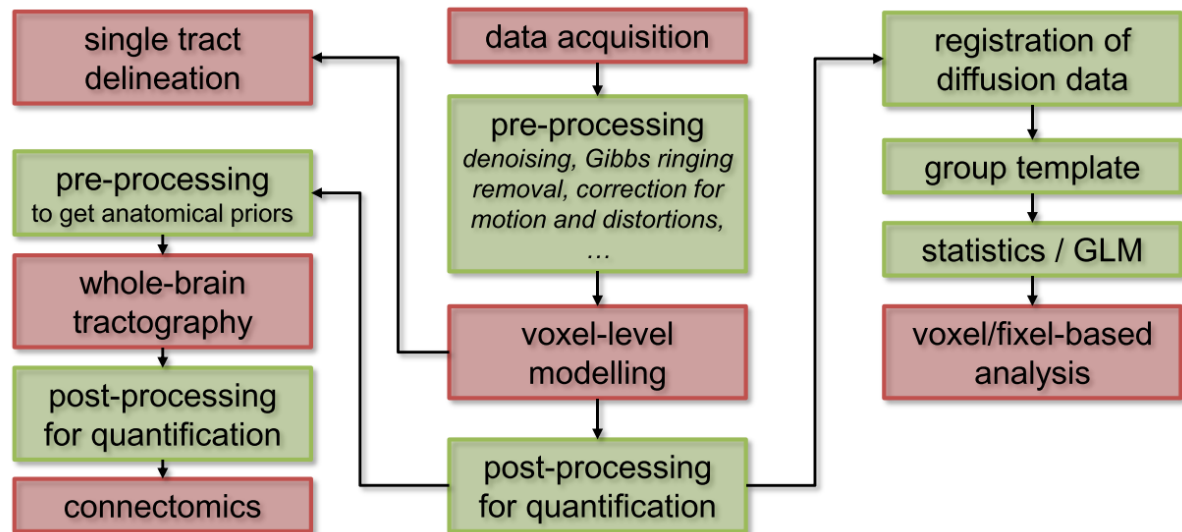


Pipeline MRtrix3 – DWI



Revisão comandos simples

\$ ls	#lista de arquivos no diretório atual
\$ ls data/	#lista de arquivos em um diretório específico
\$ ls -l	#lista de arquivos e informação de modificações
\$ ls --help	#mostra como utilizar o comando
\$ man ls	#abre a página de ajuda em uma guia do navegador
\$ cd	#mudar o diretório
\$ pwd	#printa o diretório de trabalho atual
\$ cp	#copia os arquivos
\$ mv	#move ou renomeia arquivos
\$ rm	#remove arquivos
\$ ls bin/	#mostra todos os comandos disponíveis
\$... -info	#mostra mensagens informativas, para mais detalhes -info flag
\$... -quiet	#não mostra o progresso ou mensagens informativas
\$... -debug	#reporta bugs, para mais detalhes use -debug flag
\$... -force	#força a sobrescrita no arquivo de output
\$... -version	#mostra a versão utilizada

Dica 1: utilize TAB para completar o comando que você está escrevendo.

Dica 2: pressione ↑↓ para rever comandos usados anteriormente.

Dica 3: você pode carregar diversas imagens como uma só: \$ mrconvert dwi.mif dwi-[]].mif

Obs: MRView lida com todas as operações de exibição nas coordenadas do scanner.

Streamlines (linhas de fluxo) é o nome utilizado para a representação computacional da fibra axonal.

Pipeline Processamento Tractografia e Conectoma com Mrtrix3

Obs: input (dado de entrada) = roxo and output (dado de saída) = azul

Para realizar esse processamento são necessárias 4 arquivos diferentes para cada indivíduo: dois arquivos formato NIFTI (.nii) e dois arquivos formato texto (.bvec e .bval). Os arquivos .nii se referem as imagens de difusão (DWI/DTI) e anatômica (3DT1/T1), já os arquivos texto são complementares da imagem de difusão e trazem os valores dos gradientes de difusão em x,y,z (.bvec) e os valores de b (.bval).

Caso utilize dados em formato DICOM, não é preciso fazer primeira parte, é só converter para o formato .mif através do comando:

```
$ mrconvert dwi.dicom dwi.mif.
```

Na pasta com os dados abra o terminal:

1) Conversão dos dados de difusão para formato MRtrix3

#conversao NIFT→MIF

```
$ mrconvert dwi.nii -fslgrad dwi.bvec dwi.bval dwi.mif
```

#checar a conversão

```
$ mrinfo -dwgrad dwi.mif | -norealign dwi.mif | -petable dwi.mif
```

2) Pré-processamento dos dados de difusão

Nessa parte serão feitas correções de distorções, movimento, entre outras.

(I) Remoção de ruídos

#para acelerar o processamento utilizamos uma máscara da substância branca dilatada

```
$ dwi2mask dwi.mif -|maskfilter -dilate preproc_mask.mif -npass 5
```

#correção do ruído

```
$ dwidenoise dwi.mif dwi_denoised.mif -noise dwi_noise.mif - mask preproc_mask.mif
```

#cálculo do ruído

```
$ mrcalc dwi.mif dwi_denoised.mif -subtract dwi_residual.mif
```

#visualização do ruído que foi removido

```
$ mrview dwi_residual.mif
```

(II) Correção da distorção de Gibbs

#correção de gibbs

```
$ mrdegibbs dwi_denoise.mif dwi_degibbs.mif
```

#checar correção

```
$ mrview dwi_degibbs.mif
```

(III) Processamento FreeSurfer (correção da corrente de eddy, dos movimentos do sujeito e da distorção do campo magnético)

#processamento fsl

```
$ dwifslpreproc -rpe_none -pe_dir AP dwi_degibbs.mif dwi_preproc.mif
```

#checar preprocessamento

```
$ mrview dwi_preproc.mif
```

(IV) Máscara da substância branca

máscara de substância branca automática

```
$ dwi2mask dwi_preproc.mif dwi_mask.mif
```

#checar a máscara produzida

```
$ mrview dwi_mask.mif
```

#como é automático pode haver erros e ter locais que não cobre direito a substância branca, caso isso ocorra será necessário fazer a correção a mão utilizando a opção de draw ROI no mrview.

(V) Correção distorção bias

#correção do campo bias

```
$ dwibiascorrect ants -mask dwi_mask.mif dwi_preproc.mif dwi_bias.mif -bias biasfield.mif
```

#checar a correção

```
$ mrview biasfield.mif
```

(VI) Normalização de intensidade (apenas quando há um número grande de indivíduos)

#normalização de intensidade através dos sujeitos

```
$ dwinormalise dwi_bias.mif dwi_mask.mif dwi_norm.mif fa_template.mif wm_mask.mif -force
```

3) Corregistro imagem anatômica e imagem de difusão (software FSL em conjunto com MRtrix3)

O corregistro entre a imagem anatômica e de difusão é realizado, pois ambas estão em espaços diferentes, se você abrir as duas no mrview verá que elas não ficam sobrepostas uma sobre a outra certinho. Assim, como precisamos da parte anatômica para fazer o parcelamento das regiões corticais, temos que levar a imagem do espaço da T1 para o espaço da DWI.

#extrair b0

```
$ dwiextract dwi.mif -bzero - | mrmath -axis 3 - mean b0.nii
```

#remoção da calota craniana na imagem T1

```
$ fsl5.0-bet T1.nii T1_bet.nii -R
```

#alinhamento das imagens anatômica e de difusão

```
$ fsl5.0-flirt -dof 6 -cost normmi -in T1.nii -ref b0 -omat T1_fsl.txt
```

#cálculo da matrix para corregistrar as imagens anatômica e de difusão

```
$ transformconvert T1_fsl.txt T1w.nii b0.nii flirt_import T1toDWI.txt
```

#corregistro da imagem anatômica no espaço da imagem de difusão

```
$ mrtransform -linear T1toDWI.txt T1w.nii T1corregist.nii
```

4) Segmentação dos tecidos

Nessa parte é realizada a segmentação dos tecidos cerebrais que diz qual parte da imagem é substância branca, cinzenta, líquido etc.

#segmentação dos tecidos

```
$ 5ttgen fsl -nocrop -sgm_amyg_hipp T1corregist.nii 5ttseg.mif -force
```

#checar a segmentação

```
$ mrview 5ttseg.mif -mode 4
```

#seleção da segmentação da substância branca e cinzenta

```
$ 5tt2gmwmi 5tt.mif 5ttgmwm.mif
```

#visualização da segmentação

```
$ 5tt2vis 5ttseg.mif 5ttvis.mif
```

```
$ mrview 5ttvis.mif
```

5) Tractografia

Para gerar as tractografias existem diferentes algoritmos que você pode utilizar para fazer a reconstrução. Dentre os mais utilizados temos os algoritmos FACT (determinístico), CSD-iFOD2 (probabilístico) e MT-CSD-iFOD2 (multi-tecido probabilístico).

FACT: Fibra atribuída por rastreamento contínuo. Um algoritmo determinístico que toma como entrada uma imagem 4D, com volumes 3xN, onde N é o número máximo de orientações da fibra em um voxel. Cada tripleto de volumes representa um vetor 3D correspondente a uma orientação de fibra; o comprimento do vetor adicionalmente indica alguma medida de densidade ou anisotropia. Conforme as streamlines se movem de um voxel para outro, a orientação da fibra mais colinear com a orientação da streamline é selecionada (ou seja, não há interpolação intra-voxel).

iFOD2 (padrão): Integração de segunda ordem em distribuições de orientação de fibra. Um algoritmo probabilístico que leva como entrada uma imagem de Distribuição de Orientação de Fibra (FOD) representada na base de Harmônicos Esféricos (SH). Linhas de fluxo (streamlines) de candidatos (com base em “arcos” curtos) são desenhados e as amplitudes FOD subjacentes (interpoladas trilineares) ao longo desses arcos são amostradas. Uma linha aerodinâmica é mais provável de seguir um caminho onde as amplitudes FOD ao longo desse caminho são grandes; mas também raramente pode atravessar orientações onde as amplitudes de FOD são pequenas, desde que a amplitude permaneça acima do limite de amplitude de FOD ao longo de todo o caminho.

Para mais informações acesse:

<https://mrtrix.readthedocs.io/en/latest/reference/commands/tckgen.html>

(I) Algoritmo de reconstrução FACT

#cálculo do tensor

```
$ dwi2tensor -mask dwi_mask.mif dwi_norm.mif dwi_tensor.mif
```

#extração dos parâmetros

```
$ tensor2metric -adc dwi_md.mif -fa dwi_fa.mif -ad dwi_ad.mif -rd dwi_rd.mif -value dwi_values.mif -vector dwi_vec.mif -mask dwi_mask.mif dwi_tensor.mif
```

#construção da tractografia

```
$ tckgen -algorithm FACT dwi_vec.mif FACT.tck -select 1M -seed_image dwi_mask.mif -act 5ttseg.mif -backtrack -crop_at_gmwmi -angle 35 -cutoff 0.06
```

(II) Algoritmo de reconstrução CSD-iFOD2

#estimar a função resposta para a deconvolução esférica

```
$ dwi2response tournier dwi_norm.mif wm_response.txt
```

#verificar a função resposta resultante

```
$ shview wm_response.txt
```

#estimar a distribuição de orientação das fibras usando deconvolução esférica (CSD)

```
$ dwi2fod csd -mask mask.mif dwi_norm.mif wm_response.txt fod.mif
```

#checar o resultado

```
$ mrview fod.mif -odf.load_sh fod.mif
```

#construção da tractografia

```
$ tckgen -algorithm iFOD2 fod.mif CSD.tck -select 1M -seed_dynamic fod.mif -act 5ttseg.mif -  
backtrack -crop_at_gmwmi -cutoff 0.06
```

Após gerar a tractografia ainda pode-se passar um filtro com o intuito retirar possíveis streamlines falsas. Para isso existem dois tipos de filtro que podem ser utilizados: SIFT ou SIFT2.

i) SIFT – filtra a densidade de linhas de fluxo (streamlines) para corresponder a função de orientação das fibras.

```
$ tcksift CSD.tck fod.mif CSD_SIFT.tck -act 5TT.mif -term_number 10M
```

ii) SIFT2 – filtra a tractografia para que corresponda as densidades de fibra fixel

#cálculo do filtro

```
$ tcksift2 CSD.tck fod.mif tck_weights.txt -act 5ttseg.mif -out_mu CSD_SIFT2.txt -out_coeffs  
csd_tck_coeffs.txt
```

#visualização

```
$ head -c 200 csd_tck_weights.txt
```

```
$ cat CSD_SIFT2.txt
```

```
$ mrview 5ttvis.mif -plane 2 -tractography.load CSD.tck -tractography.tsf_load  
csd_tck_coeffs.txt -tractography.tsf_range -1,1
```

#mapeamento das fibras agora com o filtro

```
$ tckmap CSD.tck CSD_SIFT2.mif \ -template mask.mif -precise -tck_weights_in tck_weights.txt
```

#visualização

```
$ mrview fod.mif \ CSD_SIFT2.mif CSD_SIFT2.mif
```

(III) Algoritmo de reconstrução MT-CSD (só se aplica a imagens multi-tecido)

#estimar a função resposta para a deconvolução esférica

```
$ dwi2response dhollander dwi_norm.mif wm_response.txt gm_response.txt csf_response.txt
```

#verificar a função resposta resultante

```
$ shview wm_response.txt
```

#estimar a distribuição de orientação das fibras usando deconvolução esférica multi-tecido (CSD)

```
$ dwi2fod msmt_csd -mask mask.mif dwi_norm.mif wm_response.txt wm_fod.mif  
gm_response.txt gm_fod.mif csf_response.txt csf_fod.mif
```

#visualização

```
$ mrconvert -coord 3 0 wm_fod.mif - | mrcat csf_fod.mif gm_fod.mif - vf_rgb.mif
```

```
$ mrview vf_rgb.mif -odf.load_sh wm_fod.mif
```

#construção da tractografia

```
$ tckgen -algorithm iFOD2 wm_fod.mif MTCSD.tck -select 1M -seed_dynamic wm_fod.mif -act  
5ttseg.mif -backtrack -crop_at_gmwmi -cutoff 0.06
```

Após gerar a tractografia ainda pode-se passar um filtro com o intuito retirar possíveis streamlines falsas. Para isso existem dois tipos de filtro que podem ser utilizados: SIFT ou SIFT2.

i) SIFT – filtra a densidade de linhas de fluxo (streamlines) para corresponder a função de orientação das fibras.

#filtragem da tractografia

```
$ tcksift MTCSD.tck wm_fod.mif SIFT.tck -act 5TT.mif -term_number 10M
```

ii) SIFT2 – filtra a tractografia para que corresponda as densidades de fibra fixel

#cálculo do filtro

```
$ tcksift2 MTCSD.tck wm_fod.mif tck_weights.txt -act 5ttseg.mif -out_mu MT_SIFT2.txt -  
out_coeffs mt_tck_coeffs.txt
```

#visualização

```
$ head -c 200 tck_weights.txt
```

```
$ cat MT_SIFT2.txt
```

```
$ mrview 5ttvis.mif -plane 2 -tractography.load MTCSD.tck -tractography.tsf_load  
mt_tck_coeffs.txt -tractography.tsf_range -1,1
```

#mapeamento das fibras agora com o filtro

```
$ tckmap MTCSD.tck MT_SIFT2.mif \ -template mask.mif -precise -tck_weights_in  
tck_weights.txt #can use MT_SIFT2.tck
```

#visualização

```
$ mrview wm_fod.mif \ MT_SIFT2.mif MT_SIFT2.mif
```

6) Segmentação de tratos específicos

Para realizar a segmentação de tratos específicos, como o Cortico Espinhal, por exemplo, você irá utilizar a ferramenta mrview. Na interface gráfica do mrview vá até ferramentas (tools) e selecione a opção “ROI editor”. Lá você poderá desenhar as rois de inclusão e exclusão de acordo com o trato que você deseja segmentar e salvar essas rois como arquivos .mif. Com as rois salvas no seu computador, digite a linha de comando a baixo para segmentar.

```
$ tckedit tractografia.tck -include ROI01.mif -include ROI02.mif -exclude ROI03.mif  
tratosegmentado.tck
```

7) Conectoma

O conectoma representa como as diferentes regiões estão conectadas estruturalmente, por meio das fibras de substância branca. Então ao processar o conectoma o resultado será uma matriz de n colunas por n linhas, e cada valor irá representar o quão forte é a relação entre duas regiões cerebrais, assim a diagonal será 1 (valor máximo), representando a conexão da região com ela mesma.

O tamanho da matriz irá depender do atlas utilizado para o parcelamento das regiões, como exemplo, se utilizado o atlas Destrieux serão 84 regiões no total gerando uma matriz de conectividade de 84x84. No parcelamento as regiões são rotuladas como nós/nodes, assim o nó 1 irá se referir a uma região, o 2 a outra e assim por diante, sendo que isso é definido no arquivo texto do atlas utilizado para o parcelamento.

Existem 2 opções de parcelamento que se pode fazer no MRtrix3 para construção da conectividade estrutural:

- A. Parcelamento do cérebro reconstruído no espaço do indivíduo (isso é feito com o software FSL e pode ser feito com o atlas Desikan Killiany ou Destrieux), para mais informações acesse: <https://surfer.nmr.mgh.harvard.edu/fswiki/CorticalParcellation>

- B. Parcelamento do cérebro transformado para o espaço do indivíduo (Utilizando a extensão ANTs para realizar o parcelamento com o atlas Automated Anatomical Labelling - AAL)

A. FSL

#reconstrução cortical com o FSL, faz diversos passos e realiza o parcelamento com os dois atlas (Desikan e Destrieux), esse processamento demora em média 7 horas para terminar
\$ recon-all -i T1w.nii -s SUBJECT_01 -all -qcach

Passos do processamento

1. Motion Correction and Conform (correção de movimento)
2. NU (Non-Uniform intensity normalization) (corrige a intensidade nos dados de RM)
3. Talairach transform computation (calcula a transformação afim do volume de origem)
4. Intensity Normalization 1 (normalização da intensidade do volume de origem)
5. Skull Strip (remove o crânio – brainmask)
6. Automatic subcortical segmentation (6 passos : -CGA registro inicial do modelo, normalização canônica, registro, remoção do pescoço, registro, marcação subcortical)
7. EM Register (linear volumetric registration) (alinha volume da RM com o atlas GCA)
8. CA Intensity Normalization (posterior normalização baseado no modelo GCA)
9. CA Non-linear Volumetric Registration (transformação não linear para alinhar com atlas GCA)
10. Remove Neck (remoção do pescoço)
11. LTA with Skull (alinha volume da rm com o volume do atlas GCA com o crânio)
12. CA Label (Volumetric Labeling, ie Aseg) and Statistics (registra as estruturas corticais, baseado-se no modelo GCA)
13. Aseg Stats (estatísticas sobre as estruturas subcorticais segmentadas encontradas na RM)
14. Intensity Normalization 2 (start here for control points) (segunda correção de intensidade usando apenas o volume do cérebro)
15. White matter segmentation (tenta separar a substância branca do resto)
16. Edit WM With Aseg
17. Fill (start here for wm edits) (cria a massa subcortical a partir da qual a superfície original é criada)
18. Tessellation (begins per-hemisphere operations) (superfície original é criada)
19. Smooth1 (depois da tessalação, a superfície original é muito irregular porque cada triangulo está na borda de uma face do voxel – angulos retos. Assim, as posições do vertices são ajustadas ligeiramente para reduzir o ângulo – necessario para proceddos de inflação)
20. Inflate1 (a inflação tenta minimizar as distorções métricas para preservar as distancias e as áreas)
21. QSphere (fixação automática de topologia)
22. Automatic Topology Fixer (acha erros topológicos e muda a superfície de origem)
23. Final Surfs (start here for brain edits for pial surf) (faz com que a substância branca siga de perto o gradiente de intensidade de cinza, conforme encontrado no volume T1) (passo depois da tessalação)
24. Smooth2 (passo depois de fixada a topologia)
25. Inflate2 (a inflação tenta minimizar a distorção métrica de modo que as distancias e as áreas sejam preservadas – ou seja, a superfície não seja esticada. O inflate 1 é o passo logo após a tessalação e o inflate 2 é o passo depois da fixação da topologia)
26. Spherical Mapping (etapa inicial de fixação automática de topologia. Transformação esférica da superfície inflada para localizar defeitos topológicos)
27. Spherical Registration (Registra a superfície original para o atlas esférico)
28. Spherical Registration, Contralateral hemisphere (registra com a contralateral atlas)
29. Map average curvature to subject (Restaura a curvatura média do atlas para a do sujeito)
30. Cortical Parcellation - Desikan_Killiany and Christophe (Labeling) (Atribui um indicador neuroanatômico a cada localização na superfície cortical)
31. Cortical Parcellation Statistics (mris_anatomical_stats - cria uma tabela de resumo de estatísticas de parcelas corticais para cada estrutura)
32. Cortical Ribbon Mask (Cria máscaras de volume binário da fita cortical, ou seja, cada voxel é 1 ou 0, dependendo se ele cai na faixa ou não)
33. Cortical Parcellation mapping to Aseg (comprime o resultado em formato .mgz)

Output: apar+aseg.mgz (parcelamento Desikan-Killiani) e aparc.a2009s+aseg.mgz (parcelamento Destrieux)

Se o processamento for realizado no computador ISIS, os resultados do recon all poderão ser encontrados na pasta home→public→freeesufer600→subjects→SUBJECT_01/nome que você der→mri→aparc.a2009s+aseg.mgz

Agora nós iremos utilizar o parcelamento para fazer a relação entre nó e região cerebral, ou seja, no final desse processamento, cada região será representada por um nó. Essa associação ou rotulação é feita por meio dos arquivos fs_default.txt (DK) e fs_a2009s.txt (Destrieux). Se você abrir esses arquivos como nota verá que cada linha representa uma região, então se no parcelamento o atlas tinha 84 regiões, você terá 84 nós, em que o número representará qual região é (1 = ctx-lh-frontalpole, por exemplo).

```
$ ls /opt/mrtrix3/share/mrtrix3/labelconvert
$ head -n 20 fs_default.txt
$ labelconvert -h
#convertendo o resultado obtido com o recon all para formato mrtrix
$ labelconvert apar+aseg.mgz $FREESURFER_HOME/FreeSurferColorLUT.txt fs_default.txt
  parc_fs.mif
#rotulando as regiões do parcelamento
$ labelsghmfix parc_fs.mif dwi.nii fs_default.txt fixSGM.mif

#alinhando todas as imagens no mesmo espaço dwi
$ mrtransform fixSGM.mif -linear T1toDWI.txt nodes.mif -inverse -force
$ mrconvert nodes.mif nodes_new.mif -datatype uint32
#geração do conectoma – se quiser fazer com o CSD basta substituir FACT.tck por
CSD_SIFT2.mif/MT_SIFT2.mif
$ tck2connectome FACT.tck nodes_new.mif connectome.csv -out_assignments streamlines.csv
```

B. Parcelamento usando ANTs e atlas AAL

```
#cálculo do parcelamento
$ antsRegistrationSyNQuick.sh -d 3 \
-f ~/Applications/fsl/data/standard/MNI152_T1_1mm.nii.gz \
-m T1w.nii.gz -o ANTs_ -n 8
#transformação
$ WarpImageMultiTransform 3 ROI_MNI_V4.nii wrapped_aal.nii \
-R T1w.nii.gz -i ANTs_0GenericAffine.mat
  ANTs_1InverseWarp.nii.gz -use-NN

$ ls /opt/mrtrix3/share/mrtrix3/labelconvert
$ head -n 20 ROI_MNI_V4.txt
#inspecionar o parcelamento
$ mrview ../T1w.mif -mode 2 -voxel 122,81,183 \
  -overlay.load wrapped_aal.nii -overlay.opacity 0.2 \
  -overlay.colourmap 3 -overlay.interpolation 0 &

$ ls share/mrtrix3/labelconvert
$ head -n 20 aal.txt
```



```
$ labelconvert -h
```

```
#rotulando as regiões do parcelamento
```

```
$ labelconvert wrapped_aal.nii ROI_MNI_V4.txt aal.txt parc_aal.mif
```

```
# geração do conectoma
```

```
$ tck2connectome FACT.tck parc_aal.mif connectome.csv -tck_weights_in tck_weights.txt -  
out_assignments assignments.txt
```

```
$ head -n 20 assignments.txt
```

- Conectoma ponderado em FA (anisotropia fracionada)

A opção default do conectoma utiliza o número de streamlines (linhas de fluxo) que chegam no nó para saber a força na conexão. Porém, é possível utilizar outros parâmetros para essa consideração, como a anisotropia fracionada (FA) ou a difusividade média (MD). Aqui apresento o exemplo de como fazer com FA, mas a ideia é a mesma para os outros parâmetros.

```
#tractografia ponderada em FA
```

```
$ tcksample FACT.tck dwi_fa.mif tck_meanFA.txt -stat_tck mean
```

```
#geração do conectoma
```

```
$ tck2connectome FACT.tck parc_aal.mif connectome_fa.csv -scale_file tck_meanFA.txt -  
stat_edge mean
```

- Comando connectome2tck

Após processar o conectoma, caso quiser extrair as streamlines que conectam duas regiões diferentes, você pode fazer isso de maneira automática com o connectome2tck. Esse código extrai as streamlines a partir do tractograma baseada na sua atribuição aos nós do parcelamento. E ainda é possível estimar o valor de FA/MD/RD/AD nas fibras resultantes do comando acima.

```
#extrai as streamlines conectando os nós 16 e 105
```

```
$ connectome2tck -nodes 16,105 -exclusive FACT.tck streamlines.csv  
tracts_between_nodes_16_and_105.tck
```

```
#estima o valor de FA das streamlines conectando os nós 16 e 105
```

```
$ tcksample -stat_tck mean tracts_between_nodes_16_and_105.tck dwi_fa.mif fa.csv
```

```
# estima o valor de MD das streamlines conectando os nós 16 e 105
```

```
$ tcksample -stat_tck mean tracts_between_nodes_16_and_105.tck dwi_md.mif md.csv
```

```
# estima o valor de RD das streamlines conectando os nós 16 e 105
```

```
$ tcksample -stat_tck mean tracts_between_nodes_16_and_105.tck dwi_rd.mif rd.csv
```

```
# estima o valor de AD das streamlines conectando os nós 16 e 105
```

```
$ tcksample -stat_tck mean tracts_between_nodes_16_and_105.tck dwi_ad.mif ad.csv
```

Observações

1) extração de picos e visualização – CSD e MT-CSD

extração da amplitude de pico

```
$ sh2peaks ms_wm.mif ms_wm_peaks.mif
```

#output no formato fixel

```
$ fod2fixel ms_wm.mif ms_wm_fixel -afd afd.mif
```

#visualização

```
$ mrview ms_wm.mif -fixel.load ms_wm_peaks.mif -fixel.load ms_wm_fixel/afd.mif
```

2) para melhorar a visualização da tractografia (apenas para visualização)

```
$ tckresample wholebraintractography_FACT.tck -endpoints wholebraintract_FACT_ACT_ends.tck
```

3) para fazer o processamento em mais de um indivíduo – exemplo 1

```
study/001_patient/dwi.mif
```

```
study/002_patient/dwi.mif
```

```
study/003_patient/dwi.mif
```

```
study/004_control/dwi.mif
```

```
study/005_control/dwi.mif
```

```
study/006_control/dwi.mif
```

```
$ for_each study/* : dwidenoise IN/dwi.mif IN/dwi_denoised.mif
```

4) para fazer o processamento em mais de um indivíduo – exemplo 2

```
study/dwi/001_patient.mif
```

```
study/dwi/002_patient.mif
```

```
study/dwi/003_patient.mif
```

```
study/dwi/004_control.mif
```

```
study/dwi/005_control.mif
```

```
study/dwi/006_control.mif
```

```
$ mkdir study/dwi_denoised
```

```
$ for_each study/dwi/* : dwidenoise IN study/dwi_denoised/NAME
```

5) para fazer o processamento em mais de um indivíduo – exemplo 3

```
study/dwi/001_patient.mif
```

```
study/dwi/002_patient.mif
```

```
study/dwi/003_patient.mif
```

```
study/dwi/004_control.mif
```

```
study/dwi/005_control.mif
```

```
study/dwi/006_control.mif
```

```
$ mkdir study/dwi_denoised
```

```
$ cd study/dwi
```

```
$ for_each * : dwidenoise IN ../dwi_denoised/IN
```

6) use -nthreads para rodar os comandos em paralelo

```
$ for_each study/* -nthreads 2 : 2 dados rodando em paralelo
```