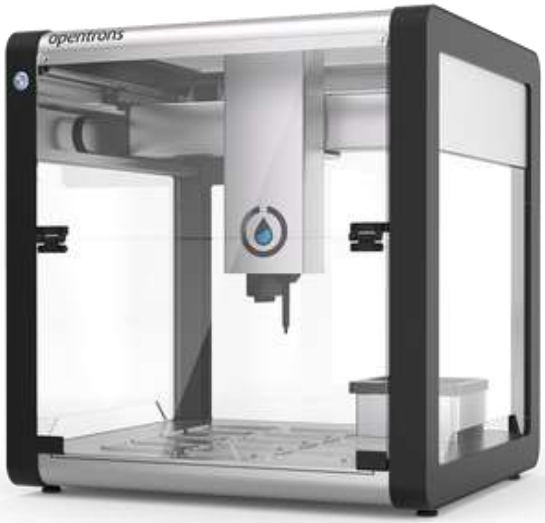


自動分注ロボット OT-2

プロトコル作成について　～基礎編～

funakoshi
FRONTIERS IN LIFE SCIENCE

OT-2使用の流れ

①プロトコル作成　＝　どういった動作をさせるか決める。

（②チップ，ピペットのキャリブレーション）

③プロトコルのアップロード

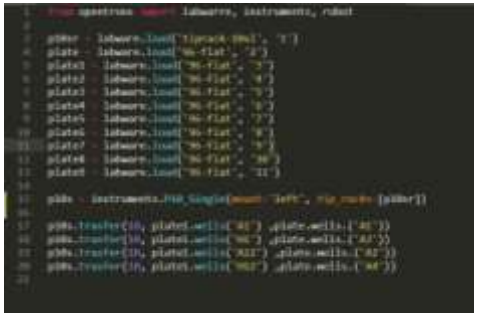
④ポジションチェック（チップラック，プレート，チューブラック）

⑤プログラム実行！

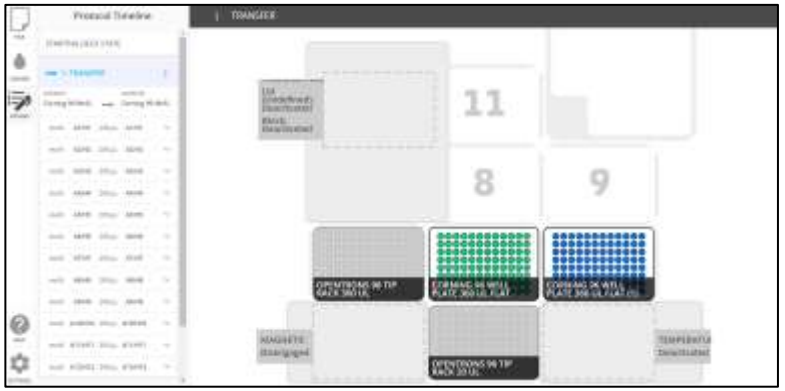
funakoshi
FRONTIERS IN LIFE SCIENCE

OT-2使用の流れ

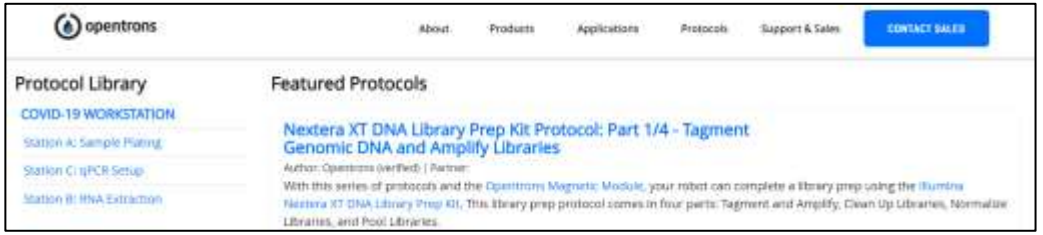
1 : Python APIを用いた作成
⇒自由度が高いプロトコル作成方法
⇒ある程度Pythonの知識が必要



2 : Protocol Designer
⇒モジュールの使用を含めて自分好みのプロトコルを作成できる
⇒ウェブベースで簡単に利用でき，視覚的にもわかりやすい



3 : Opentrons webからダウンロード
⇒自分の目的に当てはまるものがあればすぐに使える



funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

○ テキストベースで自由度が高いプログラムが作成できます

× 作成のルールやコマンドを覚える（調べる）必要があります

```python
h_asp = 1
h_dis = 1

from opentrons import protocol_api

metadata = {'apiLevel': '2.11'}

def run(protocol: protocol_api.ProtocolContext):

    tiprack = protocol.load_labware('opentrons_96_tiprack_300ul', '6')
    tiprack2 = protocol.load_labware('opentrons_96_tiprack_20ul', '9')
    dis_plate= protocol.load_labware('biorad_96_wellplate_200ul_pcr', '3')

    rack1 = protocol.load_labware('biorad_96_wellplate_200ul_pcr', '1')
    rack2 = protocol.load_labware('biorad_96_wellplate_200ul_pcr', '2')

    p300 = protocol.load_instrument('p300_single_gen2', 'left', tip_racks=[tiprack])

    p20 = protocol.load_instrument('p20_single_gen2', 'right', tip_racks=[tiprack2])

    p20.transfer(10, rack1['A1'].bottom(h_asp), dis_plate['A1'].bottom(h_dis))
    p300.transfer(30, rack1['B1'].bottom(h_asp), dis_plate['B1'].bottom(h_dis))
```

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

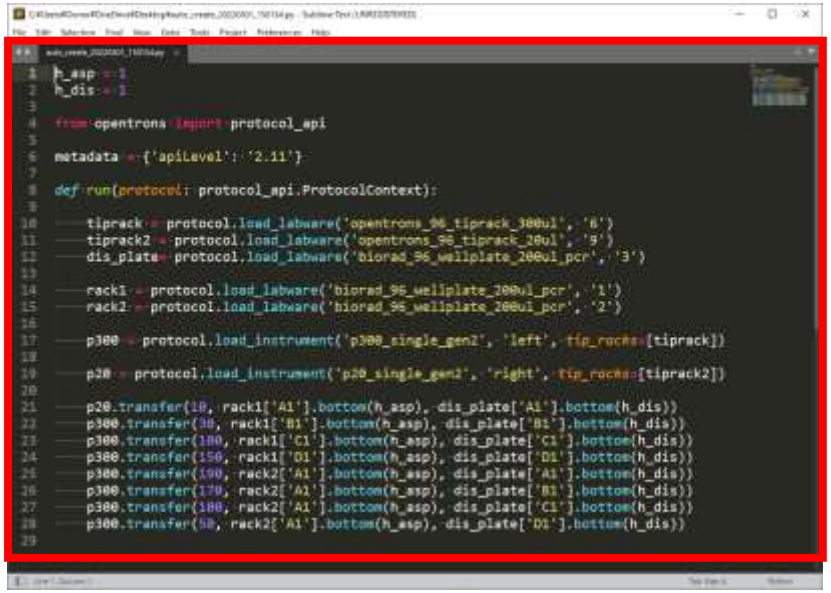
①ソフトウエアの準備

Sublime Text

下記URLより、使用するPCのOSにあったバージョンをダウンロードしてください
Pythonプロトコルを作成するために使用される一般的なコードエディターです

https://www.sublimetext.com/3

Sublime Textを起動すると下記の画面が表示されます
※特にこのソフトでなくても問題ございません



```
h_asp = 1
h_dis = 1

from opentrons import protocol_api

metadata = {'apiLevel': '2.11'}

def run(protocol: protocol_api.ProtocolContext):

    tiprack = protocol.load_labware('opentrons_96_tiprack_300ul', '6')
    tiprack2 = protocol.load_labware('opentrons_96_tiprack_20ul', '9')
    dis_plate = protocol.load_labware('biorad_96_wellplate_200ul_pcr', '3')

    rack1 = protocol.load_labware('biorad_96_wellplate_200ul_pcr', '1')
    rack2 = protocol.load_labware('biorad_96_wellplate_200ul_pcr', '2')

    p300 = protocol.load_instrument('p300_single_gen2', 'left', tip_racks=[tiprack])

    p20 = protocol.load_instrument('p20_single_gen2', 'right', tip_racks=[tiprack2])

    p20.transfer(10, rack1['A1'].bottom(h_asp), dis_plate['A1'].bottom(h_dis))
    p300.transfer(90, rack1['B1'].bottom(h_asp), dis_plate['B1'].bottom(h_dis))
    p300.transfer(100, rack1['C1'].bottom(h_asp), dis_plate['C1'].bottom(h_dis))
    p300.transfer(150, rack1['D1'].bottom(h_asp), dis_plate['D1'].bottom(h_dis))
    p300.transfer(190, rack2['A1'].bottom(h_asp), dis_plate['A1'].bottom(h_dis))
    p300.transfer(170, rack2['A1'].bottom(h_asp), dis_plate['B1'].bottom(h_dis))
    p300.transfer(180, rack2['A1'].bottom(h_asp), dis_plate['C1'].bottom(h_dis))
    p300.transfer(50, rack2['A1'].bottom(h_asp), dis_plate['D1'].bottom(h_dis))
```

- 赤枠の内部にプロトコルをタイプします
- 保存の際は、左上に表示される

  File > Save as　で保存場所とファイル名を決定します
- 保存時の拡張子は【.py】を選択します
- Sublime Text以外のコードエディターを使用した場合も同様に【.py】で保存します

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

②プロトコルの記述方法(基本)

```
from opentrons import protocol_api, types

metadata = {'apiLevel': ‘2.11'}

def run(protocol):


    plate = protocol.load_labware('corning_96_wellplate_360ul_flat', '2')
    tiprack = protocol.load_labware('opentrons_96_tiprack_300ul', '1')

    left_pipette = protocol.load_instrument('p300_single_gen2', 'left', tip_racks=[tiprack])

    left_pipette.pick_up_tip()
    left_pipette.aspirate(100, plate['A1'])
    left_pipette.dispense(100, plate['B2'])
    left_pipette.drop_tip()
```

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

②プロトコルの記述方法(基本)

```
from opentrons import protocol_api, types

metadata = {'apiLevel': ‘2.11'}

def run(protocol):


    plate = protocol.load_labware('corning_96_wellplate_360ul_flat', '2')
    tiprack = protocol.load_labware('opentrons_96_tiprack_300ul', '1')

    left_pipette = protocol.load_instrument('p300_single_gen2', 'left', tip_racks=[tiprack])

    left_pipette.pick_up_tip()
    left_pipette.aspirate(100, plate['A1'])
    left_pipette.dispense(100, plate['B2'])
    left_pipette.drop_tip()
```

ルール：1つのグループはインデント（字下げ）を行う（Pythonのルール）

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

②プロトコルの記述方法(基本)

```
from opentrons import protocol_api, types
```
定型文

```
metadata = {'apiLevel': '2.11'}
```
バージョン管理

```
def run(protocol):
```
定型文

```
    plate = protocol.load_labware('corning_96_wellplate_360ul_flat', '2')
    tiprack = protocol.load_labware('opentrons_96_tiprack_300ul', '1')
```
使用するラボウェアの定義

```
    left_pipette = protocol.load_instrument('p300_single_gen2', 'left', tip_racks=[tiprack])
```
ピペットの定義

```
    left_pipette.pick_up_tip()
    left_pipette.aspirate(100, plate['A1'])
    left_pipette.dispense(100, plate['B2'])
    left_pipette.drop_tip()
```
分注動作の定義

基本形は使用するラボウェア，ピペットを定義し，分注動作を記述していきます

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

③バージョン管理について

```
metadata = {'apiLevel': '2.11'}
```

apiLevel (API version)が重要です
API versionによって対応するOT-2ソフトウエアが異なります
また，API versionによって実装される機能が異なることがあります

API Version	Introduced In OT-2 Software
1.0	3.0.0
2.0	3.14.0
2.1	3.15.2
2.2	3.16.0
・・・	・・・
2.11	4.4.0

できる限り最新バージョンで記述いただく事をお勧めします

最新バージョンや各バージョンの情報は
https://docs.opentrons.com/v2/versioning.html をご覧ください

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

④使用するラボウェア(チップラック，チューブ，プレート)の定義 その1

```
タブ plate = protocol.load_labware('corning_96_wellplate_360ul_flat', '2‘)
タブ tiprack = protocol.load_labware('opentrons_96_tiprack_300ul', '1')
```

```
タブ任意の名前_= protocol.load_labware (‘Opentrons指定の名前’, ’設置場所’)
```

好きな名前をつけます
プログラム内ではこの名前を使用します

Opentrons社のウェブから適切なものを選びます
→次ページへ

1～11の好きな場所を選びます

ルール
1. 装着するピペットに適合するチップラックを必ず用意する。
2. 設置場所は重複できない。
3. 単語中にスペースは使用しない。（Python）
4. 任意の名前の1文字目に数字を用いない。（Python）

funakoshi
FRONTIERS IN LIFE SCIENCE

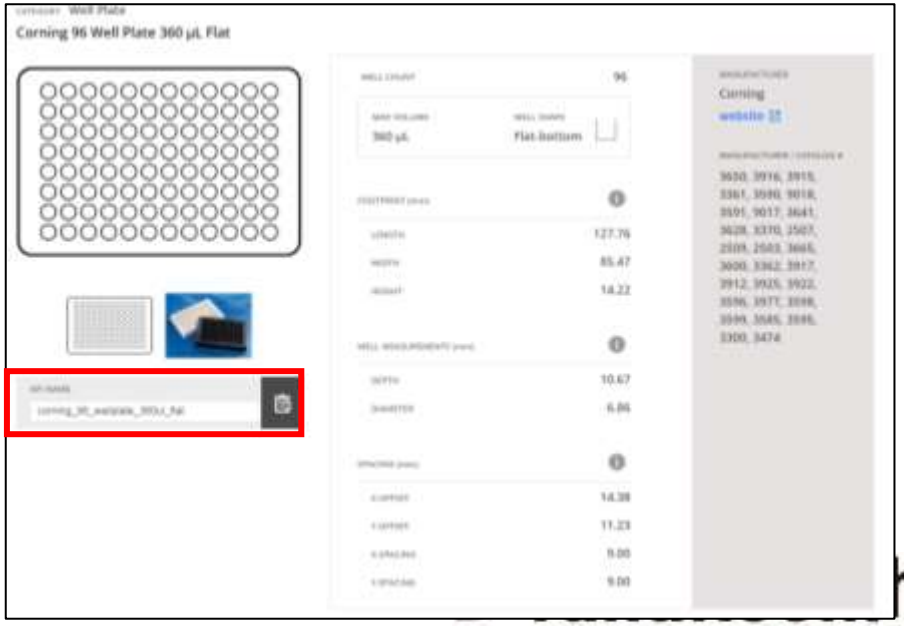
プロトコル作成1：Python APIを用いた作成

④使用するラボウェア(チップラック，チューブ，プレート)の定義 その2

タブ任意の名前_= protocol.load_labware (‘Opentrons指定の名前’, ’設置場所’)

Opentrons指定の名称はOpentrons WEBに設置されているLabware Libraryを参照します
https://labware.opentrons.com/

各ラボウェアに記載されている「API NAME」を記述します
例： corning_96_wellplate_360ul_flat
opentrons_96_aluminumblock_generic_pcr_strip_200ul



プロトコル作成1：Python APIを用いた作成

⑤使用するピペットの定義

```
タブleft_pipette = protocol.load_instrument('p300_single_gen2', 'left', tip_racks=[tiprack])
```

```
タブ任意の名前
        = protocol.load_instrument(‘ピペットの種類’, ‘左右’, tip_racks=[チップラック名])
```

装着するピペットの
種類を指定します
下記表を参照

ピペットを装着するマウントを指定します
【left】か【right】で指定します

チップラックの定義で
指定した任意の名前を記載します

Pipette Type	Model Name
P20 Single GEN2 (1 - 20 µL)	'p20_single_gen2'
P300 Single GEN2 (20 - 300 µL)	'p300_single_gen2'
P1000 Single GEN2 (100 - 1000 µL)	'p1000_single_gen2'
P20 Multi GEN2 (1 - 20 µL)	'p20_multi_gen2'
P300 Multi GEN2 (20 - 300 µL)	'p300_multi_gen2'

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

⑤分注動作の定義 その1

```
タブleft_pipette.pick_up_tip()
タブleft_pipette.aspirate(100, plate['A1'])
タブleft_pipette.dispense(100, plate['B2'])
タブleft_pipette.drop_tip()
```

タブピペット名.動作(‘容量’, ’場所’)

使用するピペットを指定します
ピペットの定義で指定した名前を使用します

動作を指定します

動作	記述法
チップの装着	pick_up_tip
チップの取り外し	drop_tip
吸引	aspirate
排出	dispense

動作する場所

funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成1：Python APIを用いた作成

⑤分注動作の定義 その2

タブピペット名.動作(‘容量’, ’場所’)

タブleft_pipette.pick_up_tip()　　チップの装着

タブleft_pipette.aspirate(100, plate['A1’])　　left_pipetteでプレートのA1から100ulを吸引

タブleft_pipette.dispense(100, plate['B2’])　　left_pipetteでプレートのB2へ100ulを排出

タブleft_pipette.drop_tip()　　チップの取り外し

タブleft_pipette.**transfer**(100, plate[‘A1’],　plate['B2'])

P10でプレートのA1ウェルから10ulを吸引し、プレートのB2ウェルへ排出

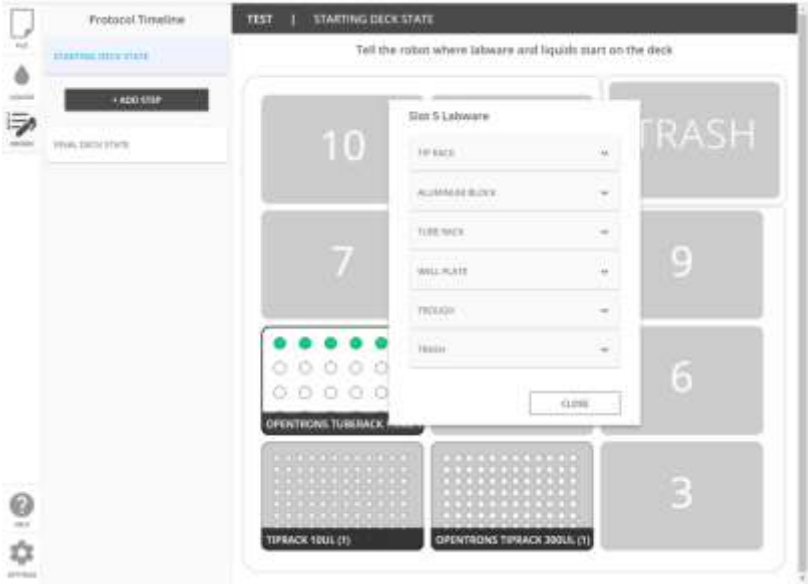
その他多種の動作コマンドが用意されています
最新情報、詳細情報は
https://docs.opentrons.com/v2/　をご確認ください

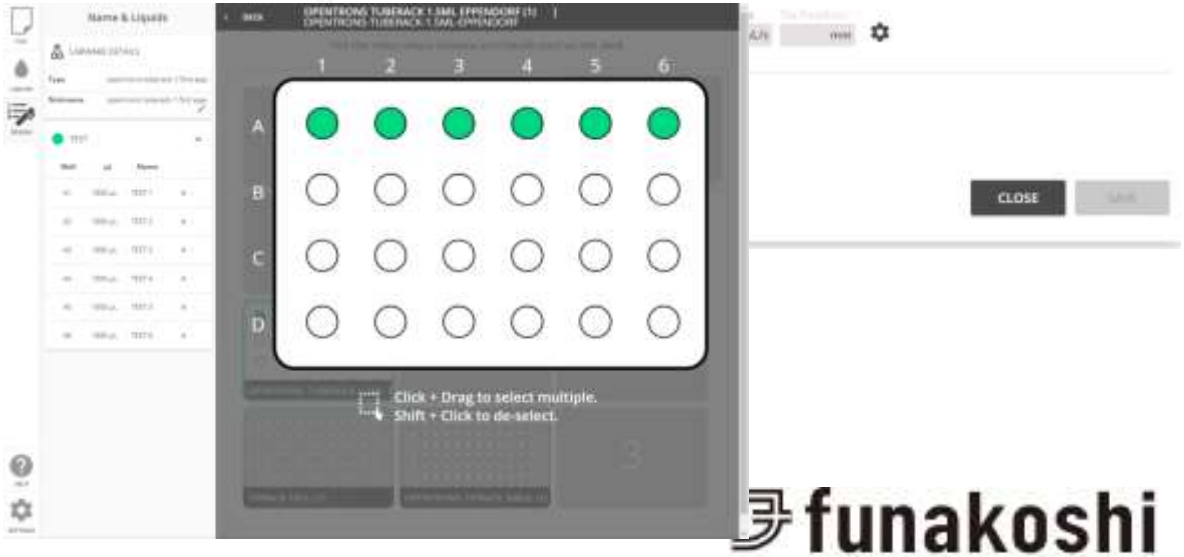
プロトコル作成2：Protocol Designerを用いた作成

○ 直感的な操作でプログラムを作成できるwebアプリケーションです

× PCごとにユーザー登録が必要です
Python API詳細設定ができない場合があります
編集はProtocol Designer内のみで可能です(テキストベースには変換できません)

Protocol DesignerへはアクセスはOpentrons webのトップページからアクセスできます



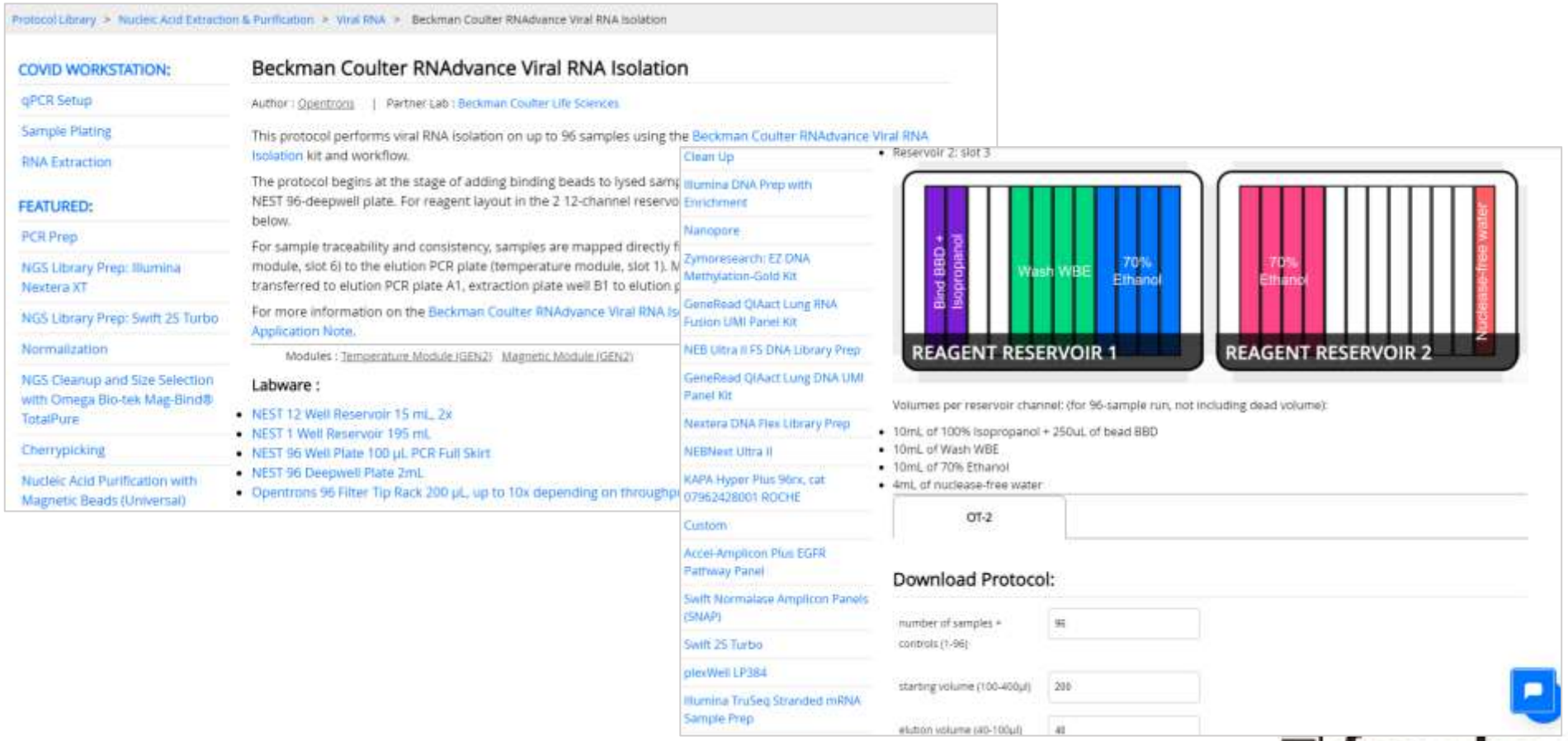


funakoshi
FRONTIERS IN LIFE SCIENCE

プロトコル作成3：Opentrons webからダウンロード

○ ダウンロードするだけで使用できます

× 目的に合ったプロトコルが無い可能性があります
微調整を行う場合、ダウンロードしたプロトコルをPython APIでの編集が必要



funakoshi
FRONTIERS IN LIFE SCIENCE