

```

#pragma rtGlobals=3      // Use modern global access method and strict wave access.

Function SplitPCAnn(root) // call example: Concat("Veh_RT*")
    String root //prefix of waves

    String xS = root + "_x"
    String yS = root + "_y"
    String zS = root + "_z"
    String xList=wavelist(xS,";", "")
    String yList=wavelist(yS,";", "")
    String zList=wavelist(zS,";", "")
    String xWave,yWave,zWave,wspName,wsnName,wList,newspName,newspName
    make /o/n=(1,3) centroid
    Variable d,n,i,j,k,l,dist,kts

    for (i = 0; i < ItemsInList(xList); i += 1)
        xWave = StringFromList(i, xList)
        yWave = StringFromList(i, yList)
        zWave = StringFromList(i, zList)
        Wave xw = $xWave
        Wave yw = $yWave
        Wave zw = $zWave
        wspName = ReplaceString("_x",xWave,"_sp")
        //split positive name
        wsnName = ReplaceString("_x",xWave,"_sn")
        //split negative name
        wList = xWave + ";" + yWave + ";" + zWave + ";"
        Concatenate /o wList, $wspName //creates new wave or overwrites existing
        Concatenate /o wList, $wsnName //creates new wave or overwrites existing
        Wave wsp = $wspName
        Wave wsn = $wsnName
        //now do the PCA
        pca/all/SEVC/SRMT/SCMT xw,yw,zw
        wavestats /q xw //form centroid
        centroid[0][0]=v_avg
        wavestats /q yw
        centroid[0][1]=v_avg
        wavestats /q zw
        centroid[0][2]=v_avg

        Wave M_C //declare M_C wave from PCA /SCMT flag
        // because 0=ax+by+cz+d where
        d=-((M_C[2][0]*centroid[0][0])+(M_C[2][1]*centroid[0][1])+(M_C[2]
[2]*centroid[0][2]))
        n=umpnts(xw)
        for (l = 0;l < n;l +=1) // use l for loop var
            dist=((M_C[2][0]*xw[l])+(M_C[2][1]*yw[l])+(M_C[2][2]*zw[l]))+d
        //find distance to plane
            if (dist>=0)
                wsn[l][0]=nan
                wsn[l][1]=nan
                wsn[l][2]=nan
            else
                wsp[l][0]=nan
                wsp[l][1]=nan
                wsp[l][2]=nan
            endif
        endif
    endfor
    matrixop/o c1=col(wsn,0)
    matrixop/o c2=col(wsn,1)
    matrixop/o c3=col(wsn,2)
    WaveTransform zapnans c1
    WaveTransform zapnans c2
    WaveTransform zapnans c3
    Concatenate /o {c1,c2,c3}, $wsnName
    // now use wsn for nn calc
    kts = dimsize(wsn,0)
    Make /O /N=(kts) TempWave
    Make /O /N=(kts) DistWave
    newspName = ReplaceString("_x",xWave,"_dist_sn")
    for (j = 0; j < kts; j+=1)
        for (k = 0; k < kts; k+=1)
            if (k==j)

```

```

        TempWave[k]=nan
    else
        TempWave[k]=sqrt(((wsn[j][0]-wsn[k][0])^2)+((wsn[j][1]-wsn[k]
[1])^2)+((wsn[j][2]-wsn[k][2])^2))
    endif
endfor
DistWave[j]=WaveMin(TempWave)
endfor
Duplicate /o Distwave, $newsnName
KillWaves TempWave, DistWave
matrixop/o c1=col(wsp,0) //zapnans - wave needs to be broken and then
zapped and reassembled
matrixop/o c2=col(wsp,1)
matrixop/o c3=col(wsp,2)
WaveTransform zapnans c1
WaveTransform zapnans c2
WaveTransform zapnans c3
Concatenate /o {c1,c2,c3}, $wspName
// now use wsp for nn calc
kts = dimsize(wsp,0)
Make /O /N=(kts) TempWave
Make /O /N=(kts) DistWave
newspName = ReplaceString("_x",xWave,"_dist_sp")
for (j = 0; j < kts; j+=1)
    for (k = 0; k < kts; k+=1)
        if (k==j)
            TempWave[k]=nan
        else
            TempWave[k]=sqrt(((wsp[j][0]-wsp[k][0])^2)+((wsp[j][1]-wsp[k]
[1])^2)+((wsp[j][2]-wsp[k][2])^2))
        endif
    endfor
    DistWave[j]=WaveMin(TempWave)
endfor
Duplicate /o Distwave, $newspName
KillWaves TempWave, DistWave
endfor
End

```