```
import sys
import iotbx.mtz

def isisis(array):
        batch=filter(lambda a:"BATCH" in a.info().labels,array)[0]

def run(ref_mtz):
    frame_arrays = iotbx.mtz.object(frame_mtz).as_miller_arrays(merge_equivalent
s=False)
    print type(frame_arrays)

    frame_I = filter(lambda a: "I" == a.info().labels[0], frame_arrays)[0]
    batch = filter(lambda a: "BATCH" in a.info().labels, frame_arrays)[0]

    print type(batch)
    print type(frame_I)

    print "Selected frame I:", frame_mtz, frame_I.info().label_string(), len(frame_I.
data())
    print "Selected BATCH :", frame_mtz, batch.info().label_string(), len(batch.da
ta())

    target_num=1
    sel= batch.data()==target_num
    newb=batch.select(sel)

    nnb,seleI=newb.common_sets(frame_I)

    print "frame_I origin: ",len(frame_I.data())
    print "New batch(sele: ",len(newb.data())
    print "NN  batch(comm: ",len(nnb.data())
    print "COMMON_SETS I : ",len(seleI.data())

        # confirmation of the function
    cnt=0
    for b in batch.data():
        if b==target_num:
            cnt+=1

        cntt=0
    for d in seleI.data():
        cntt+=1

    print "COUNTER: ",cntt

    print "Counter: 'batch' == 1:", cnt
    #print "type(seleI)=",type(seleI)
    #print seleI.info().label_string()
    #print nnb.info().label_string()
    print frame_I.info().label_string()
    print dir(seleI)

    cntcnt=0
    #for (hkl1, rI, rsigI),(bnum) in zip(seleI,nnb):
    for (hkl1, rI, rsigI),(bnum) in zip(seleI,nnb.data()):
        print hkl1, rI, rsigI,bnum
        #print hkl1, rI, rsigI
        cntcnt+=1

    print cntcnt

if __name__ == "__main__":
    frame_mtz = sys.argv[1]

    run(frame_mtz)
```