Getting consensus peaks with python:

**Step 1: Import numpy, pandas and the peakoverlap module.**

To quickly import the peakoverlap module, just include the script in the same directory.

|  |
| --- |
| **import** numpy **as** np  **import** pandas **as** pd  **import** peakoverlap **as** po |

**Step 2: Make a list of peak files.**

|  |
| --- |
| peakfiles = [“peakfile1.txt”, “peakfile2.txt”, “peakfile3.txt”] |

**Step 3: Read the peak data into a tuple:**

|  |
| --- |
| peakdata = []  **for** curfile **in** peakfiles:  curdata = pd.read\_csv(curfile,sep="\t").values  peakdata.append(curdata[:,:3])  peakdata = tuple(peakdata) |

**Step 4: Generate a union of all peaks:**

|  |
| --- |
| peakunion = po.getUnionPeaks(peakdata) |

**Step 5: (Optional) Identifying union peaks based on the number of peak sets overlapping**

|  |
| --- |
| overlap\_counts, overlap\_mat = po.getOverlapCount(peakunion, peakdata)) |

The method getOverlapCount takes as input a peak dataset and a tuple of multiple peak datasets to produce two outputs:

* The number of peaks datasets in the tuple of peaks (peakdata) that overlap a peak in the peak set (peakunion). If there are n peaks in peak union, the dimension of the counts is (n)
* A matrix indicating which of the peak datasets overlap the peak in the peak union. If there are n peaks in peakunion, and m datasets in peakdata the dimension of this matrix is (n,m)