Installation Prerequisites for the Tutorial

1. First, you need to install Python on your device. (https://wiki.python.org/moin/BeginnersGuide/Download)

2. It is recommended that you create a separate virtual environment to run Qiskit. (https://docs.python.org/3.10/tutorial/venv.html)

3. Once you create the virtual environment to run Qiskit, activate the environment to install necessary packages.

4. Install Qiskit SDK using:

   pip install qiskit

   pip install qiskit-ibm-runtime

5. If you want to run a Jupyter notebook, you will need to install Jupyter in your environment as well.

   pip install jupyter

   and open your notebook with:

   jupyter notebook path/to/notebook.ipynb

6. You can also download and use VS Code (https://code.visualstudio.com/download)

7. You **need to create an IBM Quantum account** to access IBM Quantum backends and submit jobs in simulator. Get started here: https://docs.quantum.ibm.com/guides/setup-channel (for more details on setting up the account, see below)

8. Refer to **ISMB 2024 tutorial** here to get updated on requirements and exercises: https://github.com/IBM/qml4omics/tree/Tutorial_ISMB24

Instructions are compiled from the page (https://docs.quantum.ibm.com/start/install)

A detailed video instructions can be found here https://www.youtube.com/watch?v=dZWz4Gs_BuI

Set up an IBM Quantum channel

IBM offers access to quantum computers via open access plan.

Since we have already set up our environment for Qiskit and Qiskit Runtime, now we need to create an IBM Quantum account.

1. Go to the page https://quantum.ibm.com/ and create an account by clicking on "Create an IBMid" and following the instructions.

2. You will be prompted to a screen to enter your email and other basic information, then verify your email address.

3. Your user account is associated with one or more access instances (in the form hub / group / project) that give access to IBM Quantum services. Additionally, a unique token is assigned to each account, allowing for IBM Quantum access from Qiskit.

4. When you log into your IBM Quantum account page, you will be able to copy your unique API token associated with your account.

5. You can save your credentials locally for easy access to Qiskit Runtime Services:

```
from qiskit_ibm_runtime import QiskitRuntimeService

# Save an IBM Quantum account and set it as your default account.
QiskitRuntimeService.save_account(channel="ibm_quantum",
token="<MY_IBM_QUANTUM_TOKEN>", set_as_default=True)

# Load saved credentials
service = QiskitRuntimeService()
```

- If you save your credentials to disk, you can use QiskitRuntimeService() in the future to initialize your account. The channel parameter distinguishes between different account types. If you are saving multiple accounts per channel, consider using the name parameter to differentiate them.

- Credentials are saved to $HOME/.qiskit/qiskit-ibm.json. Do not manually edit this file.

- If you don't save your credentials, you must specify them every time you start a new session.

- The channel parameter allows you to distinguish between different account types. When initializing the account, IBM Cloud is the default account used if have saved credentials for an IBM Quantum Platform and an IBM Cloud account.

Let's test the setup to make sure that everything works correctly!

```
from qiskit import QuantumCircuit
from qiskit_ibm_runtime import QiskitRuntimeService, SamplerV2 as Sampler

# Create empty circuit
example_circuit = QuantumCircuit(2)
example_circuit.measure_all()

# You'll need to specify the credentials when initializing QiskitRuntimeService,
if they were not previously saved.
service = QiskitRuntimeService()
backend = service.least_busy(operational=True, simulator=False)

sampler = Sampler(backend)
job = sampler.run([example_circuit])
print(f"job id: {job.job_id()}")
result = job.result()
print(result)
```