**Supplementary File 1**

A copy of the Python scripts described in the protocol is included below. Please refer to the GitHub repository for the latest versions.

**NoSP3\_digestion.py**

from opentrons import protocol\_api

metadata = {

'protocolName': 'Digestion Protocol 2mL Tubes',

'author': 'Cody',

'description': 'Digestion protocol for use with 2mL tubes',

'apiLevel': '2.8'

}

def run(protocol: protocol\_api.ProtocolContext):

# ---------------------------- CUSTOMIZE HERE ONLY ---------------------------- |

number\_of\_samples: int = 1 # specify the number of protein samples

sample\_concentrations = [2.0] # specify the concentration of protein samples (unit is ug/uL);length of sample\_concentrations list must match the number of the samples above;separate concentrations with period sign if sample n>1 (e.g. sample\_concentrations=[2.0, 2.5] if sample n=2).

replicates: int = 9 # specify the number of replicates

volume\_of\_DTT: float = 10.0 # manually prepare 60mM DTT in MS-grade water

volume\_of\_IAA: float = 10.0 # manually prepare 375mM IAA in MS-grade water

volume\_of\_trypsin: float = 10.0 # manually prepare to a concentration of 0.2ug/uL

incubation\_time\_DTT = 30 # in minutes

incubation\_time\_IAA = 30 # in minutes

starting\_tip\_p50 = 'A1' # change if full tip rack will not be used

starting\_tip\_p300 = 'A1' # change if full tip rack will not be used

# | ---------------------------- ^^^^^^^^^^^^^^^^^^^ ---------------------------- |

# ---------------------------- DO NOT EDIT BELOW THIS LINE ---------------------------- #

# Check for valid inputs

if len(sample\_concentrations) != number\_of\_samples:

raise ValueError('Length of sample\_concentrations must match the integer specified for number\_of\_samples.')

if number\_of\_samples \* replicates > 24:

raise ValueError('Total digests (including replicates) cannot exceed the number of slots available on the aluminum block (24).')

# | --------- tip racks --------- |

tiprack\_300 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 2)

tiprack\_50 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 1)

# | --------- pipettes --------- |

p300 = protocol.load\_instrument('p300\_single', 'right', tip\_racks=[tiprack\_300])

p50 = protocol.load\_instrument('p50\_single', 'left', tip\_racks=[tiprack\_50]) ##change p50 to p20 if p20 will be used here and throughout the script following OT-2 API; this script has not been tested with p20 therefore testing is required.”

p50.starting\_tip = tiprack\_50.well(starting\_tip\_p50)

p300.starting\_tip = tiprack\_300.well(starting\_tip\_p300)

# | --------- tube racks/plates/containers --------- |

temp\_mod = protocol.load\_module('Temperature Module', 10)

temp\_plate = temp\_mod.load\_labware('opentrons\_24\_aluminumblock\_nest\_2ml\_snapcap')

tuberack\_2mL = protocol.load\_labware('opentrons\_24\_tuberack\_nest\_2ml\_snapcap', 4)

tuberack\_15ml\_50ml = protocol.load\_labware('opentrons\_10\_tuberack\_falcon\_4x50ml\_6x15ml\_conical', 5)

# | --------- reagents --------- |

DTT = tuberack\_2mL['A6']

IAA = tuberack\_2mL['B6']

trypsin = tuberack\_2mL['C6']

ABC = tuberack\_15ml\_50ml['A3']

samples = tuberack\_2mL.wells()[:number\_of\_samples]

# ---------------------------- COMMANDS ---------------------------- #

# | --------- transfer samples to plate --------- |

protocol.pause('Ensure to change starting tip position for p50 and p300.')

for i in range(number\_of\_samples):

# transfer ABC; change 50 to 20 if p20 will be used.

if (100 - (100 / sample\_concentrations[i])) > 50:

p300.transfer(

100 - (100 / sample\_concentrations[i]),

ABC,

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

new\_tip='once',

touch\_tip=True

)

else:

p50.transfer(

100 - (100 / sample\_concentrations[i]),

ABC,

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

new\_tip='once',

touch\_tip=True

)

# transfer 100ug of protein and mix 3 times with 50 uL volume; change 50 to 20 if p20 will be used.

if (100 / sample\_concentrations[i]) > 50:

p300.transfer(

100 / sample\_concentrations[i],

samples[i],

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

mix\_after=(3, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

else:

p50.transfer(

100 / sample\_concentrations[i],

samples[i],

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

mix\_after=(3, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

# | --------- transfer DTT to plate --------- |

# change the change the mix volume from 50 to 20 if p20 will be used.

protocol.pause('Ensure DTT has been loaded into A6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

p50.transfer(

volume\_of\_DTT,

DTT,

temp\_plate.wells()[:number\_of\_samples \* replicates],

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

protocol.pause('Ensure to close caps on sample tubes.')

# | --------- first incubation --------- |

temp\_mod.set\_temperature(55)

protocol.delay(minutes=5, msg='Pausing for 5 minutes to allow samples to reach tempeature.')

protocol.delay(minutes=incubation\_time\_DTT, msg=f'Incubating at 55 degrees for {incubation\_time\_DTT} minutes.')

# | --------- set block to room temp before adding IAA --------- |

protocol.comment('Cooling down temp block.')

temp\_mod.set\_temperature(22)

protocol.delay(minutes=5, msg='Pausing for 5 minutes to allow tubes to cool down.')

protocol.pause('Ensure to open caps on sample tubes.')

# | --------- transfer IAA to samples on plate --------- |

protocol.pause('Ensure IAA has been loaded into B6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

# change the change the mix volume from 50 to 20 if p20 will be used.

p50.transfer(

volume\_of\_IAA,

IAA,

temp\_plate.wells()[:number\_of\_samples \* replicates],

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

protocol.pause('Close caps on sample tubes and cover tubes with foil')

# | --------- second incubation --------- |

temp\_mod.set\_temperature(22)

protocol.delay(minutes=incubation\_time\_IAA, msg=f'Protect tubes from light. Incubating at 22 degrees for {incubation\_time\_IAA} minutes.')

protocol.comment('Temp block will now be deactivated.')

temp\_mod.deactivate()

# | --------- transfer trypsin to samples on plate --------- |

# change the change the mix volume from 50 to 20 if p20 will be used.

protocol.pause('Ensure trypsin has been loaded into C6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

protocol.pause('Open caps on sample tubes on the temperature module')

p50.transfer(

volume\_of\_trypsin,

trypsin,

temp\_plate.wells()[:number\_of\_samples \* replicates],

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well')

protocol.comment('Transfer to tubes to shaker for overnight digestion.')

**SP3\_peptide\_cleanup.py**

from opentrons import protocol\_api

metadata = {

'protocolName': 'SP3 Peptide Cleanup',

'author': 'Cody',

'description': 'Protocol for peptide cleanup using SP3 magnetic beads',

'apiLevel': '2.8'

}

def run(protocol: protocol\_api.ProtocolContext):

# ---------------------------- CUSTOMIZE HERE ONLY ---------------------------- |

number\_of\_samples: int = 3 # specify the number of protein digest

replicates: int = 2 # specify the number of replicates per sample

transfer\_vol\_peptides: float = 55.0 # specify the volume of digest to be processed; Each protein digest is about 120 uL and we split it into 2 cleanup reactions with each cleanup starting with 55 uL.

volume\_of\_beads: float = 10.0 # Manually prepare beads for peptide binding prior to loading

volume\_of\_ACN: float = 1292.0 # Volume of 100% ACN to be used during peptide binding phase; cannot exceed 1500uL

volume\_of\_DMSO: float = 80.0 # Manually prepare 2% DMSO in MS water.

total\_samples = number\_of\_samples \* replicates # Total number of samples (including replicates) cannot exceed 48

starting\_tip\_p50 = 'A1' # change if full tip rack will not be used

starting\_tip\_p300 = 'A1' # change if full tip rack will not be used

starting\_mag\_well = 0 # 0 corresponds to 'A1' up to 95 corresponding to 'H12'

# | ---------------------------- ^^^^^^^^^^^^^^^^^^^ ---------------------------- |

# ---------------------------- DO NOT EDIT BELOW THIS LINE ---------------------------- #

# | --------- tip racks --------- |

tiprack\_300 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 2)

tiprack\_300\_2 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 3)

tiprack\_50 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 1)

#tiprack\_50\_2 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 6)

# | --------- pipettes --------- |

p300 = protocol.load\_instrument('p300\_single', 'right', tip\_racks=[tiprack\_300, tiprack\_300\_2])

p50 = protocol.load\_instrument('p50\_single', 'left', tip\_racks=[tiprack\_50]) #change p50 to p20 if p20 will be used here and throughout the script following OT-2 API; this script has not been tested with p20 therefore testing is required.

p50.starting\_tip = tiprack\_50.well(starting\_tip\_p50)

p300.starting\_tip = tiprack\_300.well(starting\_tip\_p300)

p300\_aspirate\_slow = 25 # Aspiration speed when removing supernatant

p300\_aspirate\_default = 150 # Normal aspiration speed by default

p300\_aspirate\_fast = 200

p50\_aspirate\_slow = 25 # Aspiration speed when removing supernatant;

p50\_aspirate\_default = 150 # Normal aspiration speed by default;

# | --------- tube racks/plates/containers --------- |

mag\_deck = protocol.load\_module('magdeck', 7)

if mag\_deck.status == 'engaged':

mag\_deck.disengage()

mag\_plate = mag\_deck.load\_labware('nest\_96\_wellplate\_2ml\_deep')

tuberack\_2mL = protocol.load\_labware('opentrons\_24\_tuberack\_nest\_2ml\_snapcap', 4)

tuberack\_15ml\_50ml = protocol.load\_labware('opentrons\_10\_tuberack\_falcon\_4x50ml\_6x15ml\_conical', 5)

# | --------- reagents --------- |

beads = tuberack\_2mL['A6']

DMSO = tuberack\_15ml\_50ml['A1']

ACN = tuberack\_15ml\_50ml['A3']

waste = tuberack\_15ml\_50ml['B3']

samples = tuberack\_2mL.wells()[:number\_of\_samples]

# ---------------------------- COMMANDS ---------------------------- #

# Check total number of samples and replicates

if (starting\_mag\_well + total\_samples \* 2 > 96):

raise Exception("Well plate does not have the required number of wells to hold all replicates at that starting position.")

# Function for resuspending beads in a given volume of a specified reagent

def reagentTransfer(vol, reagent, wells=mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]):

for well in wells:

p300.pick\_up\_tip()

p300.transfer(

vol,

reagent,

well.top() if reagent == ACN else well,

mix\_before=(3, 100) if reagent == DMSO else None,

air\_gap=10,

touch\_tip=True if reagent == DMSO else False,

new\_tip='never',

blow\_out=True,

blowout\_location='destination well',

)

p300.mix(10, vol if vol < 300 else 300, well.bottom(1))

p300.touch\_tip()

p300.blow\_out()

p300.drop\_tip()

# Function for mixing resuspended beads to mimic mixing on a plate shaker

def mixWells(mix\_vol, num\_mixes, delay\_min, wells=mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]):

curr\_mix = 0

while curr\_mix < num\_mixes:

protocol.delay(minutes=delay\_min)

for well in wells:

p300.pick\_up\_tip()

p300.mix(10, mix\_vol if mix\_vol < 300 else 300, well.bottom(1))

p300.blow\_out()

p300.touch\_tip()

p300.drop\_tip()

curr\_mix += 1

Transfer defined mass of peptide from sample to the plate on magnetic module

for i in range(len(samples)):

p300.flow\_rate.aspirate = p300\_aspirate\_slow

p300.flow\_rate.dispense = p300\_aspirate\_slow

p300.transfer(

transfer\_vol\_peptides,

samples[i],

mag\_plate.wells()[i \* replicates + starting\_mag\_well: i \* replicates + replicates + starting\_mag\_well],

touch\_tip=True,

new\_tip='once',

blow\_out=True,

blowout\_location='destination well'

)

p300.flow\_rate.aspirate = p300\_aspirate\_default

p300.flow\_rate.dispense = p300\_aspirate\_default

# Transfer beads, then ACN to the tubes with peptide samples

protocol.pause('Ensure prepared beads have been loaded into A6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

p50.flow\_rate.aspirate = p50\_aspirate\_default

p50.flow\_rate.dispense = p50\_aspirate\_default

#change p50 to p20 if needed

p50.transfer(

volume\_of\_beads,

beads,

mag\_plate.wells()[starting\_mag\_well:total\_samples + starting\_mag\_well],

mix\_before=(5, 50),

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

reagentTransfer(volume\_of\_ACN, ACN)

mixWells(mix\_vol=volume\_of\_ACN, num\_mixes=5, delay\_min=0.0)

mag\_deck.engage()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes.')

# Remove supernatant after initial incubation

# Reduce aspiration speed prior to removing supernatant

p300.flow\_rate.aspirate = p300\_aspirate\_slow

p300.flow\_rate.dispense = p300\_aspirate\_default

for mag\_well in mag\_plate.wells()[starting\_mag\_well:total\_samples + starting\_mag\_well]:

p300.pick\_up\_tip()

p300.transfer(

volume\_of\_ACN \* 1.1,

mag\_well.bottom(1),

waste.top(),

air\_gap=10,

new\_tip='never'

)

p300.touch\_tip()

p300.blow\_out(waste)

p300.drop\_tip()

# Return aspiration speed back to default before moving on in the protocol execution

p300.flow\_rate.aspirate = p300\_aspirate\_default

mag\_deck.disengage()

# # Wash beads with 1mL ACN

protocol.pause('make sure ACN tube caps are off')

reagentTransfer(1000, ACN)

mixWells(mix\_vol=1000, num\_mixes=1, delay\_min=0)

mag\_deck.engage()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes.')

# Remove supernatant after wash incubation

# Reduce aspiration speed prior to removing supernatant

p300.flow\_rate.aspirate = p300\_aspirate\_slow

for mag\_well in mag\_plate.wells()[starting\_mag\_well:total\_samples+starting\_mag\_well]:

p300.pick\_up\_tip()

p300.transfer(

1000 \* 1.1,

mag\_well.bottom(1),

waste.top(),

air\_gap=10,

new\_tip='never'

)

p300.blow\_out(waste)

p300.drop\_tip()

# Return aspiration speed back to default before moving on in the protocol execution

p300.flow\_rate.aspirate = p300\_aspirate\_default

protocol.delay(seconds=60, msg='Delaying for 60 seconds to allow residual ACN to evaporate.')

mag\_deck.disengage()

# # Peptide elution

# Transfer 2% DMSO to samples

protocol.pause('vortex DMSO again and open caps.')

reagentTransfer(volume\_of\_DMSO, DMSO)

mixWells(mix\_vol=volume\_of\_DMSO, num\_mixes=4, delay\_min=0)

mag\_deck.engage()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes.')

# # Transfer first elution volumes to empty wells on the plate

# # Reduce aspiration speed prior to removing supernatant

p300.flow\_rate.aspirate = p300\_aspirate\_slow

for mag\_well, dest\_well in zip(mag\_plate.wells()[starting\_mag\_well:total\_samples + starting\_mag\_well],

mag\_plate.wells()[total\_samples + starting\_mag\_well:total\_samples \* 2 + starting\_mag\_well]):

p300.pick\_up\_tip()

p300.transfer(

volume\_of\_DMSO \* 1.2,

mag\_well.bottom(0.5),

dest\_well,

new\_tip='never',

blow\_out=True,

blowout\_location='destination well'

)

p300.drop\_tip()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes to remove any residual beads in solution.')

# # Transfer final elution volumes to new tubes on the 2mL tube rack

# protocol.pause('Ensure enough 2mL LoBind tubes are in the 2mL tube rack to match total number of samples')

for mag\_well, dest\_well in zip(mag\_plate.wells()[total\_samples + starting\_mag\_well: total\_samples\*2 + starting\_mag\_well],

tuberack\_2mL.wells()[number\_of\_samples:number\_of\_samples + total\_samples]):

p300.pick\_up\_tip()

p300.transfer(

volume\_of\_DMSO \* 1.1,

mag\_well,

dest\_well,

new\_tip='never',

blow\_out=True,

blowout\_location='destination well'

)

p300.drop\_tip()

# Return aspiration speed back to default before moving on in the protocol execution

p300.flow\_rate.aspirate = p300\_aspirate\_default

mag\_deck.disengage()

# Final check to disengage magnetic module if it hasn't disengaged

if mag\_deck.status == 'engaged':

mag\_deck.disengage()

**SP3\_digestion.py**

from opentrons import protocol\_api

metadata = {

'protocolName': 'SP3 Protein Cleanup and Digestion',

'author': 'Cody',

'description': 'Digestion protocol with SP3 detergent removal',

'apiLevel': '2.8'

}

def run(protocol: protocol\_api.ProtocolContext):

# ---------------------------- CUSTOMIZE HERE ONLY ---------------------------- |

number\_of\_samples: int = 1 # Specify the number of protein samples

sample\_concentrations = [5.00] # specify the concentration of protein samples (unit is ug/uL); Length of sample\_concentrations list must match the number of the samples above; separate concentrations with period sign if sample n>1 (e.g. sample\_concentrations=[2.0, 2.5] if sample n=2).

replicates: int = 3 # specify the number of replicates for each sample

volume\_of\_DTT: float = 10.0 # manually prepare 60mM DTT in MS-grade water

volume\_of\_IAA: float = 10.0 # manually prepare 375mM IAA in MS-grade water

volume\_of\_trypsin: float = 10.0 # manually prepare to a concentration of 0.2ug/uL

incubation\_time\_DTT = 30 # in minutes

incubation\_time\_IAA = 30 # in minutes

volume\_of\_beads: float = 20.0 # Manually prepare beads for peptide binding prior to loading

volume\_of\_ethanol100: float = 140.0 # Volume of 100% ethanol to be used during protein binding phase

volume\_of\_ethanol80: float = 1000.0 # Volume of 80% ethanol to be used for washes

total\_samples = number\_of\_samples \* replicates # Total number of samples (including replicates) cannot exceed 24

starting\_tip\_p50 = 'A1' # change if full tip rack will not be used

starting\_tip\_p300 = 'A1' # change if full tip rack will not be used

starting\_mag\_well = 0 # 0 corresponds to 'A1' up to 95 corresponding to 'H12'

# | --------- tip racks --------- |

tiprack\_300 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 3)

tiprack\_300\_2 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 6)

tiprack\_50 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 1)

tiprack\_50\_2 = protocol.load\_labware('opentrons\_96\_tiprack\_300ul', 2)

# | --------- pipettes --------- |

#p300 = protocol.load\_instrument('p300\_single', 'right', tip\_racks=[tiprack\_300])

p300 = protocol.load\_instrument('p300\_single', 'right', tip\_racks=[tiprack\_300, tiprack\_300\_2])

p50 = protocol.load\_instrument('p50\_single', 'left', tip\_racks=[tiprack\_50, tiprack\_50\_2]) #change p50 to p20 if p20 will be used here and throughout the script following OT-2 API; this script has not been tested with p20 therefore testing is required”

p50.starting\_tip = tiprack\_50.well(starting\_tip\_p50)

p300.starting\_tip = tiprack\_300.well(starting\_tip\_p300)

p300\_aspirate\_slow = 25 # Aspiration speed when removing supernatant

p300\_aspirate\_default = 150 # Normal aspiration speed by default

# | --------- tube racks/plates/containers --------- |

temp\_mod = protocol.load\_module('Temperature Module', 10)

temp\_plate = temp\_mod.load\_labware('opentrons\_24\_aluminumblock\_nest\_2ml\_snapcap')

tuberack\_2mL = protocol.load\_labware('opentrons\_24\_tuberack\_nest\_2ml\_snapcap', 4)

tuberack\_15ml\_50ml = protocol.load\_labware('opentrons\_10\_tuberack\_falcon\_4x50ml\_6x15ml\_conical', 5)

mag\_deck = protocol.load\_module('magdeck', 7)

if mag\_deck.status == 'engaged':

mag\_deck.disengage()

mag\_plate = mag\_deck.load\_labware('nest\_96\_wellplate\_2ml\_deep')

# | --------- reagents --------- |

samples = tuberack\_2mL.wells()[:number\_of\_samples]

DTT = tuberack\_2mL['A6']

IAA = tuberack\_2mL['B6']

trypsin = tuberack\_2mL['C6']

beads = tuberack\_2mL['D6']

ABC = tuberack\_15ml\_50ml['A1']

ethanol100 = tuberack\_15ml\_50ml['A3']

ethanol80 = tuberack\_15ml\_50ml['A4']

waste = tuberack\_15ml\_50ml['B3']

# ---------------------------- COMMANDS ---------------------------- #

# Check well plate for adequate number of wells available after the starting well

if (starting\_mag\_well + total\_samples > 95):

raise Exception("Well plate does not have the required number of wells to hold all replicates at that starting position.")

# Function for resuspending beads in a given volume of a specified reagent

def reagentTransfer(vol, reagent, wells=mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]):

for well in wells:

p300.pick\_up\_tip()

p300.transfer(

vol,

reagent,

well.top() if reagent == ethanol80 else well,

air\_gap=10,

new\_tip='never',

blow\_out=True,

blowout\_location='destination well',

)

p300.mix(10, vol if vol < 300 else 300, well.bottom(1))

p300.blow\_out()

p300.drop\_tip()

# Function for mixing resuspended beads to mimic mixing on a plate shaker

def mixWells(mix\_vol, num\_mixes, delay\_min, wells=mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]):

curr\_mix = 0

while curr\_mix < num\_mixes:

protocol.delay(minutes=delay\_min)

for well in wells:

p300.pick\_up\_tip()

p300.mix(5, mix\_vol if mix\_vol < 300 else 300, well.bottom(1))

p300.touch\_tip()

p300.blow\_out()

p300.drop\_tip()

curr\_mix += 1

# Transfer 100mM ABC then 100ug of protein from samples to tubes on temp plate. Concentration in tubes will be 1 ug/uL

mass\_of\_protein = 100.0

for i in range(number\_of\_samples):

# transfer ABC; change 50 to 20 if p20 will be used

if (mass\_of\_protein - (mass\_of\_protein / sample\_concentrations[i])) > 50:

p300.transfer(

100 - (mass\_of\_protein / sample\_concentrations[i]),

ABC,

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

new\_tip='once',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

else:

p50.transfer(

100 - (mass\_of\_protein / sample\_concentrations[i]),

ABC,

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

new\_tip='once',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

# transfer 100ug of protein and mix 3 times with 50 uL volume; change 50 to 20 if p20 will be used

if (mass\_of\_protein / sample\_concentrations[i]) > 50:

p300.transfer(

mass\_of\_protein / sample\_concentrations[i],

samples[i],

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

mix\_after=(3, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

else:

p50.transfer(

mass\_of\_protein / sample\_concentrations[i],

samples[i],

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

mix\_after=(3, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

# transfer DTT to tubes on temp plate and change the mix volume from 50 to 20 if p20 will be used.

protocol.pause('Ensure DTT has been loaded into A6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

p50.transfer(

volume\_of\_DTT,

DTT,

temp\_plate.wells()[:number\_of\_samples \* replicates],

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

protocol.pause('Ensure to close caps on sample tubes.')

# DTT incubation

temp\_mod.set\_temperature(55)

protocol.delay(minutes=5, msg='Pausing for 5 minutes to allow samples to reach tempeature.')

protocol.delay(minutes=incubation\_time\_DTT, msg=f'Incubating at 55 degrees for {incubation\_time\_DTT} minutes.')

# cool temp block and tubes to room temp prior to adding IAA to samples

protocol.comment('Cooling down temp block.')

temp\_mod.set\_temperature(22)

protocol.delay(minutes=5, msg='Pausing for 5 minutes to allow tubes to cool down.')

protocol.pause('Ensure to open caps on sample tubes.')

# transfer IAA to tubes on temp plate and change the mix volume from 50 to 20 if p20 will be used.

protocol.pause('Ensure IAA has been loaded into B6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

p50.transfer(

volume\_of\_IAA,

IAA,

temp\_plate.wells()[:number\_of\_samples \* replicates],

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

protocol.pause('Close caps on sample tubes and cover tubes with foil')

# IAA incubation

temp\_mod.set\_temperature(22)

protocol.delay(minutes=incubation\_time\_IAA,

msg=f'Protect tubes from light. Incubating at 22 degrees for {incubation\_time\_IAA} minutes.')

protocol.comment('Temp block will now be deactivated.')

temp\_mod.deactivate()

protocol.pause('open tube caps')

#Transfer protein samples from tubes to the deep-well plate on magnetic module

for i in range(total\_samples):

p300.transfer(

120 \* 1.1,

temp\_plate.wells()[i \* replicates: i \* replicates + replicates],

mag\_plate.wells()[(starting\_mag\_well + i \* replicates) : (starting\_mag\_well + i \* replicates + replicates)],

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well'

)

# add beads to samples

protocol.pause('Ensure prepared beads have been loaded into D6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

p50.transfer(

volume\_of\_beads,

beads,

mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well],

mix\_before=(5, volume\_of\_beads),

mix\_after=(5, volume\_of\_beads),

new\_tip='always',

blow\_out=True,

blowout\_location='destination well'

)

protocol.pause('Ensure 100 percent ethanol has been loaded into A3 of the 15mL\_50mL tube rack located in slot 5 prior to resuming protocol.')

reagentTransfer(volume\_of\_ethanol100, ethanol100)

mixWells(mix\_vol=volume\_of\_ethanol100, num\_mixes=5, delay\_min=0)

mag\_deck.engage()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes.')

# Remove supernatant after initial incubation

# Reduce aspiration speed prior to removing supernatant

p300.flow\_rate.aspirate = p300\_aspirate\_slow

# for mag\_well in mag\_plate.wells()[:total\_samples]:

for mag\_well in mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]:

p300.pick\_up\_tip()

p300.transfer(

volume\_of\_ethanol100 \* 1.1,

mag\_well.bottom(1),

waste.top(),

air\_gap=10,

new\_tip='never'

)

p300.touch\_tip()

p300.blow\_out(waste)

p300.drop\_tip()

# Return aspiration speed back to default before moving on in the protocol execution

p300.flow\_rate.aspirate = p300\_aspirate\_default

mag\_deck.disengage()

# Wash beads with 80% ethanol (3 washes in total)

protocol.pause('Ensure 80 percent ethanol has been loaded into A4 of the 15mL\_50mL tube rack located in slot 5 prior to resuming protocol.')

for i in range(3):

if mag\_deck.status == 'engaged':

mag\_deck.disengage()

reagentTransfer(volume\_of\_ethanol80, ethanol80)

mag\_deck.engage()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes.')

# Remove supernatant after wash incubation

# Reduce aspiration speed prior to removing supernatant

p300.flow\_rate.aspirate = p300\_aspirate\_slow

# for mag\_well in mag\_plate.wells()[:total\_samples]:

for mag\_well in mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]:

p300.pick\_up\_tip()

p300.transfer(

volume\_of\_ethanol80 \* 1.1,

mag\_well.bottom(1),

waste.top(),

air\_gap=10,

new\_tip='never'

)

p300.blow\_out(waste)

p300.drop\_tip()

p300.flow\_rate.aspirate = p300\_aspirate\_default

# Wash beads with 250 uL ABC

protocol.pause('Open cap on ABC tube.')

if mag\_deck.status == 'engaged':

mag\_deck.disengage()

reagentTransfer(250, ABC)

mixWells(mix\_vol=250, num\_mixes=0, delay\_min=0)

mag\_deck.engage()

protocol.delay(minutes=2, msg='Incubating on magnet for 2 minutes.')

# Remove supernatant after wash incubation

# Reduce aspiration speed prior to removing supernatant

p300.flow\_rate.aspirate = p300\_aspirate\_slow

# for mag\_well in mag\_plate.wells()[:total\_samples]:

for mag\_well in mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well]:

p300.pick\_up\_tip()

p300.transfer(

250 \* 1.1,

mag\_well.bottom(1),

waste.top(),

air\_gap=10,

new\_tip='never'

)

p300.blow\_out(waste)

p300.drop\_tip()

# Return aspiration speed back to default before moving on in the protocol execution

p300.flow\_rate.aspirate = p300\_aspirate\_default

mag\_deck.disengage()

# resuspend proteins and beads in 100uL of 100mM ABC and move to 2mL tubes for incubation

reagentTransfer(100, ABC)

protocol.pause('Ensure new collection tubes have been placed in 2.0 mL aluminum block prior to resuming protocol.')

p300.transfer(

100 \* 1.5,

# mag\_plate.wells()[:total\_samples],

mag\_plate.wells()[starting\_mag\_well: total\_samples + starting\_mag\_well],

temp\_plate.wells()[number\_of\_samples:number\_of\_samples + total\_samples],

mix\_before=(10, 100),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blow\_out\_location='destination well'

)

# transfer trypsin to each sample and change the mix volume from 50 to 20 if p20 will be used

protocol.pause('Ensure trypsin (0.2ug/uL) has been loaded into C6 of the 2ml tube rack located in slot 4 prior to resuming protocol.')

p50.transfer(

volume\_of\_trypsin,

trypsin,

temp\_plate.wells()[:total\_samples],

mix\_after=(5, 50),

new\_tip='always',

touch\_tip=True,

blow\_out=True,

blowout\_location='destination well')

protocol.comment('Transfer digest tubes to plate shaker for overnight digestion.')