Tutorial to Run FastSurfer Batch Processing

Using Docker and Bash

This tutorial explains how to automate batch processing with **FastSurfer** — a deep learning-based brain segmentation tool — using a Bash script that runs Docker containers for each individual subject.

Prerequisites

- Docker installed and configured with GPU support (`--gpus all`).
- Data organized in BIDS format (e.g., `sub-XXX/anat/sub-XXX_T1w.nii.gz`).
- A valid FreeSurfer license file available locally.
- Read access to the remote BIDS directory (`remote_bids_path`) and write access to the remote output directory (`remote_output`).
- A Linux environment with access to Docker, `rsync`, and appropriate permissions.

Directory Structure

Variable	Description	Example
`remote_bids_path`	Remote path containing BIDS data	`/run/user/.../rawdata`
`tmp_local_copy`	Local temporary folder	`/media/.../Temporarios/bids`
`output_fastsurfer`	Local output folder for FastSurfer	`/media/.../Temporarios/output`
`remote_output`	Remote output path for storing results	`/run/user/.../fastsurfer`
`freesurfer_license`	Path to your FreeSurfer license file	`/home/.../license.txt`

Bash Script

```
#!/bin/bash

export PATH="/home/pisa/.local/bin:$PATH"
```

```
echo "Start"

# === USER CONFIGURATION ===
USE_FS_PROCESSED=false

# === PATH DEFINITIONS ===
remote_bids_path="..."
tmp_local_copy="..."
output_fastsurfer="..."
freesurfer_license="..."
remote_output="..."

# === FUNCTION: CHECK IF SUBJECT IS ALREADY PROCESSED ===
is_processed() {
    local sub=$1
    if [ -d "${remote_output}/sub-${sub}" ]; then
        return 0
    else
        return 1
    fi
}

mkdir -p "$tmp_local_copy"

# === SUBJECT LOOP ===
for dir in "${remote_bids_path}"/sub-*; do
    if [ -d "$dir" ]; then
        sub_id=$(basename "$dir")
        sub="${sub_id#sub-}"

        if is_processed "${sub}"; then
            echo "Subject ${sub} has already been processed.
Skipping..."
            continue
        fi

        anat_dir="${dir}/anat"
        t1_image="${anat_dir}/${sub_id}_T1w.nii.gz"

        if [ ! -f "$t1_image" ]; then
```

```
            echo "T1w image not found for subject ${sub}.
Skipping..."
            continue
        fi

        echo "Copying subject ${sub} to local path..."
        local_sub_dir="${tmp_local_copy}/${sub_id}"
        mkdir -p "${local_sub_dir}/anat"
        cp "$t1_image" "${local_sub_dir}/anat/"

        echo "Processing subject: $sub"
        mkdir -p "${output_fastsurfer}/sub-${sub}"

        docker run --gpus all \
            -v "${tmp_local_copy}:/data:ro" \
            -v "${output_fastsurfer}:/output" \
            -v "$(dirname ${freesurfer_license}):/fs_license:ro" \
            --rm --user $(id -u):$(id -g) \
            deepmi/fastsurfer:latest \
            --fs_license /fs_license/$(basename
${freesurfer_license}) \
            --t1 "/data/${sub_id}/anat/${sub_id}_T1w.nii.gz" \
            --sid "sub-${sub}" \
            --sd /output \
            --parallel --3T

        echo "Copying result to remote fastsurfer directory..."
        rsync -r "${output_fastsurfer}/sub-${sub}"
"${remote_output}/"

        echo "Cleaning up..."
        sudo rm -rf "${output_fastsurfer}/sub-${sub}"
"${local_sub_dir}"
    fi
done
```

Step-by-step Usage

1. **Update the paths**:
    - Replace the values of `remote_bids_path`, `tmp_local_copy`, `output_fastsurfer`, `remote_output`, and `freesurfer_license` with

the actual paths from your system.

2. **Set permissions**:
    - Ensure you have read access to the BIDS dataset in `remote_bids_path`.
    - Ensure you can write to `tmp_local_copy`, `output_fastsurfer`, and `remote_output`.

3. **Save the script**:
    - Copy the code above into a file, e.g., `run_fastsurfer.sh`.

4. **Execute the script**:
    - Run it: `bash run_fastsurfer.sh`

Explanation of Key Script Components

- `for dir in "${remote_bids_path}"/sub-*`: loops through all subject folders.
- `sub="${sub_id#sub-}"`: extracts just the subject ID number.
- `docker run --gpus all ...`: runs the FastSurfer container with GPU support.
- `sudo rm -rf`: removes temporary local copies to free disk space.

Additional Tips

- To speed up processing, consider using GNU Parallel or running multiple background jobs (`&`).
- Ensure your local disk has enough free space for temporary data.
- Make sure your Docker setup supports GPU (verify using `nvidia-smi`).
- For more options and documentation, visit FastSurfer official docs.