no') 'yes'/'no' if completed
"container" (str): A nickname for the plate, is reference to a key in 'containers'
"details" (dict): (Default: {}) Additional details

- "is_paired" (str): (Default: 'no') 'yes'/'no' if the operation is paired
  - A paired operation constrains the next operation to occur at a specific time after this operation (This time is specified via "schedule_time" which will default to 0 if not present)

"end_time" (str): (Default: None) "mm/dd/yyyy hh:mm:ss"
"**operation**" (str): A function name (Heading Level 2)
"start_time" (str): (Default: None) "mm/dd/yyyy hh:mm:ss"
"time_est" (int): (Default: ~5) Estimated duration of the task in seconds

- A missing time_est is interpreted as different values depending on which process is looking, but it's typically brief (<60) value

All Systems

"complete_queue" (functionals.complete_queue)

Final step in queue, marks the queue completed and moves it to the historical collection
Details: (none)

- "model_details" (dict): (Optional)
  - E.g. {"model_A": {"status": complete"} }
- "start_from" (int): (Default: 0) Allows the completion script to restart from a partially incomplete attempt

"void" (functionals.db_nop)

A "do nothing" command. Used to put temporary placeholder steps in a queue and can be used for testing.

Details:

- “wait” (number): (Default: 0) Sleeps ‘wait’ seconds before completing

LC

“run_analytical_batch” (run_analytical_batch)

Description

- Runs an analytical batch on the given plate. If the contents of a well are marked as ‘final_product’ : ‘yes’, then the analytical batch adds a semiprep batch to take place immediately after the analytical batch

Details: (none)

“run_semiprep_batch” (run_semiprep_batch)

Description

- Works the same as run_analytical_batch except with a specified target container to collect fractions into.

Details:

- “target_container” (str): where to collect fractions into
- “use_new_plate” (str): optional, acceptable values: ‘yes’, ‘no’. Whether or not to collect into an existing plate (if already on fraction collector) or to fetch a new plate. Defaults to no if plate already on fraction collector and yes if a plate must be supplied

AH

“prepare_wellplate” (evoware_method_selector)

Standard method to prepare a well-plate in one of three use-cases: (1) inert box reaction preparation, (2) low-temperature preparation, (3) standard preparation

Details:

0. Bog-standard detail keys

- empty_wells (Boolean) True/False
  - Adds extra washing steps when adding anything into a non-empty well-plate
  - (May be obsolete if we change to free-dispensing everything)
- tip_prep_rinse (Boolean) True/False
  - Washes the tips with 10mL/10mL instead of 3mL/3mL
  - Useful for getting bubbles out of the lines when liha has not be used in a while
- washing_frequency (String)
  - Currently under construction
  - Potentially allows for washing to take place at different frequencies

1. Inert box reaction preparation

- Allows the inert box to be assembled and prepared, this leaves the reaction plate inside of the inert reaction box after the Evoware method finishes
- If the temperature_setpoint is specified, then the heater-shaker will finish with the specified temperature
- If the temperature_setpoint is not specified, then the heater will not start, requiring a start heater-shaker operation in the queue to execute the reaction
  - Keys
    - inert_atmosphere (Boolean) True/False
      - Must be true for inert reaction preparation

- temperature_setpoint (Number) [20, 120]
    - If not present, defaults to no temperature and does not heat
- preparation_shaking (Boolean) True/False
    - If not present, defaults to no shaking
- shaking_rpms (Number) [100, 2000]
    - If not present and preparation_shaking is True, defaults to 500 RPM
    - If present and preparation_shaking is True, rounds to the nearest integer
- waiting_prep_time (Number) - Seconds
    - Exists to allow the well-plate to heat-up before adding solvents
    - Time provided is independent of inert_purge_time
- inert_purge_time (Number) - Seconds
    - Exists to allow the well-plate to purge before adding solvents
    - Time provided is independent of waiting_prep_time
- solvent_waiting_time (Number) - Seconds
    - Exists to wait after adding solvents into the reaction well-plate
    - Time provided is independent of initial plate prep waiting times

2. Low-temperature preparation
    - Allows a well-plate to be prepared at low-temperature, this leaves the reaction plate on the low-temperature heater-shaker after Evoware finishes the method
        - Keys
            - low_temperature (Boolean) True/False
                - Must be true for low-temperature preparation and inert_atmosphere must be False or not present
            - temperature_setpoint (Number) [4, 80]
                - If not present, defaults to no temperature
            - preparation_shaking (Boolean) True/False
                - If not present, defaults to no shaking
            - shaking_rpms (Number) [100, 2000]
                - If not present and preparation_shaking is True, defaults to 500 RPM
                - If present and preparation_shaking is True, rounds to the nearest integer
            - waiting_prep_time (Number) - Seconds
                - Exists to allow the well-plate to cool before adding solvents
            - solvent_waiting_time: Number (seconds)
                - Exists to wait after adding solvents into the reaction well-plate
                - Time provided is separate from initial preparation time steps
                - If the solvent_waiting_time cannot be converted into a Number, skips the solvent waiting time
3. Standard reaction well-plate preparation
    - Allows a well-plate to be prepared on a normal well-plate position on the liquid handler
        - Keys
            - solvent_waiting_time: Number (seconds)
                - Exists to wait after adding solvents into the reaction well-plate
                - If the solvent_waiting_time cannot be converted into a Number, skips the solvent waiting time

“prepare_characterization_plate” (evoware_method_selector)

- This function attempts to prepare a characterization plate from either reagent stock solutions or from previously isolated products. There is more functionality for wells with one target molecule/molecule of interest due to complications of simultaneously preparing multiple components from arbitrary stock solution concentrations. For multiple molecules of interest, currently if they are within 1-300xDilution from the target concentration (the closer to 300 the better) multiple components are also preparable

Details:

- tip_prep_rinse (Boolean) True/False
- empty_wells (Boolean) True/False

Extra Details:

- Format of the characterization well plate container for the queue document (example)
- Allowed specifications for well to prepare: OD (requires extinction coefficient to be calculated, not always there for reagent stock solutions currently), concentration, amount
- There is also a requirement to specify the container category to look for, a product made on the platform has several potential plates (reaction_plate, extraction_plate, filtrate_plate, solid_filter_plate, fraction_plate) while a reagent stock solution only needs to look for a ‘reagent_tray’. Without knowing a plate category it is very easy to find the wrong well-plate

```
characterization_plate:
        container_name: null
        plate_type: 96 Well Microplate
        contents:
           A1:
                target_molecules:
                - - '[O-]S(=O)(=O)C1=CC=CC=C1C1=NC2=C(S1)N=C(S2)\C=C\C1=CC=CC=C1'
                  - 1.5
                  - OD
                  - reaction_plate
                  - 1
                solvents:
                - - dmf
                  - 1
                total_volume: 300
```

“cleanup_characterization_plate” (evoware_method_selector)

- This function allows characterization wells to be changed to meet the needs of characterization assays, currently this includes dilution and concentration although this is also a good place to have quenching or addition of a species included into the workflow

Details:

- target_cleanup (String)
    - String name for set of cleanup operations

Extra Details:

- Format of the characterization well plate container, this is an update to the container that was used to create the well plate

```
characterization_plate:
        container_name: null
        plate_type: 96 Well Microplate
        contents:
           A1:
                target_molecules:
                - - '[O-]S(=O)(=O)C1=CC=CC=C1C1=NC2=C(S1)N=C(S2)\C=C\C1=CC=CC=C1'
                  - 1.5
                  - OD
                  - reaction_plate
                  - 1
                solvents:
                - - dmf
                  - 1
```

```
total_volume: 300
additional_cleanup:
        cleanup_1:
                concentration_factor: 1.5
```

"transfer_wellplate" (evoware_method_selector)

Responsible for transferring well-plates around the liquid-handler bed and for disassembling the inert reaction box

Details:

- disassemitmble_inert (Boolean) True/False
    - If true returns the inert reaction box back to the hotel
    - A standard implementation is to disassemble the inert box and then transfer the well-plate somewhere else to free the heater-shaker
- target_destination (String)
    - Allowed_values: heater_shaker, transfer_hotel, storage_hotel, bed_position, spark, tip_prep, tip_cleanup
- temperature (Number) [4, 120]
    - Used to select an appropriate heater shaker to transfer the well-plate into, returns an error if there are no acceptable heater-shakers present on the bed

"start_stop_heater_shaker" (evoware_method_selector)

Used to turn on and turn off the heater shaker

Details:

- power (String) on/off
- rpms (Number) [100, 2000]
    - Required if the power is 'on'
- temperature (Number) [4, 120]
    - Required if the power is 'on'
    - If turning off the heater shaker the set point is reset to 25C
- is_paired (String) yes/no
    - Connects the start command to the stop command
    - Function depends on MCN implementation

"filter_wellplate" (evoware_method_selector)

Responsible for constructing, filtering, and disassembling the TeVaC system

Details:

- target_container (String)
    - Matches a valid empty container in the queue document
    - The filter_wellplate function creates a well-plate
- hplc_container (String)
    - Matches a valid container in the queue document
- initial_solvent_volume (Number)
    - The number of uL of solvent to dispense into the initial well plate
    - Requires that a liha_solvent_to_dispense key is present
- initial_filter_solvent (String)
    - Must match a solvent that is present in the solvents collection
- hplc_container_solvent (String)
    - Must match a solvent that is present in the solvents collection
    - Defaults to "dmso" when not provided

- hplc_prep_volumes (Object)
  - A dictionary that defines how the hplc plate should be prepared
  - Default dictionary is hard-coded as:

```
{'final_product_transfer': 100,
 'extra_final_product_volume': 0,
 'intermediate_product_transfer': 50,
  'extra_intermediate_product_volume': 75,
  'analytical_product_transfer': 20,
  'extra_analytical_product_volume': 50}
```

- mca_filter_volume (Number)
  - Volume of solvent for the MCA to send through the well-plate filter
- mca_mix (Boolean) True/False
  - Decides if the mca should mix the liha_solvent that was added
  - This should basically always be True unless there is a good reason
- tevac_location (String) front/back
  - Currently the only location that is suitable is the **front** location
  - This is a work-in-progress to allow tall well-plate filtering in the front (when it becomes important to do so)
- tevac_filter_time (Number) Seconds
  - Adds a waiting time to filter solvents through the well-plate filter
- internal_standards (Boolean) True/False — ***CURRENTLY NOT ENABLED***
  - If true this just avoids adding things into wells that already contain the internal standard (these wells should already have enough liquid present)
  - This is likely deprecated pending reaction optimization workflow

“liquid_liquid_extraction” (evoware_method_selector)

Performs a liquid-liquid extraction of wells that appear in the target container contents in the queue document container

Details:

- target_container (String)
  - Matches a valid empty container in the queue document
  - The liquid_liquid_extraction function creates a well-plate
- extraction_iterations (Number)
  - Number of extraction iterations to do (must be greater than 0)
  - The first iteration will add iteration1_volume
  - Subsequent iterations will add back the filter_volume_to_dispense
  - A person needs to verify that the volume is correct for the well-plate type currently
- important_layer (String) top/bottom
  - Determines which layer the MCA will extract from the well-plates
  - Also determines the soft mixing cycle (top>bottom vs. bottom>top)
- other_phase_solvent (String)
  - Matches a valid solvent in the solvents collection
  - Very typically this would be ‘Water’ for example
- other_phase_volume_to_add (Number)
  - The amount of the other phase solvent to add into the well-plate for extraction
  - A person needs to verify that the container can hold the volume
- liha_solvent_to_dispense (String)
  - Matches a valid solvent in the solvents collection
  - A pretty typical example would be ethyl_acetate
- iteration1_volume (Number)

  - Volume of liha_solvent_to_dispense to start with for the extraction phase
- filter_volume_per_iteration (Number)
  - Volume of solvent for the MCA to extract from the well-plate with each iteration defined by the extraction_iterations
  - This is volume of liha_solvent_to_dispense that is added after the first iteration
- settling_time (Number) Seconds to wait
  - Time to wait after the MCA soft-mixing cycle before extracting from well-plate
- number_of_soft_mixes (Number)
  - Number of times that the mca will put the important_layer into the other_layer
- internal_standards (Boolean) True/False
  - If true this just avoids adding things into wells that already contain the internal standard (the volume should be large enough of pure internal standard from prep
  - This is likely deprecated pending reaction optimization workflow

“copy_wellplate” (evoware_method_selector)

Makes a copy of a wellplate or takes an aliquot from a wellplate
Details:

- target_container (String)
  - Matches a valid empty container in the queue document
- vol_to_add_initial (Number)
  - Volume of solvent to add to the initial wellplate before making a copy
- makeup_solvent (String)
  - Matches a valid solvent in the solvents collection
  - Solvent added to the initial wellplate
- dilution_factor (Number)
  - Dilution of the final wellplate relative to the initial wellplate
  - Number must be less than 1
- target_volume (Number)
  - Volume of the final well plate
  - Used when calculating the volume to aspirate from the initial wellplate
- aliquot_volume (Number)
  - Volume of initial wellplate to aspirate as an aliquot
- solvent_to_use (String)
  - Matches a valid solvent in the solvents collection
  - This solvent is used with aliquot_volume

MC

“move_wellplate” (move_wellplate)

Moves a well-plate from its location to a target destination via the robotic arm (SCARA)
Details:

- “target_destination” (str): A string naming the destination, one of
  - “liquid_handler”: Liquid handler (for all plates, big and small)
  - “fraction_collector”: HPLC fraction collector bed
  - “thermal_reactor”: High-T reactor
  - “autosampler”: HPLC autosampler bed
  - “ir_spectrometer”: FTIR HTX location
  - “lpx”: Storage carousel

    - “nmr_transfer”: for loading and unloading nmr tubes into the NMR_tube_carrier
- “scan_barcode” (str): (Default: None) ‘yes’/’no’ is the Ra should run the plate by the barcode reader

“load_pfa_film” (load_pfa_film)

Loads a pfa film into the thermo-reactor
Details: (none)

“unload_pfa_film” (unload_pfa_film)

Unloads a pfa film from the thermo-reactor
Details: (none)

“soft_wait” (soft_wait)

Enforces that at least some time passes between two operations.
Note: “time_est” (int) is used without “is_paired” for soft_wait to enforce that at least ““time_est” seconds are waited before continuing to the next step. A soft_wait that has “is_paired” should be replaced by a specification of “schedule_time” in an “is_paired” productive step.
Details: (none)

“await_queue_step” (await_queue_step)

Enforces that the calling queue cannot continue until the specified step in a referenced queue is complete.
Note: “time_est” (int) should be a very large number so that the scheduler does not issue a timeout of the operation. (the tardy response should be handled, but this lessens the number of faults reported). This method should not await operations which comprise recovery protocols or operations which occur multiple times in a queue as this may throw off its ability to resolve changed step numbers.
Details:

- ‘await_queue’ (str): The name of the queue being watched
- ‘step_num’ (int): The step number being awaited
- ‘operation’ (str): (Default: None) The name of the operation—used to protect against step numbers in the awaited queue changing from recovery steps. When not None, it will adjust step_num to be the step # of the first occurrence of ‘operation’ which occurs on or after the original ‘step_num’.
- ‘interval’ (int): (Default: 5 [units: minutes]) The frequency at which the DB should be checked [Units: Minutes]
- ‘timeout’ (int): (Default: 5*24*60 [units: minutes] or 5 days) How long the awaited queue should be monitored before timing out [Units: Minutes]. This should be less than the time_est parameter.

SP

“Ss.request” (lpx_request)

Moves a plate from the LPX carousel to its transfer station

Note 1: ‘container_name’ field is used to specify a plate reference (queue nickname, the dictionary key in the queue document; not an entry in the wellplate/consumables collection)
Note 2: While it can be used directly, the best practice is to use aceso.resource_request() as this method will handle populating ‘container_name’ (and tell you what value it uses, as this may be procedurally generated) and adding additional steps into the queue such that the resource it transferred to your desired location.

aceso.resource_request(

- queue_name (str): Name of the queue
- step (str/int): Step requesting the resource
    - steps will be inserted prior to this step
- target_destination (str): Where the resource is needed
    - Uses same codex as with robotic arm
    - Has some safety (eg. ‘lh’ and ‘fc’ will remap to the correct value)
- is_new (bool):
    - True - The plate is not reference by the queue, this entry must be added
    - False - The plate is referenced by the queue (keys ignored: plate_type & generate)
- name_basis (str):
    - Is_new = True - Use this nickname for adding the plate reference to the queue (see: generate)
    - Is_new = False - Look up and use this specific key/nickname
- plate_type (str): The plate/resource type (default: None, so that it can be omitted when ‘is_new’ is False, but it must be a valid string if ‘is_new’ is True)
    - "96 Well Microplate"
    - "96 Well DeepWell"
    - “DiTi SBS Waste”
    - “96 Well Filtration Plate” (not tested)
- generate (bool):
    - True - Interpret ‘name_basis’ as a seed for generating a name
        - The queue’s containers are searched for ‘name_basis’
            - If found: it will tack “__#” to the end, with # incrementing for each prior instance
            - If not found: it will use ‘name_basis’ without tacking anything onto it
        - Because, the user may not know the number that is used, aceso.resource_request() will return the nickname it winds up using as part of a tuple of return values
    - False - Interpret ‘name_basis’ as a fixed name/key, use it as-is
- scan_barcode (str/bool):
    - ‘yes’/True - Scan the barcode between the LPX and the target_destination
    - ‘no’/False - Do not scan the barcode

Details: (none)

“Ss.stow” (lpx_stow)

Moves a plate from the transfer station to the first available slot on the carousel
Details:

- “plate_name” (str): Name of the container in the wellplates/consumables collection
    - if None, the step’s associated container, plate type, and location will be looked up from the database.

- “plate_type” (str): (Optional) The container type (e.g. “96 Well Microplate", "96 Well DeepWell")
    - Defaults to the value looked up if “plate_name” is None. If “plate_name” is not None, but this parameter is not specified, the value will be looked up using the plate name provided.
- “plate_oloc” (str): (Optional) The old location of the plate.
    - Defaults to the value looked up if “plate_name” is None. If “plate_name” is not None, but this parameter is not specified, the value defaults to LPX transfer location.

“Th.prepare_to_send_receive” (thermo_prepare)

Sets the thermo-reactor into a state to send or receive a plate (door open, piston up)
Details: (none)

“Th.run” (thermo_run)

Performs a reaction according to a specified heating profile
Details:

- “heating_profile” (dict): A dictionary containing specifications for a run
    - “evap_hold_time” (int): #
    - “evap_temperature” (int): Temperature of evaporation (Celcius)
    - “heat_hold_time” (int): #
    - “heat_temperature” (int): Temperature of reaction (Celcius)
    - “safe_temperature” (int): Temperature at which door can be opened (Celcius)
    - “venting_time” (int): #

“Th.go_to_initial_state” (thermo_initial_state)

Sets the thermoreactor into a safe state (door closed, piston up)
Details: (none)

“Pr.prepare_to_send_or_receive” (prepare_to_receive_and_send)

Move the plate out from the left door on the tecan
Details:

- “direction” (str): Either ‘send’ or ‘receive’ for if the Spark is sending or receiving a plate
    - E.g. ‘send’ - the plate is moving from the Spark to the Liquid Handler

“Pr.cancel_job” (cancel_job)

Attempts to cancel a job on the plate reader
NOTE: uses “container” key for the job id being canceled
Details: (none)

“Pr.run_platereader” (run_platereader)

Executes a script on the Tecan Spark
Details:

- ‘assay’ (str): The type of scan being performed

    - “peaks”:
    - “logd”:
    - “logpi”:
    - “oxideg”:
    - “photodeg”:
    - “hyperpol”:
    - “tox”:
- The following arguments are not needed if a call to the Tecan Spark with the same queue name and “assay” property has already been run. Omission will lookup the same properties from the previous; individuals keys will overwrite the existing values in the method–should only be used for “offset”
    - ‘mode’ (str): What scanning mode the Spark should use
        - “abs”: Absorbance scan
        - “pl”: Fluorescence
        - “lum”: Photoluminescence
        - “preheat”: (implementation pending)
    - ‘start’ (int): (Optional, default 300) Wavelength at which to start the scan (either nm or 1/10th of nm)
    - ‘stop’ (int): (Optional, default 900) Wavelength at which to end the scan (either nm or 1/10th of nm)
    - ‘step’ (int): (Optional, default 2) Wavelength resolution for the scan (either nm or 1/10th of nm)
    - ‘excite’ (int): If PL, the excitation wavelength (either nm or 1/10th of nm); setting this parameter can cause confusion with the mode
    - ‘settle_time’ (int): (Optional, default 50) Number of ms to wait between wells before scanning
    - ‘iterations’ (num): (Optional, default 1) The number of scans to perform
    - ‘interval’ (num): (Optional, default -1) The number of minutes to wait/expose between scans. (Any value less than or equal to 0 is seen as “no wait”)
    - ‘is_photo’ (bool): (Optional, default False) True - use photoreactor (interval is exposure time), False - do not use photoreactor (interval is wait duration)
    - ‘offset’ (num): (Optional, default 0) Starts the iteration counter from a higher value (used when multiple scans are needed with liquid handling between; e.g. LogP, Solvatochromism)
    - ‘temperature’ (num): (Optional, default -1) The setpoint for the Spark (-1 if no setpoint)
    - ‘photo_temp’ (num): (Optional, default -1) The setpoint for the photoreactor
        - -1: no temperature control at all (no logging; no gas flow)
        - 0: the “off” state(no logging; no gas flow)
        - 1-10: temperature cannot be controlled (logging; no gas flow)
        - 10-27: temperature cannot be controlled (logging; gas flow)
        - 27-60: temperature can be controlled (logging; gas flow)
    - ‘tol’ (num or tuple): (Optional, default 05.) If number, it is the tolerance from the setpoint; if a tuple, it is the acceptable temperature range.
    - ‘off_when_done’ (bool): True - turns temperature control off when the scan completes; False - leaves the temperature control on when the scan completes.
    - ‘wait_for_t’ (bool): True - wait for T to be within tol before starting the scan, False - start scan immediately
    - ‘excluded_wells’ (list): Allows wells to be manually excluded from a scan, this will stack on top of wells which have no well contents when determining which wells to scan (handles a well ID being in both lists)

“Pr.check_wellpalte_signal” (check_platereader)

Verifies that a plate’s signal intensities are within the desired range and attempts to make corrections if possible.
Details:

- Inherits all arguments from “Pr.run_platereader” (though loop properties should be left unmodified)
- The “assay” property can be set to “check” for safety to prevent method parameters switching around—as method details will look up any extant method specifications from another run of the same queue and assay name (if absent, it will default to “check”)
- Additional arguments are requested for a plate check, their default values are for an absorbance scan, so they are optional for those but required for other scan types:
    - ‘mode’ should be specified, ‘assay’ will default to “check”
    - ‘lb’ (num): Lower bound for the acceptable range for a signal (peak prominence)
        - Default = 0.5
    - ‘ub’ (num): Upper bound for the acceptable range for a signal (peak prominence)
        - Default = 2.0
    - ‘wavelengths’ (tuple): A tuple of the form [lower, upper] for the wavelength range to analyze
        - Default = [200, 1000], the entire range. Values can exceed those of the range actually scanned.
    - ‘target’ (num): (Optional, default is the power mean of lb and ub), the desired/optimal signal value
    - ‘iter’ (num): (Optional, default -1) The number of times to try to rectify the plate
        - -1: Retry until a final signal is sent by the liquid handler
        - 0: Do not attempt rectification
        - >0: Retry at most this many times

“Pr.analyze_data” (analyze_platereader)

Calculates a property from spectral data.
Details:

- ‘assay’ (str): The type of analysis being performed
    - “peaks”:
    - “logd”:
    - “logpi”:
    - “oxideg”:
    - “photodeg”:
    - “hyperpol”:
    - “tox”:
- ‘mode’ (str): This is used to lookup the right data to perform the analysis on
    - “abs”:
    - “pl”:
- ‘from_assay’ (str): The type of assay the data was originally from
    - Mostly to allow ‘peaks’ to be run on the 0th iteration of another assay
- (Options for the different assays will be added if needed)
- (TODO: Add options to allow for different types of background subtraction) Currently, there are methods for the following types of background subtraction—just no way to select if you’d like to use something other than the default for the given assay
    - Wellplate background

- Solvent by volume
- Solvent by solvent peaks
- Signal baseline (less than uncertainty of measurement, so not needed)
- Reference well(s)
  - A single well ID - (e.g “H12”) it will subtract the signal of H12 from all other wells
  - A well row - (e.g. “A:”) it will subtract the signal of the well A1 from all other wells in column 1, well A2 from all other wells in column 2, and so forth.
  - A well column - (e.g. “:12”) like the previous, but will treat each well in column 12 as the reference for the corresponding row of other wells.
  - Total plate - (i.e. “::”) this will make all wells their own reference
  - References can be static or dynamic
    - Static, the reference of a well is always its first scan
    - Dynamic, the reference of a well is the current scan
  - It can be specified if the signal used as the reference should be the raw or processed signal
- Calling project_iterator with method = `lambda *_, **__: None` can be used to floor the data