20130125 Experiments-Compare Affine and Syn

20130125 - Experiments-Compare Affine and Syn

This page should fill in any important notes about the project. Think of it as a page in a global lab notebook.

1. Description of data to be processed
2. Scripts used to process the data
3. Failed processing attempts
4. Successful processing attempts

New Meeting Notes Instructions -- Remove this note when finished!

- Remove this note when ready to publish
- Recommended Page Notes Name: YYYMNMDD <<Note contents>>
- OPTIONAL Includes :
- OPTIONAL Dynamic Tasklist : {dynamictasklist:ToDo DESCRIP YYMMDD}
- OPTIONAL Table of Contents: {toc:style=outline|maxLevel=2}

Related JIRA ticket

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Data Description

- **Working Dir:** /IPLlinux/hjohnson/HDNI/20130225_BRAINSABC_Publication/BRAINSABC_Affine
- Roughly pick two sessions per site regarding variety of ventricle sizes: We are regarding ventricle size as a indicative of brain shrinkage. Dave is working on providing measures of CSF, (WM+GM)/CSF to see if those are better for indicative. I will pick data set baseed on those.

BSD (Brain Simulated Data from Montreal Group) Test

Directory:

/IPLlinux/hjohnson/HDNI/20130125_BRAINSABC_Publication/BRAINSABC_brainweb/SegmentationComparison/CompareToGroundTruth

Python Script to Compute Similarities:

```
from pinc import *
def GetTotalMaskFilename( inputList, outputFilename, threshold ):
    import SimpleITK as sitk
    sumOfImage = None
    for image in inputList:
        if sumOfImage is None:
            sumOfImage = sitk.ReadImage( image )
        else:
            sumOfImage = sumOfImage + sitk.ReadImage( image)
    import os
    if not os.path.exists( os.path.dirname( os.path.abspath( outputFilename ) ) ):
        os.mkdir( os.path.dirname( os.path.abspath( outputFilename ) ) )
    sitk.WriteImage( sumOfImage, outputFilename )
    binaryFilename = outputFilename + "_binary.nii.gz"
    sumOfImage_binary = sitk.BinaryThreshold( sumOfImage, threshold, 1.01)
    sitk.WriteImage( sumOfImage_binary , binaryFilename )
    return( os.path.abspath( binaryFilename ) )
def Compare( threshold, outputCSV ):
```

```
    import SimpleITK as sitk
    statDict = []
    for pnNo in [0, 1, 3, 5, 7, 9]:
        pn= "pn" + str( pnNo )
        for rfNo in [20,40]:
            rf = "rf" + str( rfNo )
            # reference WM (rf=0 with same noise level
            refGM = GetTotalMaskFilename( [ "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_CRBLGM.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_HIPPOCAMPUS.
nii.gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_PUTAMEN.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_SURFGM.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_THALAMUS.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_CAUDATE.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_ACCUMBEN.nii.
gz"],
                                            "./ACCUMULATED/Image_1mm_" + pn + "_rf0_TOTAL_GM" + str
(threshold) +".nii.gz",
                                          threshold)
            # current RF
            biasedGM = GetTotalMaskFilename( ["../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_CRBLGM.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_HIPPOCAMPUS.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_PUTAMEN.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_SURFGM.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_THALAMUS.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_CAUDATE.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_ACCUMBEN.nii.gz"],
                                              "./ACCUMULATED/Image_1mm_" + pn + "_" + rf + "_TOTAL_GM" +
str(threshold) +".nii.gz",
                                            threshold )
            import analysis
            autoFilename = biasedGM
            refFilename = refGM
            defFilename = refGM
            autoLabel = 1
            roi = "GM"
            session = pn + rf
            statDict.append( analysis.computeSimilarity( autoFilename, defFilename, refFilename,
autoLabel, roi, session) )
            # reference WM (rf=0 with same noise level
            refWM = GetTotalMaskFilename( [ "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_CRBLWM.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_WM.nii.gz"],
                                            "./ACCUMULATED/Image_1mm_" + pn + "_rf0_TOTAL_WM" + str
(threshold) +".nii.gz",
                                          threshold)
            # current RF
            biasedWM = GetTotalMaskFilename( ["../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_CRBLWM.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine/POSTERIOR_WM.
nii.gz"],
                                              "./ACCUMULATED/Image_1mm_" + pn + "_" + rf + "_TOTAL_WM" +
str(threshold) +".nii.gz",
                                            threshold )
            import analysis
            autoFilename = biasedGM
            refFilename = refGM
            defFilename = refGM
            autoLabel = 1
```

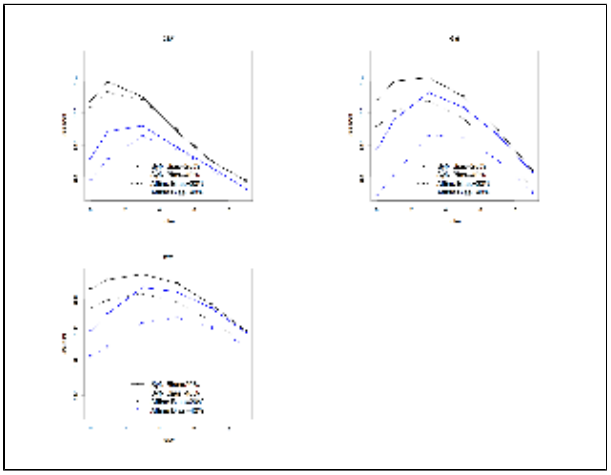
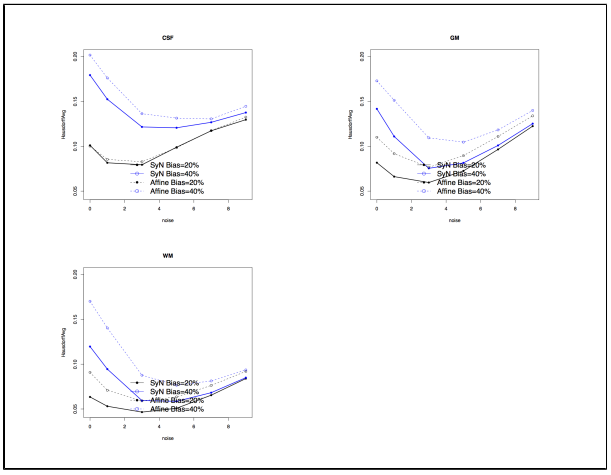
```
            roi = "GM"
            session = pn + rf
            statDict.append( analysis.computeSimilarity( autoFilename, defFilename, refFilename,
autoLabel, roi, session) )
            # reference WM (rf=0 with same noise level
            refWM = GetTotalMaskFilename( [ "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_CRBLWM.nii.
gz",
                                            "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_WM.nii.gz"],
                                            "./ACCUMULATED/Image_1mm_" + pn + "_rf0_TOTAL_WM" + str
(threshold) +".nii.gz",
                                          threshold)
            # current RF
            biasedWM = GetTotalMaskFilename( ["../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_CRBLWM.nii.gz",
                                              "../../Image_1mm_" + pn + "_" + rf + "_Affine/POSTERIOR_WM.
nii.gz"],
                                              "./ACCUMULATED/Image_1mm_" + pn + "_" + rf + "_TOTAL_WM" +
str(threshold) +".nii.gz",
                                            threshold )
            import analysis
            autoFilename = biasedWM
            refFilename = refWM
            defFilename = refWM
            autoLabel = 1
            roi = "WM"
            session = pn + rf
            statDict.append( analysis.computeSimilarity( autoFilename, defFilename, refFilename,
autoLabel, roi, session) )
            # reference CSF (rf=0 with same noise level
            refCSF = GetTotalMaskFilename( [ "../../Image_1mm_" + pn + "_rf0_Affine/POSTERIOR_CSF.nii.
gz"],
                                            "./ACCUMULATED/Image_1mm_" + pn + "_rf0_TOTAL_CSF" + str
(threshold) +".nii.gz",
                                          threshold)
            # current RF
            biasedCSF = GetTotalMaskFilename( ["../../Image_1mm_" + pn + "_" + rf + "_Affine
/POSTERIOR_CSF.nii.gz"],
                                              "./ACCUMULATED/Image_1mm_" + pn + "_" + rf + "_TOTAL_CSF" +
str(threshold) +".nii.gz",
                                            threshold )
            import analysis
            autoFilename = biasedCSF
            refFilename = refCSF
            defFilename = refCSF
            autoLabel = 1
            roi = "CSF"
            session = pn + rf
            statDict.append( analysis.computeSimilarity( autoFilename, defFilename, refFilename,
autoLabel, roi, session) )

    # <codecell>
    import analysis
    analysis.writeCSV( statDict, outputCSV )
    # <codecell>
Compare( 0.51, "./compare_to_groundTruth_Threshold_0.51_Affine.csv" )
```

R Script for Plot

```
wholeDt<-read.csv("compare_to_groundTruth_Threshold_0.51.csv", head=T)
for( i_roi in levels( factor( wholeDt$roi ) ) )
{
pdf( paste( i_roi, "_compare.pdf", sep=""))
i_rf = levels( factor(wholeDt$rf))[1]
  dt<-subset( wholeDt, wholeDt$rf == i_rf & wholeDt$roi == i_roi &  method == "SyN" )
  print( dt )
  plot( dt$noise, dt$SimilarityIndex,
        type="o",
        pch=20,
        lwd=2,
        lty=1,
        ylim= range( wholeDt$SimilarityIndex ),
        xlab="noise",
        ylab="Dice Index"
        )
  title( i_roi )
i_rf = levels( factor(wholeDt$rf))[2]
  dt<-subset( wholeDt, wholeDt$rf == i_rf & wholeDt$roi == i_roi &  method == "SyN")
  print( dt )
  points( dt$noise, dt$SimilarityIndex,
        type="o",
        pch=20,
        lwd=2,
        lty=1,
        col="blue"
        )
i_rf = levels( factor(wholeDt$rf))[1]
  dt<-subset( wholeDt, wholeDt$rf == i_rf & wholeDt$roi == i_roi &  method == "affine" )
  print( dt )
  points( dt$noise, dt$SimilarityIndex,
        type="o",
        pch=21,
        lty=2
        )
i_rf = levels( factor(wholeDt$rf))[2]
  dt<-subset( wholeDt, wholeDt$rf == i_rf & wholeDt$roi == i_roi &  method == "affine")
  print( dt )
  points( dt$noise, dt$SimilarityIndex,
        type="o",
        pch=21,
        lty=2,
        col="blue"
        )
  legend( "bottom",
          c("SyN Bias=20%", "SyN Bias=40%", "Affine Bias=20%","Affine Bias=40%"),
          col=c("black", "blue"),
          lty=c(1,1,2,2),
          pch=c(20,21),
          bty="n",
          cex=1.5
        )
dev.off()
}
```

Plot: Well at lease onething worked..

Dice Index	Hausdorff Distance Average
	

Tasks attempted description

- Check if BRAINSABC works with Affine only registration.

Working Dir: /IPLlinux/hjohnson/HDNI/20130225_BRAINSABC_Publication/BRAINSABC_Affine

```
inputDir="/IPLlinux/hjohnson/HDNI/20130225_BRAINSABC_Publication/BRAINSABC_Affine"
BRAINSABC  \
    --atlasDefinition $inputDir/ExtendedAtlasDefinition.xml \
    --atlasToSubjectInitialTransform $inputDir/landmarkInitializer_atlas_to_subject_transform.h5 \
    --atlasToSubjectTransform atlas_to_subject.h5 \
    --atlasToSubjectTransformType Affine \
    --debuglevel 10 \
    --filterIteration 3 \
    --filterMethod GradientAnisotropicDiffusion \
    --gridSize 10,10,10 \
    --inputVolumeTypes T1,T1,T2 \
    --inputVolumes $inputDir/subject_t1_01.nii.gz \
    --inputVolumes $inputDir/subject_t1_02.nii.gz \
    --inputVolumes $inputDir/subject_t2_01.nii.gz \
    --interpolationMode Linear \
    --maxBiasDegree 4 \
    --maxIterations 3 \
    --outputDir ./ \
    --outputDirtyLabels volume_label_seg.nii.gz \
    --outputFormat NIFTI \
    --outputLabels brain_label_seg.nii.gz \
    --outputVolumes subject_T1-30_2_corrected.nii.gz \
    --outputVolumes subject_T1-30_3_corrected.nii.gz \
    --outputVolumes subject_T2-30_4_corrected.nii.gz \
    --posteriorTemplate POSTERIOR_%s.nii.gz
```