

Connecting to BioHPC from outside of Cornell network

Table of Contents

Who should read this.....	1
Prerequisite: Set up 2-factor authentication with BioHPC	1
Overview: Two general options for connecting from off campus	2
About login servers.....	3
Connecting from Windows Computers using MobaXterm.....	3
1. Using MobaXterm for basic SSH (quick start).....	3
2. Transferring files with MobaXterm	4
3. Using MobaXterm to SSH to remote server (using login server as ‘jump host’)	5
4. Using MobaXterm for VNC	6
5. Using MobaXterm for services connected to other ports (VNC, Rstudio, jupyter, custom webpages, etc)	7
Connecting from Windows Computers Using Putty	9
1. Putty for terminal and command line software only	10
2. Putty for command line software and X-Windows software with MobaXterm.	11
3. Using putty for VNC connection (Linux remote desktop).....	12
4. Using Putty for other network services restricted to Cornell campus.	13
Connecting from Unix Computers (Linux, Mac).....	16
1. Using terminal and command line software only.....	16
2. Using command line software and X-Windows software.	17
3. Using network services restricted to Cornell campus.	17
Optional Tip: set up passwordless login between BioHPC machines	18

Who should read this

This document is for users wishing to connect to BioHPC from **outside** the Cornell campus firewall. If your computer is connected to the Cornell network, you should be behind the firewall already, and thus able to connect directly to all BioHPC machines. If any of the following are true, you should NOT need this document:

- You are physically on the Cornell campus and connected to campus network – this includes Ithaca and Weill Cornell Medicine (NYC or Qatar)
- You have a Cornell (Ithaca) NetID and are connected to Cornell VPN (more info here: <https://it.cornell.edu/cuvpn>)
- If you are connected to Weill Cornell VPN, it is a gray area (depending on the IP address assigned to you by the VPN software). You can try to connect to BioHPC directly. If it does not work, you will need to use instructions in this document. An easy check is to try to log onto the BioHPC webpage (https://biohpc.cornell.edu/login_bio.aspx). If you can log on **without** a 2-factor code, then you should have direct access to all BioHPC resources.

If any of the above are true, and you still need help connecting, you can refer to our documentation for on-campus users: https://biohpc.cornell.edu/lab/doc/Remote_access.pdf.

Prerequisite: Set up 2-factor authentication with BioHPC

BioHPC requires two-factor authentication (2FA) for logins outside the Cornell network; both to login

machines and the BioHPC webpage (as well as some enhanced security compute machines). The 2FA protocol is different from what is used by other Cornell services. Rather than a push notification, you will need to supply a time-based one-time password (TOTP) as well as your BioHPC username/password. The TOTP is provided by an app installed on your smartphone and linked to your BioHPC account (the app can also be installed on laptops/desktops/browsers if you prefer). To enroll in two-factor authentication, and for more details, visit http://biohpc.cornell.edu/lab/2fa_setup.aspx. This only needs to be done once, though you can re-visit this site to link additional devices to your BioHPC account.

When connecting to BioHPC login machines or the BioHPC webpage, you will be prompted for a TOTP after providing your BioHPC password, but this is only required once per week per IP address. There are a few applications that do not support 2FA – such as when using ‘QuickConnect’ in FileZilla. In this case, we recommend first logging into the BioHPC website (https://biohpc.cornell.edu/login_bio.aspx). You will supply your 2-factor code there, and then will be able to use other apps without needing a TOTP from that IP address for the next week.

Overview: Two general options for connecting from off campus

We have two options for off-campus users:

1. You can create a ‘custom firewall’ through the BioHPC webpage (<https://biohpc.cornell.edu/lab/iptables.aspx>). At this URL, you can enter the IP address of your laptop/desktop, and the name of the BioHPC server you would like to connect to (you need to have access to this server first, either through an active reservation or by belonging to a hosted machine access group). This will open the BioHPC server to your IP address for a limited duration of time (up to 3 months; this can be renewed on the webpage indefinitely). The custom firewall is perhaps the easiest option, but has some drawbacks:
 - It is limited to SSH / SCP / SFTP connections (port 22) only. If you need to connect to a different port (for example, to connect to Rstudio, jupyter, or VNC session), then you will need an SSH tunnel- you can either tunnel through a login server, or, with the custom firewall, you can tunnel through port 22 on the compute server. Creation of SSH tunnels is described in this document.
 - Depending on your local internet connection, your IP address may change frequently, in which case it may be easier (and somewhat more secure) to set up tunnels through the login server.

If you choose this ‘custom firewall’ option, and only need SSH connections, then you should not need the rest of this document. Just go to the webpage, create the custom firewall to your desired machine, and connect as if you are an on-campus user.

2. You can connect to one of the 3 BioHPC login servers, which are open to the world. From the login servers, you can connect to any of the BioHPC compute servers. This can be done manually: simply SSH to a login server, and from the login server terminal, you can SSH to the compute machine. Or, you can configure an ‘SSH Tunnel’ that will automatically direct network traffic between your local machine and the compute server, through a login machine. Both approaches are described in detail in this document.

About login servers

Login servers are the only BioHPC machines open to the world. The three login servers are:

`cbsulogin.biohpc.cornell.edu`
`cbsulogin2.biohpc.cornell.edu`
`cbsulogin3.biohpc.cornell.edu`

Throughout this document, we use the first machine, cbsulogin, as an example, but any of the three login machines can be used interchangeably.

For security purposes, the login servers may temporarily ban your IP address after several unsuccessful login attempts. This ban generally lasts one hour. If this happens, you can try another login server. If you continue to have trouble logging in, you can contact BioHPC support at support@biohpc.cornell.edu.

Connecting from Windows Computers using MobaXterm

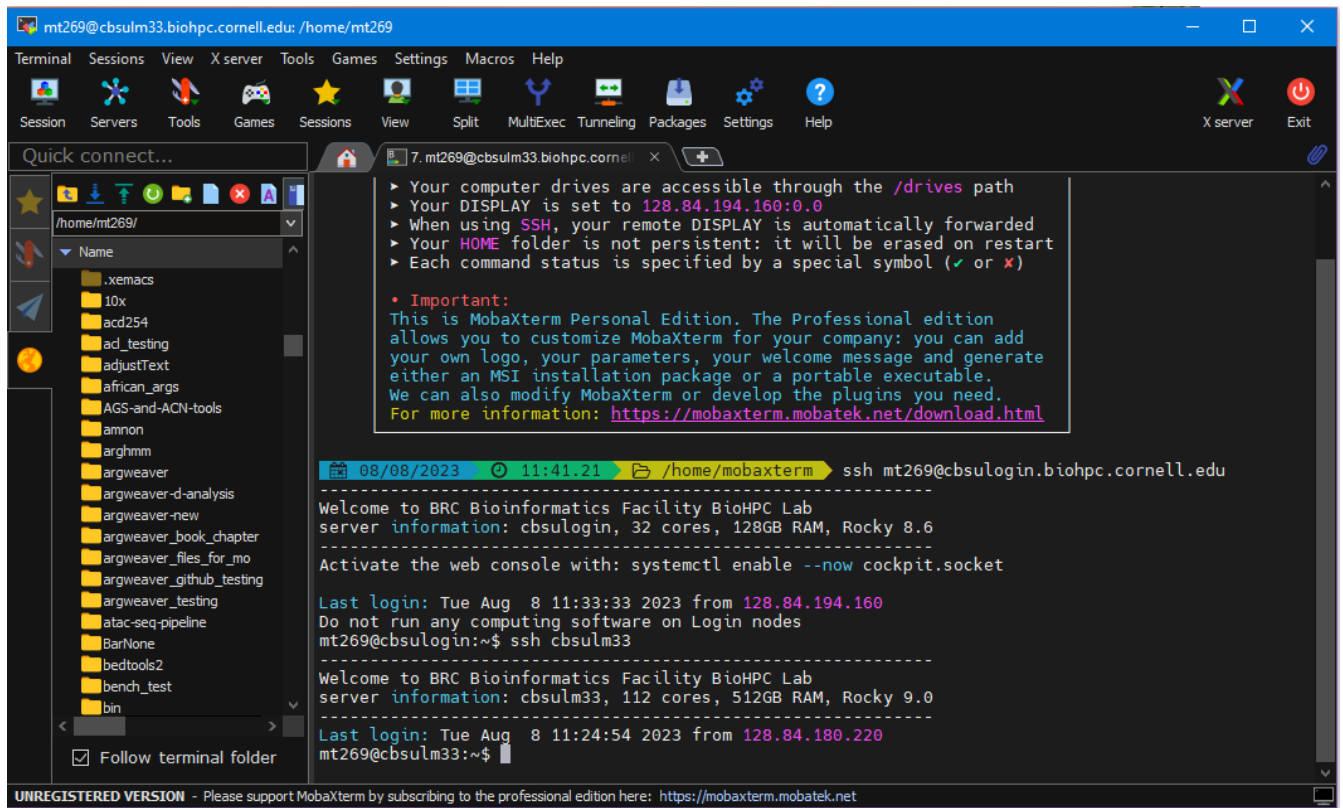
MobaXterm is a popular SSH/SCP/SFTP client with X11 support. It is intuitive to use, but is a commercial product (with free version), and therefore NOT open source. If you prefer an open source solution, see the section in this document: “Connecting from Windows Computers Using Putty”. Note that if you want graphical (X11) connection, you will still need MobaXterm or another X11 software installed.

You can install the Free Home Edition of MobaXterm from their webpage:

<https://mobaxterm.mobatek.net/>. You can choose either the Portable edition or the Installer edition - either is fine. The installer edition is installed like a normal program (and may require admin access to install), whereas the Portable edition is just a binary you save to your Desktop and run as needed. (It is downloaded as a zip package, make sure to extract the zip file first.) Both installer/portable versions will save your connections and settings for future use, though the free edition only supports a limited number of saved sessions.



1. Using MobaXterm for basic SSH (quick start)

Start MobaXterm, and click “Start local terminal”. From here, you can type an ssh command, such as `ssh username@cbsulogin.biohpc.cornell.edu` (replacing username with your BioHPC user ID). You may be prompted for both a password and a 2-factor code; if desired, you can tell MobaXterm to save your password, and you shouldn’t need it again (until your BioHPC password changes).



Once you have logged into a login machine, you can execute another ssh command, such as `ssh cbsulm33`, to connect to a compute machine where you have a reservation (note, you do not need username or domain this time, since they are the same as on the login machines). With default settings, graphical applications should work, and will open windows on your local machine as needed.

2. Transferring files with MobaXterm

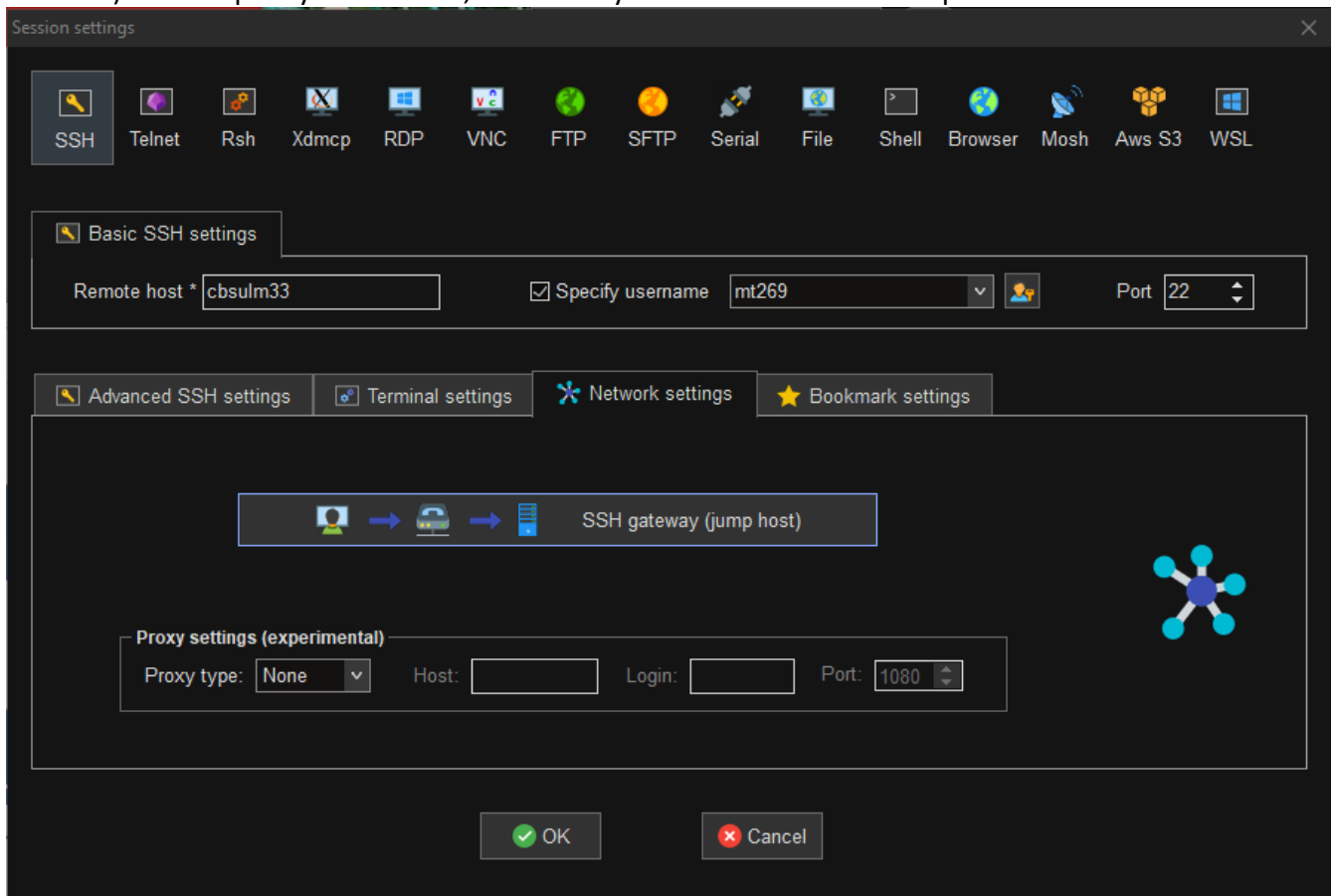
Once logged in to BioHPC, notice the left-hand panel with several different tools. You can see the names of the tools if you hover over the icons. The default tool with the yellow/orange circle icon is SSH browser (SFTP), and shows files/folders on the remote server (in this case, cbsulogin). You can use this tool to upload files from your local computer to the remote server, and vice versa. When using this tool, there are several options along the top – including a down arrow – for downloading files from the BioHPC to your local machine, and an up arrow for uploading files to from your machine. You can select a file/folder on the remote machine, then click the down arrow  to download your selection to your local machine. Or, you can choose a folder, click the up arrow , to upload files into that folder. If you click the “Follow terminal folder” button, then as you change directories on the remote machine (using the `cd` command), the files displayed on the left panel will be those in your working directory.

Important: if you use the login method above (ssh'ing first to cbsulogin, then to a compute machine), the file transfer tool will be connected to cbsulogin, **not** the compute machine. The

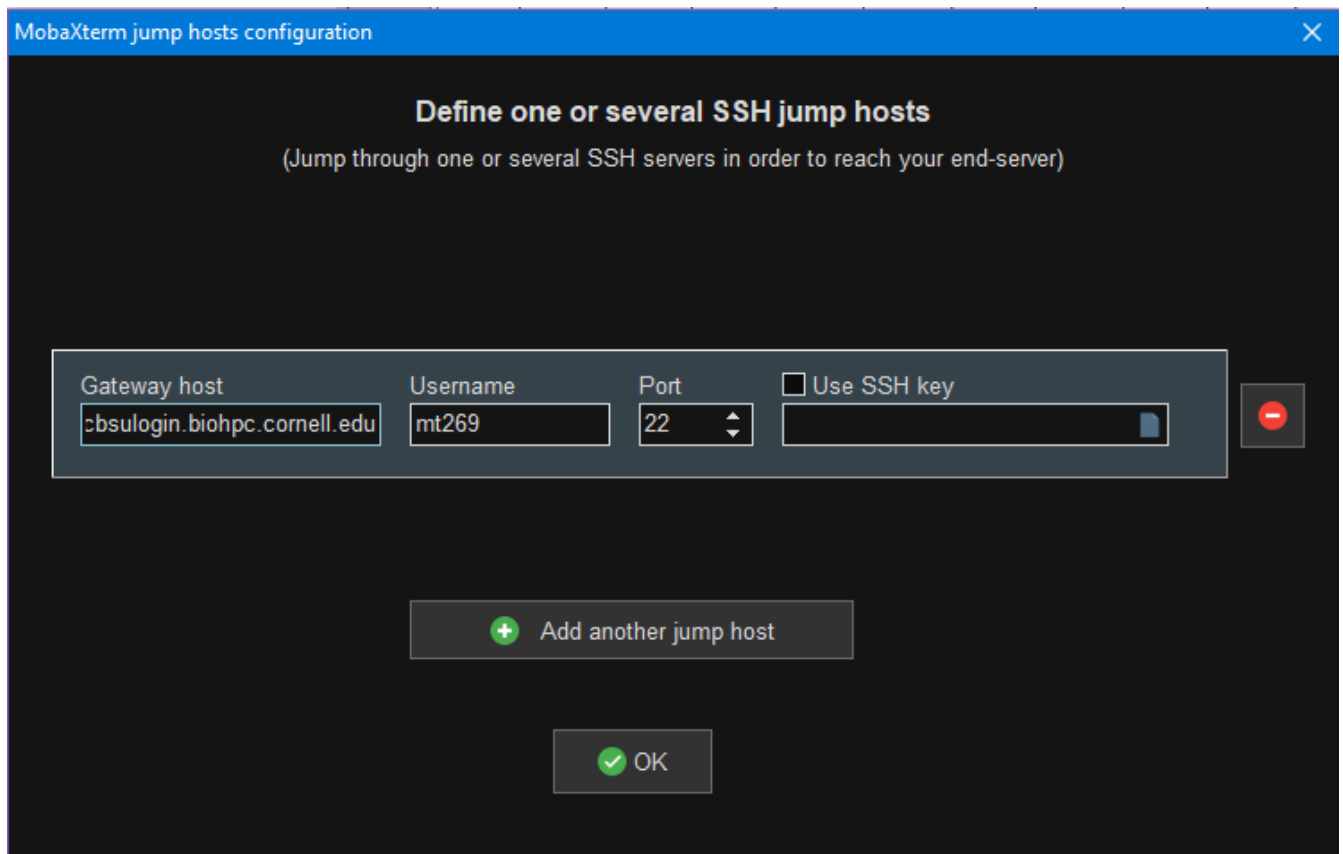
'follow terminal folder' button will only work for cd commands executed on cbsulogin, not the compute machine, and you will not be able to transfer files to /workdir on the compute machine this way. In the next section, we show how to log into the compute server more directly, using cbsulogin as a 'Jump host'. With the jump host approach, the file transfer tool will transfer files between your local machine and the compute server.

3. Using MobaXterm to SSH to remote server (using login server as 'jump host')

Open MobaXterm and click on the 'Session' icon in the top graphical menu. A new window will pop up. Click on the 'SSH' icon at top left. Under Remote host, put the name of the compute machine you want to connect to. In this example, I will use cbsulm33. (Note: the full name is not necessary here, since you will be connecting through a login machine, which is on the same domain.) Check 'specify user name', and enter your BioHPC user ID. The port should be 22.



Now, click on 'Network settings', and the 'SSH gateway (jump host) button'. This will open a new window:



Enter the full address of a login server as the Gateway host, as well as your username, and Port 22. Then click OK. Click OK again on the previous window, and it should initiate the connection. If it is your first time logging in, you may need to 'Accept' the identity of new remote hosts. You will be prompted to enter your BioHPC password, perhaps twice (once for login and once for compute machine), as well as (possibly) a 2-factor code. If you tell MobaXterm to save the passwords, you will not need to enter them in the future (until your password gets reset).

If successful, you should be logged into the remote host now. If you close the MobaXterm window, it should save this session – the next time you open MobaXterm, you should see the machine name listed under the "User sessions" menu on the main screen, and you can re-connect to the same machine by double-clicking it.

4. Using MobaXterm for VNC

About VNC: VNC is a persistent desktop-like environment for linux. If you start a VNC session, you can connect to it, start jobs running, close the VNC window, and when you re-connect, your desktop will be just like you left it, and your jobs still running. To start a VNC session, you first need to navigate to 'My Reservations' site (<https://biohpc.cornell.edu/lab/labresman.aspx>) on the BioHPC webpage. Start your VNC connection by clicking on "Connect_VNC" on the 'Action' column of your reservations list. You may first want to choose a screen resolution for your VNC session, this is in a drop-down menu under your list of reservations. After clicking 'Connect VNC', a port number will be displayed in the right-most column of the Reservations table. Note the port number, you will need this in the next step.

You can use MobaXterm VNC viewer; on the main MobaXterm window, click VNC. Here, you can set up a jump host just as in the previous section; enter your compute server name at top, and the VNC port number displayed on the webpage. Click 'Network settings', and then 'SSH gateway (jump host)', and use a login server (cbsulogin.biohpc.cornell.edu) and port 22 as your jump host.

Once the VNC session is connected, you can (if you wish) drag the tab outside of mobaxterm and resize it. If you open a terminal in the VNC session, the command 'xrandr' can be used to resize the desktop to your desired resolution. With no arguments, xrandr will display a list of supported resolutions. To set the resolution, use the -s option, for example: xrandr -s 1920x1080.

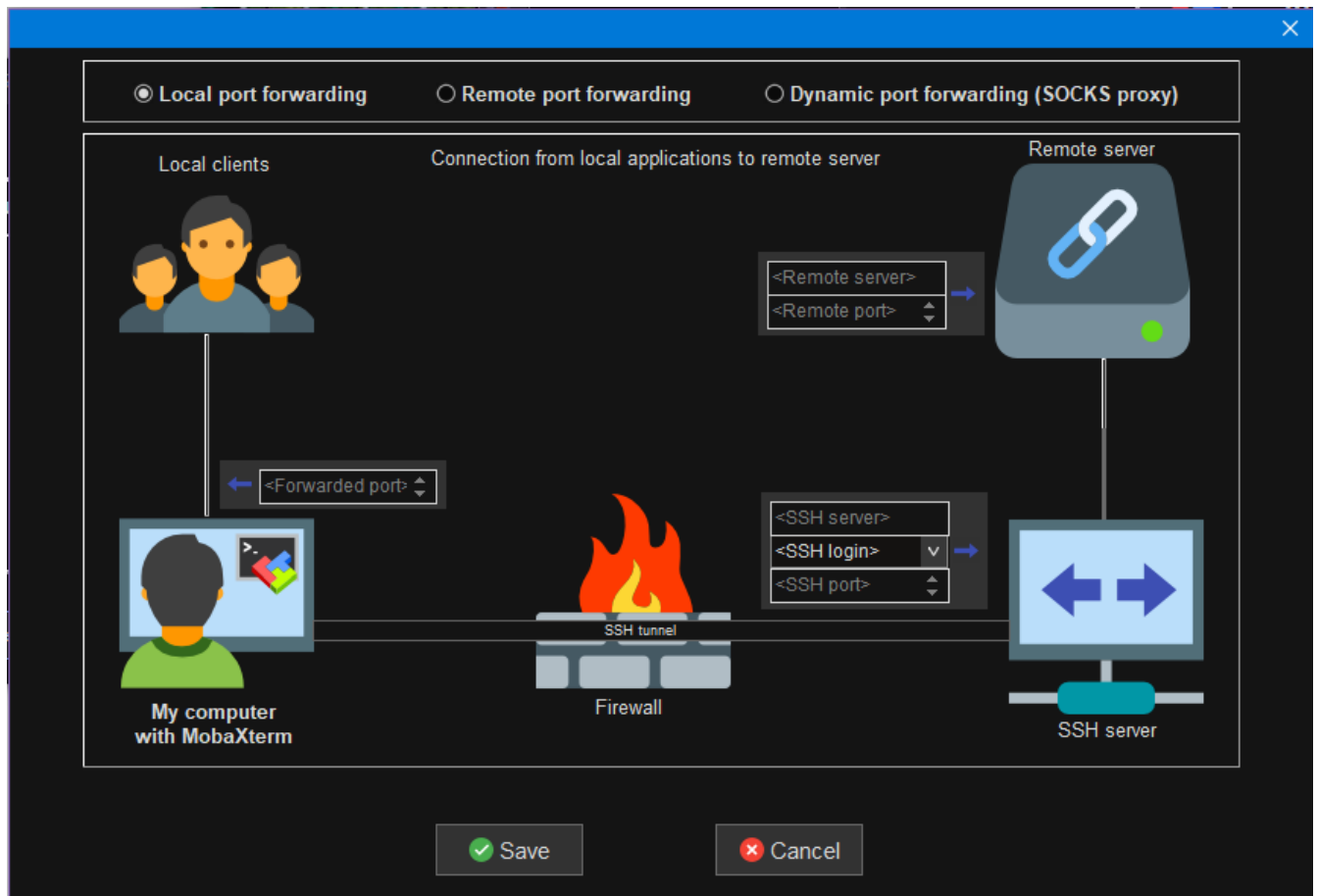
Note: You can use other VNC clients instead of MobaXterm, such as RealVNC. To use a different client, follow the instructions in the next section, to tunnel your VNC port through a login machine. After starting the tunnel, you can connect your VNC client to localhost:port (i.e. "localhost:5901").

5. Using MobaXterm for services connected to other ports (VNC, Rstudio, jupyter, custom webpages, etc)

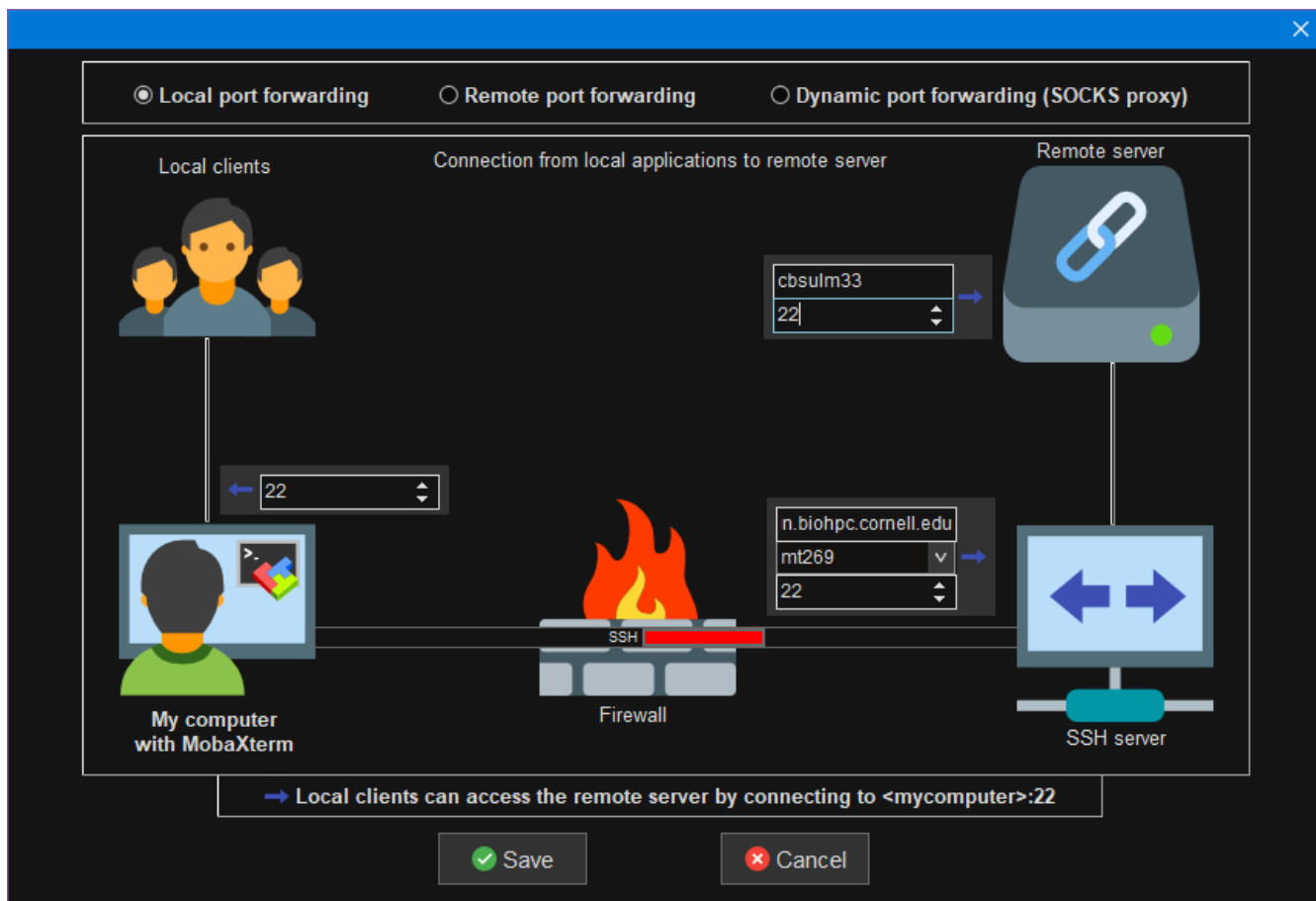
Services like Rstudio and jupyter require using a web browser to connect to a specific port on your compute machine. For more information about running Rstudio and jupyter, see specific instructions in the Software section of the BioHPC webpage (<https://biohpc.cornell.edu/lab/userguide.aspx?a=software>). Rstudio usually runs on port 8015 (unless it is run through docker), jupyter (when not run through docker) is often port 8016 (though you can specify the port when you start jupyter). Whichever service you are running, you will first need to start it, and note the port number. At BioHPC, the port will generally be between 8009-8039.



The idea is to make an 'SSH tunnel' that will direct internet traffic between this port on the remote machine, and another port on your own machine, through the SSH port on a login machine. Once the tunnel is running, you can direct your browser to "localhost:8015" instead of "cbsulm33:8015" (for example). The port number used on the local side does not need to be the same as the port on the remote machine, but for simplicity, we will keep it the same here.

To make the tunnel: Open MobaXterm and click on the 'Tunneling' icon in the top graphical menu. This same tool can also be found under the menu 'Tools → MobaSshTunnel (port forwarding)'. A new window will pop up, click on 'New SSH Tunnel'. This will open a new window:



At the top, keep the default option, 'Local port forwarding'. The 'Forwarded port' on the left side is the port on your local machine that you will ultimately connect to (i.e., the 8015 in localhost:8015). On the top right side, the 'Remote server' is the compute machine, and the 'Remote port' is the port number that your service is running on (this could be the VNC port, rstudio/jupyter port, etc). The bottom right corner is the information for the login machine, here you want to use port 22, and enter your BioHPC user ID for 'SSH login'. An example (mapping localhost:8015 to cbsulm33:8015 through cbsulogin.biohpc.cornell:22 is shown below – though note the cbsulogin.biohpc.cornell.edu address is not fully shown).



Now, click 'Save', and in the previous window you can enter a name for your tunnel, then click the 'Start' arrow  to start the tunnel (alternatively, if this is your only tunnel, you can click 'Start all tunnels'). You may have to Accept the connection to a new server, and enter your BioHPC password (which you can opt to save), and/or 2FA code. Once the tunnel is successfully running, you should see the 'Start' button is inactivated. At this point, you should be able to navigate to localhost:8015 on your web browser (or VNC client, if you are porting a VNC tunnel) to access the service. If your internet connection is lost, you may find the tunnel has stopped, and needs to be restarted. After setting the tunnel up the first time, you can simply go to the 'Tunneling' section of MobaXterm, and click the 'Start' button, or edit the tunnel by clicking on the Gears icon  under 'Settings'.

Connecting from Windows Computers Using Putty

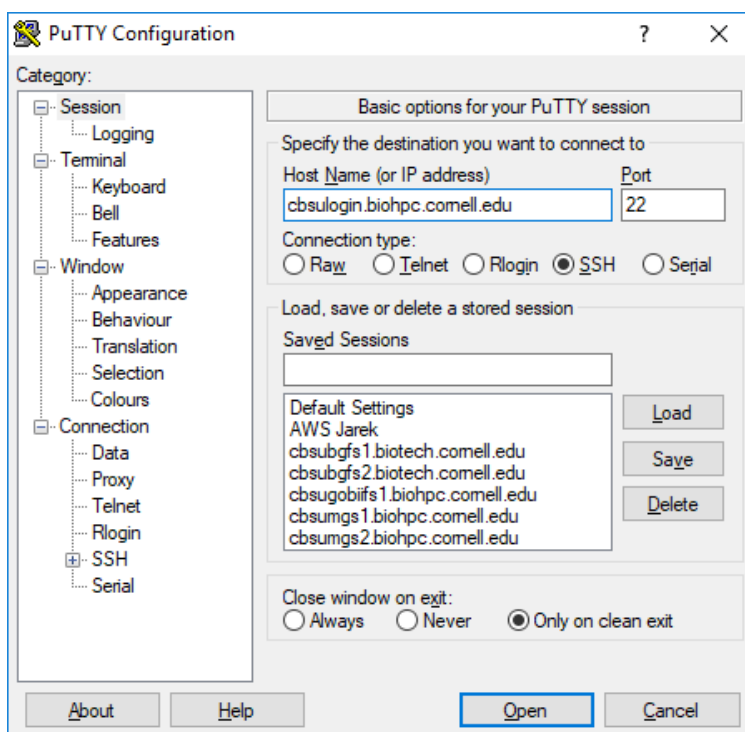
First, install software needed to connect to BioHPC Lab as described in our online documentation (Putty for connecting to workstations, FileZilla or WinSCP for file transfer, MobaXterm for X-Windows). You may want to install RealVNC VNC Viewer (Free edition) if you prefer to have access to the fully functional Linux desktop environment.

Reserve a workstation you want to use via our webpage (<http://biohpc.cornell.edu/lab/labres.aspx>). In this section we will assume your workstation is cbsum1c2b001, please substitute the name of your reserved workstation when setting up the connection.

There are several scenarios, depending on the software you will use.

1. *Putty for terminal and command line software only*

Set up your Putty to connect to the login node, cbsulogin.biohpc.cornell.edu



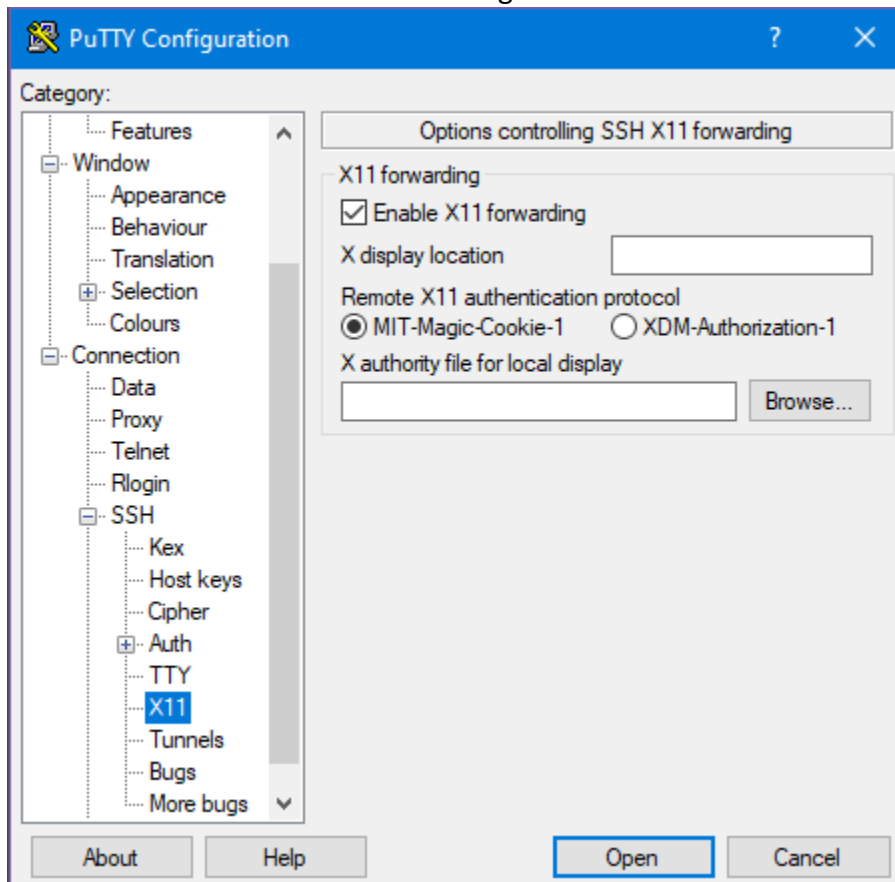
You will be asked for user id and password, and possibly a one-time password from your two-factor authentication app (unless you have already provided one from the same IP that day). Once your terminal session on the login node is established, type the command

```
ssh cbsum1c2b001
```

which will connect you to your reserved workstation (you will be asked for your BioHPC password again). Remember to substitute the example workstation name with your reserved workstation name! Now you will be able to work on your workstation with command line software. Once you are done, please disconnect from both the reserved workstation and cbsulogin (press “Ctrl-D” twice).

2. *Putty for command line software and X-Windows software with MobaXterm.*

Set up your Putty to connect to `cbsulogin.biohpc.cornell.edu` as above, but before connecting scroll down the left panel, expand :”Connection”, “SSH”. Click on “X11” and make sure “Enable X11 forwarding” is checked:



Click “Open”. Once connected to the login node (`cbsulogin.biohpc.cornell.edu`), type the command

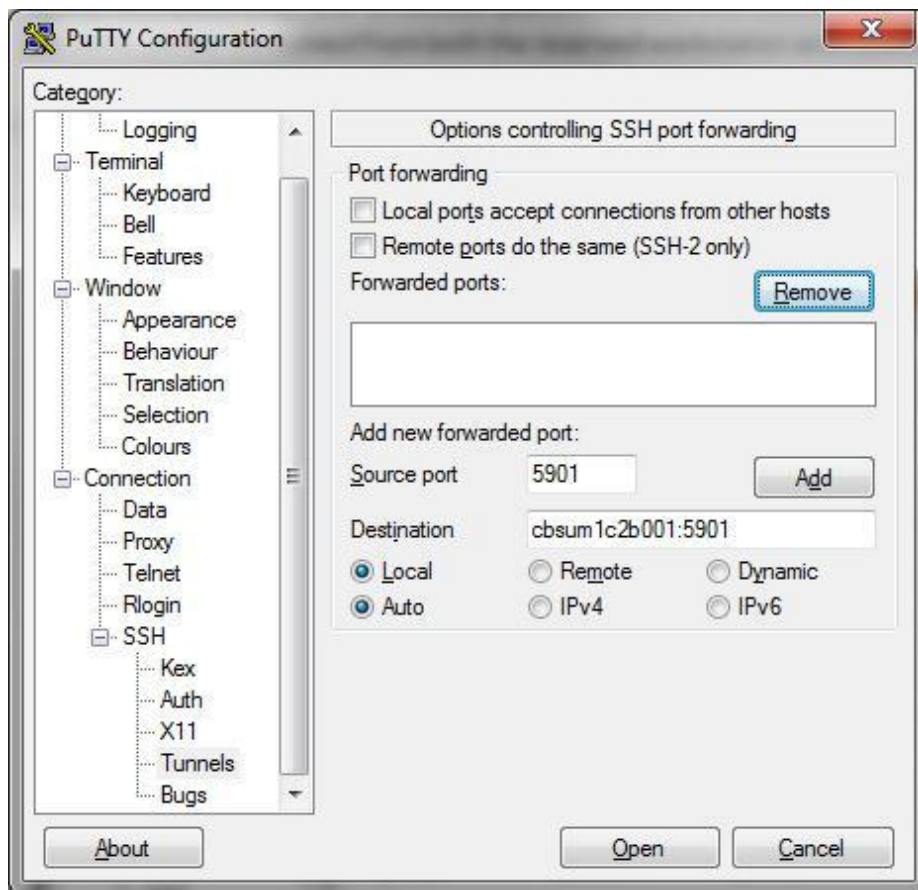
```
ssh -X cbsum1c2b001
```

which will connect you to your reserved workstation (you will be asked password again). Remember to substitute example workstation name with your reserved workstation name! Now you will be able to work on your workstation with command line software and X-Windows software, just start MobaXterm on your local computer and then GUI software on BioHPC Lab machine. Once you are done, please disconnect from both the reserved workstation and `cbsulogin` (press “Ctrl-D” twice).

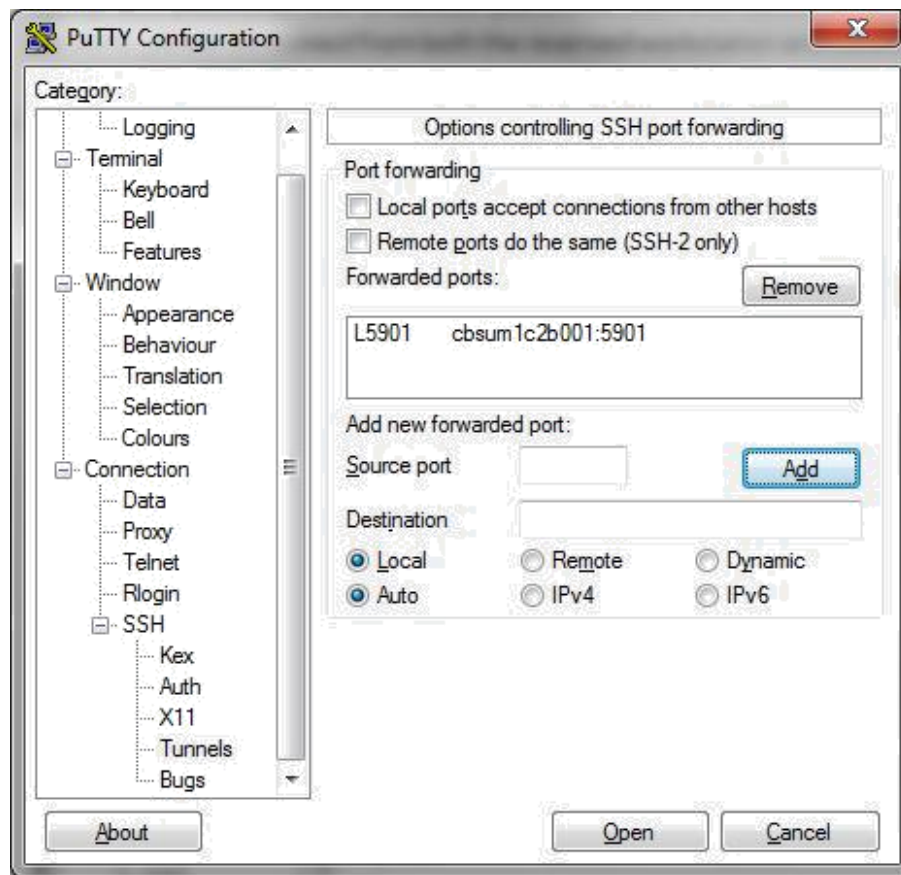
3. Using putty for VNC connection (Linux remote desktop).

Navigate to the 'My Reservations' site (<https://biohpc.cornell.edu/lab/labresman.aspx>) on the BioHPC webpage. Start your VNC connection by clicking on "Connect_VNC" on the 'Action' column of your reservations list. You may first want to choose a screen resolution for your VNC session, this is in a drop-down menu under your list of reservations. After clicking 'Connect VNC', a port number will be displayed in the right-most column of the Reservations table. Note the port number, you will need this in the next step.

Set up your Putty to connect to cbsulogin.biohpc.cornell.edu as above, but before connecting scroll down the left panel, expand : "Connection", "SSH". Click on "Tunnels", and fill tunneling information as below (remember to substitute workstation name, and use the port number from the webpage instad of 5901):



Click "Add" and Putty will display forwarding information as below:



Click “Open” to connect to the login node (cbsulogin.biohpc.cornell.edu). Once connected, type the command

```
ssh cbsum1c2b001
```

which will connect you to your reserved workstation (you will be asked password again). Remember to substitute example workstation name with your reserved workstation name! Now you will be able to work on your workstation with command line software and X-Windows software. Once you are done, please disconnect from both the reserved workstation and cbsulogin (press “Ctrl-D” twice).

In the RealVNC Viewer type “localhost:5901” as the address and connect. Replace 5901 with the port number displayed on the rightmost column in the “My Connections” page for your reservation.

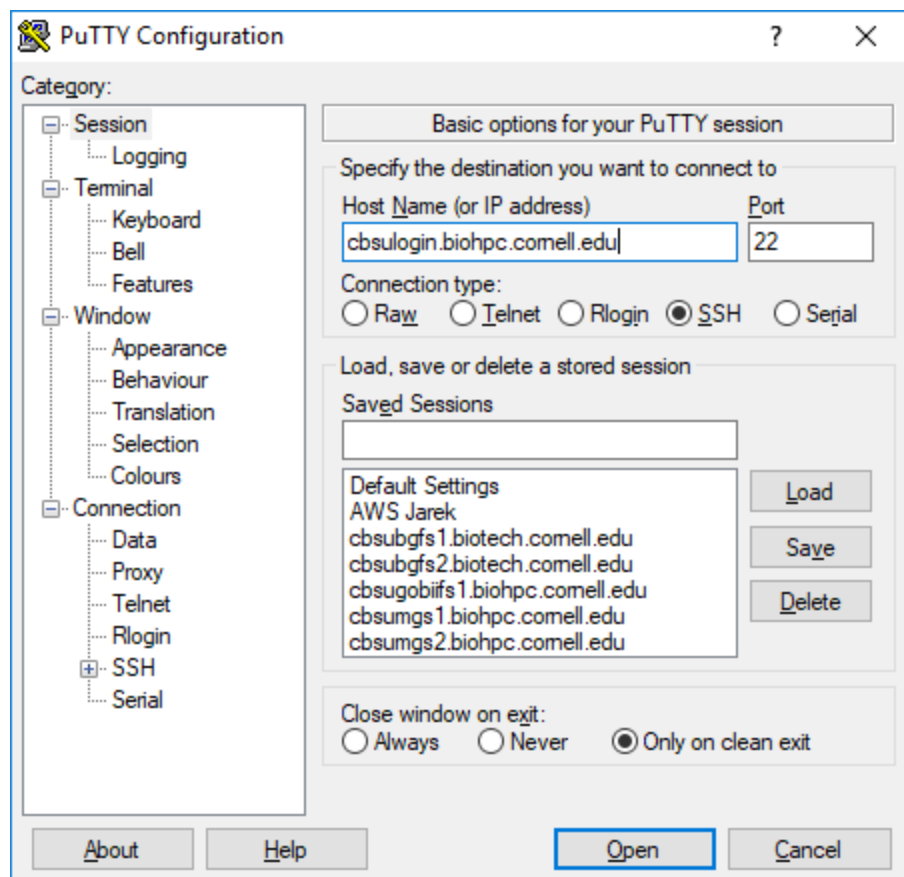
4. *Using Putty for other network services restricted to Cornell campus.*

There are a lot of services that are restricted to Cornell campus network for security reasons, for example R-Studio servers, custom websites run on reserved or hosted

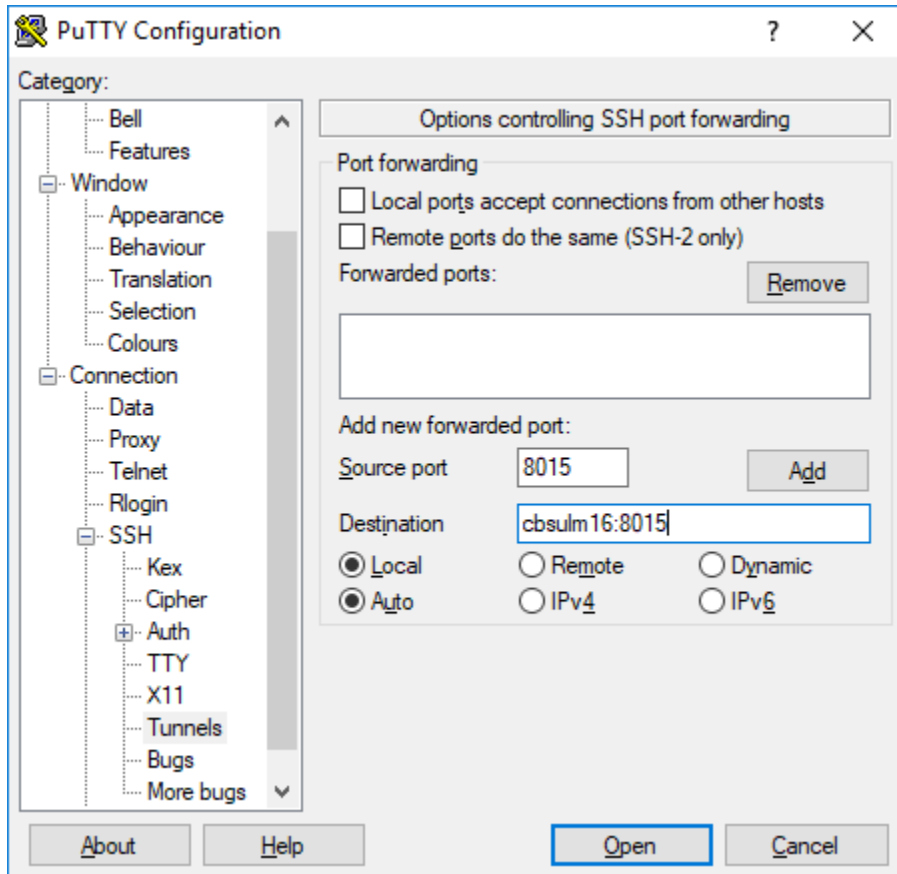
workstations. You can access them with ssh tunneling from your external computer. For example, R-Studio server on BioHPC computers (if started) uses network port 8015. Other services use different ports – in the examples below simply substitute port 8015 with the port of your service.

Set up your Putty to connect to one of the login nodes, cbsulogin.biohpc.cornell.edu (or cbsulogin2, cbsulogin3).

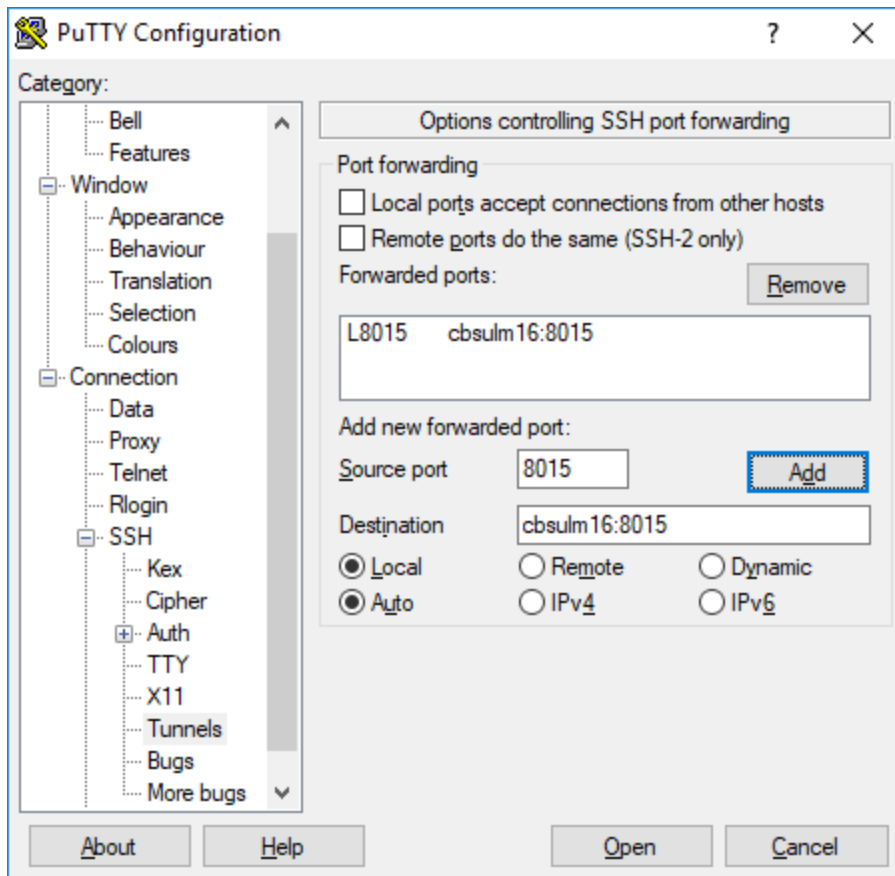
Set up your Putty to connect to one of the login nodes, cbsulogin.biohpc.cornell.edu (or cbsulogin2, cbsulogin3).



Before connecting scroll down the left panel, expand: "Connection", "SSH". Click on "Tunnels", and fill tunneling information as below (remember to substitute server name with your server!):



Click “Add”, and the information will be added to the tunneling.



Now you can click “Open” and proceed as usual with PuTTY ssh connection. Once connected you can use local port 8015 as a gateway to your R-Studio server on cbsulm16 port 8015, to use it just enter <http://localhost:8015/> in your local web browser. NOTE: local port number may be different than the remote port number, the remote port number must correspond to the network service port number, but the local one is arbitrary.

Connecting from Unix Computers (Linux, Mac).

Reserve the workstation you want to use via our web pages (<http://biohpc.cornell.edu/lab/labres.aspx>), just as any user. For this document let’s assume your workstation is cbsum1c2b001, please substitute your workstation name when setting up the connection.

1. Using terminal and command line software only

Connect to cbsulogin.biohpc.cornell.edu using ssh (replace “mylabid” with your BioHPC user id):


```
ssh mylabid@cbsulogin.biohpc.cornell.edu
```

Once connected, type second ssh command:

```
ssh cbsum1c2b001
```

Remember to substitute example workstation name with your reserved workstation name! You are now ready to work with command line programs. Once done, please disconnect both connections.

2. *Using command line software and X-Windows software.*

Note: if logging in from a Mac, you will need to install XQuartz on your machine (<https://www.xquartz.org/>).

Connect to cbsulogin.biohpc.cornell.edu using ssh:

```
ssh -X -t -t -t mylabid@cbsulogin.biohpc.cornell.edu "ssh  
mylabid@cbsum1c2b001 -X"
```

You will be asked for password twice, as well as a one-time password from your 2FA app. Remember to substitute example workstation name with your reserved workstation name and “mylabid” with your Lab ID! You are now ready to work with command line and X-Windows programs. Once done, please disconnect both connections.

3. *Using network services restricted to Cornell campus.*

There are a lot of services that are restricted to Cornell campus network for security reasons, for example R-Studio servers, custom websites run on reserved or hosted workstations and VNC. You can access them with ssh tunneling from your external computer. For example, R-Studio server on BioHPC computers (if started) uses network port 8015. Other services use different ports – in the examples below simply substitute port 8015 with the port of your service.

To tunnel to R-Studio server running on cbsulm16.biohpc.cornell.edu on port 8015 you need to open a terminal window and type the following command

```
ssh -N -L 8015:cbsulm16:8015  
biohpcid@cbsulogin.biohpc.cornell.edu
```

Of course you need to substitute cbsulm16 with your server name, biohpcid with your BioHPC user id and you may replace cbsulogin with cbsulogin2 or cbsulogin3. After connection is established you can access R-Studio server by typing url <http://localhost:8015/> in your local computer web browser. You can use different port number for local port, for example you can use 8080 if you want, and the command will be then

```
ssh -N -L 8080:cbsulm16:8015
biohpcid@cbsulogin.biohpc.cornell.edu
```

The local url will be then <http://localhost:8080/>

Optional Tip: set up passwordless login between BioHPC machines

When logged into any BioHPC machine, you can issue the following commands to set up passwordless login between BioHPC machines. This can be especially useful for off-campus users who need to tunnel through a login server, to avoid multiple password prompts.

```
cd      # this takes you to your home directory
ssh-keygen -t rsa # press enter a few times to skip over questions
cat .ssh/id_rsa.pub >> .ssh/authorized_keys
echo Host * >> .ssh/config
echo StrictHostKeyChecking no >> .ssh/config
```

```
chmod 700 .ssh
chmod 600 .ssh/authorized_keys .ssh/config
```

You can also append the contents of `.ssh/id_rsa.pub` to `$HOME/.ssh/authorized_keys` on other linux/Mac machines (and vice-versa) to authenticate logins between the cluster and your other workstations.