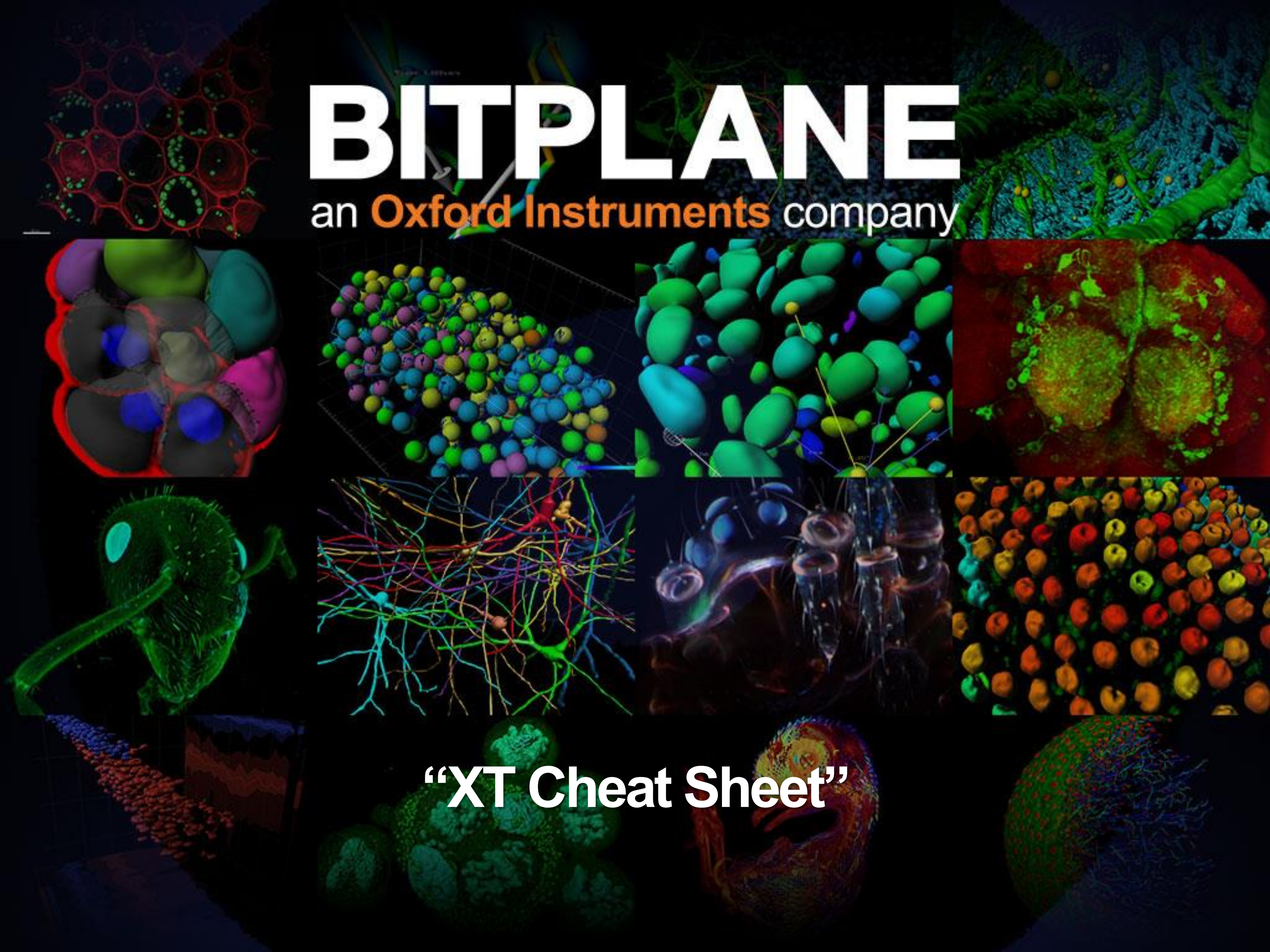


BITPLANE

an Oxford Instruments company

"XT Cheat Sheet"

Start XT Get ImarisID

```
% get the application object
if isa(aImarisApplicationID, 'Imaris.IApplicationPrxHelper')
  % called from workspace
  vImarisApplication = aImarisApplicationID;
else
  % connect to Imaris interface
  javaaddpath ImarisLib.jar
  vImarisLib = ImarisLib;
  if ischar(aImarisApplicationID)
    aImarisApplicationID = round(str2double(aImarisApplicationID));
  end
  vImarisApplication = vImarisLib.GetApplication(aImarisApplicationID);
end
```

To launch XTension for debugging in Matlab type “NAMEofXTension 0”

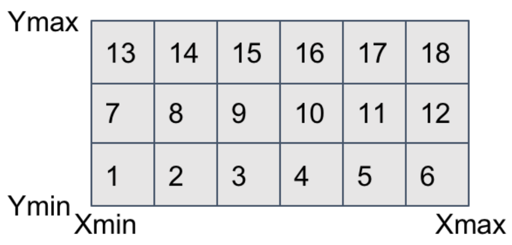
- ‘0’ represents the Imaris application ID (by default this starts at 0)
    - if debugging fails to recognize Imaris ID, the ImarisID is likely not zero
        - close Imaris, let rest for 10-15sec and restart to establish ID=0
- no spaces in .m file are allowed

Listing Spots Objects and Selecting One

```
vSpots = vImarisApplication.GetFactory.ToSpots(vImarisApplication.GetSurpassSelection);
%SelectSpotsObject
vNumberOfSpots = 0;
vSpotsList{vSurpassScene.GetNumberOfChildren} = [];
vNamesList{vSurpassScene.GetNumberOfChildren} = [];
for vChildIndex = 1:vSurpassScene.GetNumberOfChildren
    vDataItem = vSurpassScene.GetChild(vChildIndex - 1);
    if vImarisApplication.GetFactory.IsSpots(vDataItem)
        vNumberOfSpots = vNumberOfSpots+1;
        vSpotsList{vNumberOfSpots} = vImarisApplication.GetFactory.ToSpots(vDataItem);
        vNamesList{vNumberOfSpots} = char(vDataItem.GetName);
    end
end
%test if there are any Spots Objects
if vNumberOfSpots==0
    msgbox('Please create at a spots object!');
    return;
end
vNamesList = vNamesList(1:vNumberOfSpots);
 %input dialog to list all Spot objects and select one
if vNumberOfSpots > 1
    [vPair, vOk] = listdlg('ListString',vNamesList,'SelectionMode','single',...
        'ListSize',[250 150],'Name','Spots','InitialValue',1,...
        'PromptString',{'Please select spots object:'});
    if vOk<1, return, end
    vSpots = vSpotsList{vPair(1)};
else
    vSpots = vSpotsList{1};
end
```

Position Information in Volume Conversion of µm to Pixel Position

13	14	15	16	17	18
7	8	9	10	11	12
1	2	3	4	5	6



- numbers indicate the index in a 1DSlice in 2D
- volume size in 2D
    - 6x3 (XY)

```
%Create 2D map of XY positions in volume space
 %a way of identifying the pixels masked in surface relative to XYZ spot or
 vSliceXY=[repmat((linspace(vDataMin(1), vDataMax(1),
 vDataSize(1)))',vDataSize(2),1)...
     (repelem(linspace(vDataMin(2), vDataMax(2),
 vDataSize(1)),vDataSize(2)))'];
 %Identify the Z-position from the Slice#
 vSliceZ=(linspace(vDataMin(3), vDataMax(3), vDataSize(3)))';


%This code find the closest pixel position to the sub voxel spot positions
for vIndexZ = 1:vDataSize(3)
        %Select spots that lie closest to current slice
        idx=subvoxelSpotPosition(:,3) > Step(vIndexZ) & ...
            subvoxelSpotPosition(:,3) < Step(vIndexZ+1);
        if idx==true
               CurrentSliceSpots=subvoxelSpotPosition(idx,:);
              %Find the pixel row and column in XY that most closely fits Spot
 position
              CountY=round((CurrentSliceSpots(:,2)-
 vDataMin(2))/Yvoxelspacing,0);
              CountX=round((CurrentSliceSpots(:,1)-
 vDataMin(1))/Xvoxelspacing,0);
               SpotIndexfor1D=(CountY)*vDataSize(1)+CountX;
               %Calculate the total index if assume entire Volume of slices
              SpotIndexWholeVoulme=SpotIndex+(vIndexZ*SliceInterval);
              vSlice(SpotIndexSlice1D)=255;
         end
 end
```

Setting New Spots Object in Imaris Scene

```
vSpotsA = vImarisApplication.GetFactory.CreateSpots;
vSurpassScene = vImarisApplication.GetSurpassScene;
vSpotsA.Set(vSpotsPosXYZ, vSpotsPosT, vSpotsRadius);
vSpotsA.SetName(sprintf('NEW SURFACE'));
vSpotsB.SetColorRGBA(255*256*256);
%Add Spots to scene
vImarisApplication.GetSurpassScene.AddChild(vSpotsA, -1);
```

NOTE: this is the format for the variables (equal # rows required for each one)

```
PositionsXYZ = [10 37 10; 15 20 8];
aIndices = [0 0];% Indices start at zero for first time point
aRadii = [0.5 0.5];
```

Identify All Track IDs for All Spots Tracked

```
vSpots = vImarisApplication.GetSurpassSelection;

edges = vSpots.GetTrackEdges + 1;
edges_forspots = 1:size(vSpots.GetPositionsXYZ, 1);%Get the number of spots or surfaces
edges_forspots( : ) = size(edges, 1) + 1; % initialize array to fictive edge
edges_forspots(edges(:, 1)) = 1:size(edges, 1);
edges_forspots(edges(:, 2)) = 1:size(edges, 1);
trackid_foredges = [vSpots.GetTrackIds; 0]; % add fictive track id
trackid_forspots = trackid_foredges(edges_forspots);
tempx = double(trackid_forspots);
vtrackID = tempx-1000000000;%track IDs as a single integer
vtrackIDmax = max(vtrackID);%Identifies the total number of tracks in the dataset
%Loop each trackID
for trackloop = 0:vtrackIDmax;
                vSpotsT = vtrackID == trackloop;
                tracklength = sum(vSpotsT) ;
                vSpotsIndex = find(vSpotsT);%
      %Loop through each object in the TrackID
      for c = 1:tracklength-1;
                                vWorkingSpotsIds=vIds(vSpotsIndex,:);
                                vWorkingId=double(vWorkingSpotsIds(c));
                                vWorkingTimeIndex=vSpots.GetTimeIndex(vWorkingId);
      end
end
```

Identify All Track IDs for All Surfaces Tracked

```
vSurfaces = vImarisApplication.GetSurpassSelection;

edges = vSurfaces.GetTrackEdges + 1;
edges_forsurfaces = 1:size(Spots.GetPositionsXYZ, 1);%Get the number of spots or surfaces
edges_forsurfaces( : ) = size(edges, 1) + 1; % initialize array to fictive edge
edges_ forsurfaces (edges(:, 1)) = 1:size(edges, 1);
edges_ forsurfaces (edges(:, 2)) = 1:size(edges, 1);
trackid_foredges = [vSurfaces.GetTrackIds; 0]; % add fictive track id
trackid_ forsurfaces = trackid_foredges(edges_ forsurfaces);
tempx = double(trackid_ forsurfaces);
vtrackID = tempx-1000000000;%Track  IDs as a single integer.
vtrackIDmax = max(vtrackID);%Identifies the total number of tracks in the dataset
%Loop each trackID
for trackloop = 0:vtrackIDmax;
                vSurfacesT = vtrackID == trackloop;
                tracklength = sum(vSurfacesT) ;
                vSurfacesIndex = find(vSurfacesT);%
    %Loop through each object in the TrackID do whatever
    for c = 1:tracklength;
                                vWorkingSurfacesIds=vIds(vSurfacesIndex,:);%Get all surface IDs in track
                                vWorkingId=double(vWorkingSurfacesIds(c));%Get first object ID in Track
                                vWorkingTimeIndex=vSurfaces.GetTimeIndex(vWorkingId);
    end
end
```

Loop Each Dendrite

BITPLANE
an Oxford Instruments company

```
 vFilamentsIndexT = vFilaments.GetTimeIndex(FilamentIndex);
 vFilamentsXYZ = vFilaments.GetPositionsXYZ(FilamentIndex);
 vFilamentsRadius = vFilaments.GetRadii(FilamentIndex);
vSegmentIds=unique(vFilamentsEdgesSegmentId);%Idenitfy unique filament
 segmentID
for vBranchIndex=1:(vNumberOfDendriteBranches)
        %Set the ID for dendrite segment
        wSegmentIndex = vSegmentIds(vBranchIndex);
        %Logical argument to identify spots in segment
        vSpotsT = vFilamentsEdgesSegmentId == wSegmentIndex;
        vSpotsIndex = find(vSpotsT');
        %Identify position for dendrite segment
        %Test with new method of filtering
        vDendriteEdgesWorking=vFilamentsEdges(vSpotsIndex,:);
        vEdgesUnique=unique(vDendriteEdgesWorking);
        vDendritePositionsWorking=vFilamentsXYZ(vEdgesUnique,1:3);
        vDendriteRadiusWorking=vFilamentsRadius(vEdgesUnique,1);
        vTypesWorking=vTypes(vEdgesUnique,:);
    end
```

Reorder Filament Points per Dendrite

```
%find single indices, identifying first and last points
vDendriteStartEndindex=find(accumarray(vDendriteEdgesWorking(:),1)==1);
for k=1:numel(vDendritePositionsWorking(:,1))
    if k==1 %for first point
        vDendritePositionsNEW(k,:)=vFilamentsXYZ(vDendriteStartEndindex(1),:);
        test=vDendriteEdgesWorking(:,1)==vDendriteStartEndindex(1);
        if test==false;
            test=vDendriteEdgesWorking(:,2)==vDendriteStartEndindex(1);
        end
        %Find next Filament Index
        vDendriteNextFilamentIndex=vDendriteEdgesWorking(test,:);
        vDendriteNextFilamentIndex(vDendriteNextFilamentIndex==vDendriteStartEndindex(1))=[];
        %remove current edge row
        vDendriteEdgesWorking(test,:)=[];
        test(test,:)=[];

    else % each additional point
        vDendritePositionsNEW(k,:)=vFilamentsXYZ(vDendriteNextFilamentIndex,:);
        test=vDendriteEdgesWorking(:,1)==vDendriteNextFilamentIndex;
        if test==false;
            test=vDendriteEdgesWorking(:,2)==vDendriteNextFilamentIndex;
        end
        vDendriteLastFilamentIndex=vDendriteNextFilamentIndex;
        %Find next Filament Index
        vDendriteNextFilamentIndex=vDendriteEdgesWorking(test,:);
        vDendriteNextFilamentIndex(vDendriteNextFilamentIndex==vDendriteLastFilamentIndex)=[];
        %remove current edge row
        vDendriteEdgesWorking(test,:)=[];
        test(test,:)=[];
    end
end
```

Adding a New Object-Based Statistic

BITPLANE
an Oxford Instruments company

```
%Where the variable NewStat represents the values to be used in the stat. They should be equal to number of surfaces, spots or tracks that are a part to the object.

%Variation #1
        %adding new stat to spot/surface object for each time point in objects %already tracked in Imaris
vInd = 1:numel(NewStat);%Quantifies the number of values.
vIds = vObject.GetIds;%Gets all Ids to correspond to Imaris 8.2 Ids
vUnits(vInd) = { char(vImarisApplication.GetDataSet.GetUnit) };%gets default units (um) and converts to character
vFactors(vInd) = {'Spot'};%Generates
vFactors(2, vInd) = num2cell(vNewSpots.GetIndicesT);
vFactors(2, vInd) = cellfun(@num2str, vFactors(2, vInd), 'UniformOutput', false);
vFactorNames = {'Category', 'Time'};
vNames(vInd) = {'NewStatName'};%Name the statistic as it will appear in Imaris
NewStat=NewStat';%Transposes the values from vertical column to a horizontal arrangement

vNewObject.AddStatistics(vNames, NewStat, vUnits, vFactors, vFactorNames, vIds);
```

OXFORD INSTRUMENTS

Adding a New Track Statistic

BITPLANE
an Oxford Instruments company

```
%Set New Track Statistic
vIndT=1:vtrackIDmax+1;%Total number of tracks
vIdsT=0:vtrackIDmax;%Total number from tracks starting at 0
vIdsT=vIdsT+1000000000;%Conversion to Tracks reported by Imaris
vUnitsT(vIndT) = {'seconds'};
vFactorsT(vIndT) = {'Track'};
vNamesT(vIndT) = {' NewStat Name'};%Statistic name
vFactorNamesT = {'Category'};

vObject.AddStatistics(vNamesT, vContactTime, vUnitsT, vFactorsT, vFactorNamesT,
vIdsT);
```

NOTE: vtrackIDmax, calculated from previous script to identify trackIDs for all objects

Adding a New Overall Statistic

```
    vInd=1:aSizeT;
    vIds(vInd)=0;%For overall stats all Ids are equal to 0
    vUnits(vInd) = {‘UnitName'};%{
char(vImarisApplication.GetDataSet.GetUnit) };
    Indices=1:aSizeT;%These range for each time point
starting at 1
    vFactors(vInd) = {'Overall'};
    vFactors(2, vInd) = num2cell(Indices);
    vFactors(2, vInd) = cellfun(@num2str, vFactors(2,
vInd), 'UniformOutput', false);
    vFactorNames = {'Overall','Time'};
    vNames(vInd) = {sprintf(‘NEW stat NAME’)};
    vSpots.AddStatistics(vNames,
PercentageContactsperTimpoint', vUnits, vFactors,
vFactorNames, vIds);
```

Extracting Statistics from Surpass Objects

```
%Get All Statistics
    vAllStatistics = vSpots.GetStatistics;
    vNames       = cell(vAllStatistics.mNames);
    vValues      = vAllStatistics.mValues;
    vUnits       = cell(vAllStatistics.mUnits); % not used
    vFactors     = cell(vAllStatistics.mFactors);
    vFactorNames = cellstr(char(vAllStatistics.mFactorNames));
    vIds         = vAllStatistics.mIds;
    vObjectIndex=strmatch('Intensity Min', vNames);
    vObjectValues = vValues(vObjectIndex,:);
```

Setting Colors of Objects

BITPLANE
an Oxford Instruments company

```
vRGBA=[255 0 0 0];%red
vRGBA=[0 255 0 0];%green
vRGBA=[0 0 255 0];%blue
vRGBA = uint32(vRGBA * [1; 256; 256*256; 256*256*256]);
vNewObject.SetColorRGBA(vRGBA);
```

OXFORD INSTRUMENTS

Find Imaris Version, Quit Imaris, Start New Instance with Next Imaris ApplicationID

```
%Get Imaris version
aVersion = char(vImarisApplication.GetVersion);
aImarisFolderEnd = strfind(aVersion, ' [');
if numel(aImarisFolderEnd) ~= 1
    msgbox('Invalid Imaris version')
    return
end
aImarisFolder = aVersion(1:aImarisFolderEnd);
aDelimiters = strfind(aImarisFolder, '-');
if numel(aDelimiters) == 2
    aImarisFolder(aDelimiters(2)) = [];
    aImarisFolder(aDelimiters(1)) = ' ';
end

%Quit Imaris Application
vImarisApplication.SetVisible(~vImarisApplication.GetVisible);
vImarisApplication.Quit;

%Start new Imaris instance with ID99
eval(['! C:\\Program Files\\Bitplane\\', aImarisFolder, '\\Imaris.exe &'])

%%

aImarisApplicationID=aImarisApplicationID+1;
vImarisApplication = vImarisLib.GetApplication(aImarisApplicationID)
```

BITPLANE

an Oxford Instruments company

Find us on