Opciones de XNAT CLI para el QC de Freesurfer

Para obtener los archivos necesarios de XNAT, guardar y consultar los resultados hemos hecho extensiones de xnatapic.

Bajar directorios de FS

prepare_fsqc

prepare_fsqc.sh

```
#!/bin/bash

ARGS=( "$@" )
HELP="prepare_fsqc - create the structure for Fresurfer QC with
visualQC"

#local variables
PROJ_ID=""
OUTPUT_DIR=""

#global variables
[ -s "$XNATAPIC_APPS/xnat.conf" ] && . "$XNATAPIC_APPS/xnat.conf"
[ -s "$XNATAPIC_HOME_APPS/xnat.conf" ] && .
"$XNATAPIC_HOME_APPS/xnat.conf"

#parse arguments
for ((n=0; n<${#ARGS[@]}; n++)) ; do
    case "${ARGS[$n]}" in
        --help-short)
            echo "$HELP"
            exit 0
            ;;
        --help)
            echo "$HELP"
            cat <<.EOF
 --help: show this help
 --project <project id> [mandatory]
 --outdir <output directory> [defaults to <project id>_fsresults]
 --list <list of downloaded experiments> [defaults to <project
id>_experiments.list]
.EOF
            exit 0
            ;;
```

```
        --project)
            let n=n+1
            PROJ_ID="${ARGS[$n]}"
            ;;
        --outdir)
            let n=n+1
            OUTPUT_DIR="$(echo "${ARGS[$n]}" | sed 's/[
\n\r\t]\+/+/g')"
            ;;
        --list)
            let n=n+1
            OUTPUT_LIST="$(echo "${ARGS[$n]}" | sed 's/[
\n\r\t]\+/+/g')"
            ;;
        -*)
            echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
            ;;
        *)
            echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
            ;;
    esac
done

#checks
if [ -z "$PROJ_ID" ]; then
    echo "Error: a project ID is required" >&2
    exit 1
fi
#defaults
[ -z "$OUTPUT_DIR" ] && OUTPUT_DIR="${PROJ_ID}_fsresults"
[ -z "$OUTPUT_LIST" ] && OUTPUT_LIST="${PROJ_ID}_experiments.list"
#prepare
[ ! -d $OUTPUT_DIR ] && mkdir $OUTPUT_DIR
if [ -f $OUTPUT_LIST ]; then rm -rf $OUTPUT_LIST; fi
TEMP_ELIST=$(mktemp -t fsqc.XXXXXXXX)
#run
if ! curl -f -X GET -u "$USER:$PASSWORD"
"$HOST/data/projects/$PROJ_ID/experiments?xsiType=xnat:mrSessionData"
2>/dev/null | jq '.' > $TEMP_ELIST ; then
    echo "Error: server reported an error" >&2
    exit 1;
else
    TOTAL=$(cat $TEMP_ELIST | jq '.ResultSet.totalRecords' | sed
's/"//g')
    let TOTAL=TOTAL-1
    for SBJ in $(seq 0 $TOTAL); do
        XEXP=$(cat $TEMP_ELIST | jq ".ResultSet.Result[$SBJ].ID" | sed
's/"//g')
        echo "Processing $XEXP ... ($(($SBJ+1))/$(($TOTAL+1)))"
        FSR=`curl -f -X GET -u "$USER:$PASSWORD"
```

```
"$HOST/data/experiments/$XEXP/files" 2>/dev/null | jq
'.ResultSet.Result[].URI' | grep "\.tar\.gz" | sed 's/"//g'`
        if [ $FSR ]; then
            echo "$XEXP" >> $OUTPUT_LIST
            [ ! -d $OUTPUT_DIR/$XEXP ] && mkdir $OUTPUT_DIR/$XEXP
            TEMP_TAR=$(mktemp -t fsdir.XXXXXXXX.tar.gz)
            curl -f -X GET -u "$USER:$PASSWORD" "$HOST$FSR" -o
$TEMP_TAR 2>/dev/null
            #tar xzf $TEMP_TAR --strip-components=1 -C
$OUTPUT_DIR/$XEXP
            tar xzf $TEMP_TAR --strip-components=1 -C $OUTPUT_DIR/$XEXP
*/mri/{orig,aparc+aseg}.mgz */label/*.aparc.* */surf/*.pial
            rm $TEMP_TAR
        fi
    done
fi
rm $TEMP_ELIST
echo "So long, and thanks for all the fish!"
```

Subir QC de FS

upload_fsqc

upload_fsqc.sh

```
#!/bin/bash

ARGS=( "$@" )
HELP="upload_fsqc - upload visualQC Freesurfer results"

#local variables
INPUT_DIR=""

#global variables
[ -s "$XNATAPIC_APPS/xnat.conf" ] && . "$XNATAPIC_APPS/xnat.conf"
[ -s "$XNATAPIC_HOME_APPS/xnat.conf" ] && .
"$XNATAPIC_HOME_APPS/xnat.conf"

#parse arguments
for ((n=0; n<${#ARGS[@]}; n++)) ; do
    case "${ARGS[$n]}" in
        --help-short)
            echo "$HELP"
            exit 0
            ;;
        --help)
            echo "$HELP"
```

```
            cat <<.EOF
 --help: show this help
 --qcdir <directory with visualQC results> [mandatory]
.EOF
            exit 0
            ;;
        --qcdir)
            let n=n+1
            INPUT_DIR="${ARGS[$n]}"
            ;;
        -*)
            echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
            ;;
        *)
            echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
            ;;
    esac
done
#checks
if [ -z "$INPUT_DIR" ]; then
    echo "Error: input directory is required" >&2
    exit 1
fi
#input files and directories
IMG_DIR="$INPUT_DIR/annot_visualizations"
RATINGS0="$INPUT_DIR/ratings/cortical_contour_rate_freesurfer_orig.mgz_
ratings.all.csv"
END=$(tail -c 1 $RATINGS0)
RATINGS=$(mktemp -t fsqc_ratings.XXXXXXXX)
cp $RATINGS0 $RATINGS
if [ "$END" != "" ]; then echo "" >> $RATINGS; fi
#run
while read LINE; do
    XEXP=$(echo $LINE | awk -F"," '{print $1}')
    QCR=$(echo $LINE | awk -F"," '{print $2}')
    NOTE=$(echo $LINE | awk -F"," '{print $3}' | sed 's/Notes://')
RESULT="{\"ResultSet\":{\"Result\":[{\"rating\":\"$QCR\",\"notes\":\"$N
OTE\"}]}}"
    TEMP_RESULT_FILE=$(mktemp -t fsqc_results.XXXXXXXX)
    echo $RESULT > $TEMP_RESULT_FILE
    curl -f -X PUT -u "$USER:$PASSWORD"
"$HOST/data/experiments/$XEXP/resources/fsqc"
    curl -f -X PUT -u "$USER:$PASSWORD"
"$HOST/data/experiments/$XEXP/resources/fsqc/files/rating.json?overwrit
e=true" -F file="@$TEMP_RESULT_FILE"
    for IMG in $IMG_DIR/$XEXP*.tif; do
        NAME=$(basename $IMG)
        curl -f -X PUT -u "$USER:$PASSWORD"
"$HOST/data/experiments/$XEXP/resources/fsqc/files/$NAME?overwrite=true
" -F file="@$IMG"
```

```
    done
    rm $TEMP_RESULT_FILE
done < $RATINGS
#echo "So long, and thanks for all the fish!"
```

Consultar datos de QC

get_fsqc

get_fsqc.sh

```
#!/bin/bash

ARGS=( "$@" )
HELP="get_fsqc - get the results of Fresurfer QC"

#local variables
PROJ_ID=""
OUTPUT=""

#global variables
[ -s "$XNATAPIC_APPS/xnat.conf" ] && . "$XNATAPIC_APPS/xnat.conf"
[ -s "$XNATAPIC_HOME_APPS/xnat.conf" ] && .
"$XNATAPIC_HOME_APPS/xnat.conf"

#parse arguments
for ((n=0; n<${#ARGS[@]}; n++)) ; do
    case "${ARGS[$n]}" in
        --help-short)
            echo "$HELP"
            exit 0
            ;;
        --help)
            echo "$HELP"
            cat <<.EOF
 --help: show this help
 --project_id <project id> [mandatory]
 --output <output directory> [defaults to <project id>_fsqc.csv]
.EOF
            exit 0
            ;;
        --project_id)
            let n=n+1
            PROJ_ID="${ARGS[$n]}"
            ;;
        --output)
            let n=n+1
```

```
            OUTPUT="$(echo "${ARGS[$n]}" | sed 's/[ \n\r\t]\+/+/g')"
            ;;
        -*)
            echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
            ;;
        *)
            echo "Warning: ignoring option or command ${ARGS[$n]}" >&2
            ;;
    esac
done

#checks
if [ -z "$PROJ_ID" ]; then
    echo "Error: a project ID is required" >&2
    exit 1
fi
#defaults
[ -z "$OUTPUT" ] && OUTPUT="${PROJ_ID}_fsqc.csv"
#prepare
if [ -f $OUTPUT ]; then rm -rf $OUTPUT; fi
TEMP_SLIST=$(mktemp -t xsbjs.XXXXXXXX)
#run
if ! curl -f -X GET -u "$USER:$PASSWORD"
"$HOST/data/projects/$PROJ_ID/subjects" 2>/dev/null | jq '.' >
$TEMP_SLIST ; then
    echo "Error: server reported an error" >&2
    exit 1;
else
    TOTAL=$(cat $TEMP_SLIST | jq '.ResultSet.totalRecords' | sed
's/"//g')
    let TOTAL=TOTAL-1
    for SBJ in $(seq 0 $TOTAL); do
        XSBJ=$(cat $TEMP_SLIST | jq ".ResultSet.Result[$SBJ].ID" | sed
's/"//g')
        XSBJ_LABEL=$(cat $TEMP_SLIST | jq
".ResultSet.Result[$SBJ].label" | sed 's/"//g')
        echo "Processing $XSBJ ... ($(($SBJ+1))/$(($TOTAL+1)))"
        MRI=`curl -f -X GET -u "$USER:$PASSWORD"
"$HOST/data/projects/$PROJ_ID/subjects/$XSBJ/experiments?xsiType=xnat:m
rSessionData" 2>/dev/null | jq '.ResultSet.Result[].ID' | sed 's/"//g'`
        if [ $MRI ]; then
            TEMP_FSQC=$(mktemp -t xfsqc.XXXXXXXX)
            if curl -f -X GET -u "$USER:$PASSWORD"
"$HOST/data/experiments/$MRI/resources/fsqc/files/rating.json"
2>/dev/null | jq '.' > $TEMP_FSQC ; then
                XRAT=$(cat $TEMP_FSQC | jq
'.ResultSet.Result[].rating')
                XNOTES=$(cat $TEMP_FSQC | jq
'.ResultSet.Result[].notes')
                if [ ! -z "$XRAT" ]; then echo "$XSBJ_LABEL, $XRAT,
```

```
$XNOTES" >> $OUTPUT; fi
            fi
            rm $TEMP_FSQC
        fi
    done
fi
rm $TEMP_SLIST
echo "Bye, check results at $OUTPUT"
```