Installing Collidoscope on a Windows PC (updated 1 August 2019)

1. Download the zip file for Collidoscope at <https://github.com/prellgroup/Collidoscope>. (The current version is called “coll\_8\_1\_19.tar.gz”.) Put the whole zip file on your Desktop, right click it, and choose “unzip here”. You should now have a folder on your Desktop called “Collidoscope”, possibly with another folder inside it called “Collidoscope”, depending on how your unzipping program works. Inside this folder is another folder called “src”. That is where all the source files are for Collidoscope. There are also folders containing input files, coordinate files, and a few other things that Collidoscope will need to run, but you don’t need to do anything with these just yet.
2. Before you can do anything with Collidoscope, you will need a C++ compiler. Go to <https://www.visualstudio.com/downloads/> and download and install Visual Studio 2019 (choose the Community version). This is the lengthiest part of the process, and it will take a while, maybe 20 minutes. When the installer prompts you to choose, you really only need the “Universal Windows Platform development”, “.NET desktop development”, and “Desktop development with C++” options; there is no need to download or install the “Web & Cloud” options. Note that, if your Windows version hasn’t been updated in a while, it may require that you update Windows first.
3. Once Visual Studio is installed, go to <https://www.microsoft.com/en-us/download/details.aspx?id=57467> to install MSMPI (this is Microsoft’s support for Message Passing Interface, which can be used for parallelization in Collidoscope.) Make sure to download and install BOTH “msmpisdk.msi” and “msmpisetup.exe”.
4. Open up the project “src/Collidoscope.vcxproj” from the Collidoscope folder using Visual Studio.
5. Visual Studio has a bunch of drop-down menus at the top. Click “Project”, choose “Properties”, then select “Configuration Properties 🡪 C/C++ 🡪 Language”, and change the “OpenMP Support” option to “Yes”. This will tell Visual Studio that you want to enable parallelization in Collidoscope.
6. Now choose “Build” from the drop-down menus at the top of Visual Studio, and select “Build Solution”. Visual Studio will now compile Collidoscope and put out a whole bunch of messages describing its progress, pausing a few times. Just sit back and wait. Note that it will probably say something about being “finished” and “failing” at the end, but that is just Visual Studio saying it didn’t know how to debug Collidoscope, which you can ignore. You should now have an executable program called “Collidoscope.exe” in your Collidoscope folder.
7. Test that the program has compiled correctly by opening a Command Prompt (in the “Search Windows” dialog box in your task bar, type “Command Prompt” and open one). Navigate in the Command Prompt to the folder containing Collidoscope (usually, this will require typing “cd Desktop\Collidoscope\Collidoscope” or “cd Desktop\Collidoscope”, whichever actually contains Collidoscope.exe, and then pressing Enter). Once you are in this directory, type “Collidoscope.exe”, which should automatically calculate the CCS of ondansetron (about 110 square angstroms). If that worked, you are ready to go!
8. For further information about running Collidoscope on your own structures, see the “README.txt” file in your Collidoscope folder. Everything should work exactly as on Linux systems from here on out, except that, in Windows, the command to use Collidoscope is “Collidoscope.exe” instead of “./coll”.