Implementation and Evaluation of a Semantic Workflow for Neuroimage Processing

By

Brandon Keibler

A Thesis

Presented to the Department of Medical Informatics & Clinical Epidemiology
and the Oregon Health and Science University

School of Medicine

in partial fulfillment of
the requirements for the degree of

Master of Science

June 2017

School of Medicine

Oregon Health & Science University

Certificate of Approval

This is to certify that the Master's Thesis of

Brandon S. Keibler

"Implementation and Evaluation of a Semantic Workflow for Neuroimage Processing"

Has been approved

[redacted]

[redacted]

[redacted]

Table of Contents

Introduction

Neuroimaging

Analysis of the human brain is most commonly performed using Magnetic Resonance Imaging (MRI) due to it being relatively fast and non-invasive.[1] MRI technology leverages the fact that the human body is comprised mostly of water, using a strong magnetic field to align the hydrogen protons of the H_2O molecules within the subject. A radio signal is used to excite those protons and is then turned off. As the protons relax, they release radio waves that are detected by the sensor, producing an image. The amount of time required for the atoms to return to equilibrium will be different for different types of body tissue, so this is used to distinguish separate anatomical features.

This technology has been used heavily in medicine for the detection of brain injury[2] and tumors.[3,4,5] It is also used extensively in the field of neurology to study early brain development[6,7] and the characteristics of mental disorders.[8] The combination of diffusion MRI, which provides a structural image of the brain, and functional MRI, which is used to visualize functional activity of the brain, allows for in-depth analysis of human brain function. Brain activity can be measured in response to different stimuli and compared between individuals suffering from various neurological conditions.[9] It can also be used to develop connectivity maps to study the interrelatedness of different brain regions.[10]

The Human Connectome Project[11] (HCP) is a particularly high-profile example of current research aimed at improving our understanding of the human brain. This large collaborative undertaking maps the neural pathways underlying brain function by analyzing correlations in the activity patterns of different brain regions. This provides insight into how the various regions of the brain work together to handle tasks, giving us a better understanding of how the complex and interrelated structure of the brain functions. Such insight has been used as a springboard to study neurological phenomena, such as how connectivity patterns differ throughout the population,[12] and to improve our understanding of white matter anatomical connectivity to better inform surgical protocols.[13] Already, the HCP has established a solid foundation for studying functional brain anatomy and variation.

Pipelines

The raw images from MRI machines require a significant amount of processing just to get them to the point where they can begin to be analyzed. Minimally, there are multiple sources of image distortions inherent in MR imaging which must be corrected for. For example, in order to analyze a patient's brain from an MRI scan, it must first be isolated from the image, and the separate anatomical regions must be defined for that individual's brain, as each brain has its own slight variations in size and shape.

Data processing pipelines help to automate this intricate process, and allow analysts to orchestrate the many steps required. In an effort to standardize the processing procedure for the Human Connectome Project, a minimal processing pipeline[14] (Figure 1) has been developed by researchers at Washington University to process structural and functional brain MRI data and prepare it for analysis. The use of this pipeline has been employed by many researchers, including studies investigating the neurology of drug dependence[15], possible treatments for Parkinson's Disease[16], and neurologic basis of human personalities.[17]

The availability of the minimal processing pipelines has gone a long way toward making human brain analysis more manageable and reproducible.[18,19,20] As with other fields of science, reproducibility is of an ongoing concern.[21,22] Some estimates suggest that up to 50% of all pre-clinical research is not reproducible, representing approximately $28.2 billion in wasted government spending annually in the U.S.[23] Understandably, scientific approaches that are able to enhance reproducibility by standardizing the handling of data and removing potential sources of human error are highly desirable.

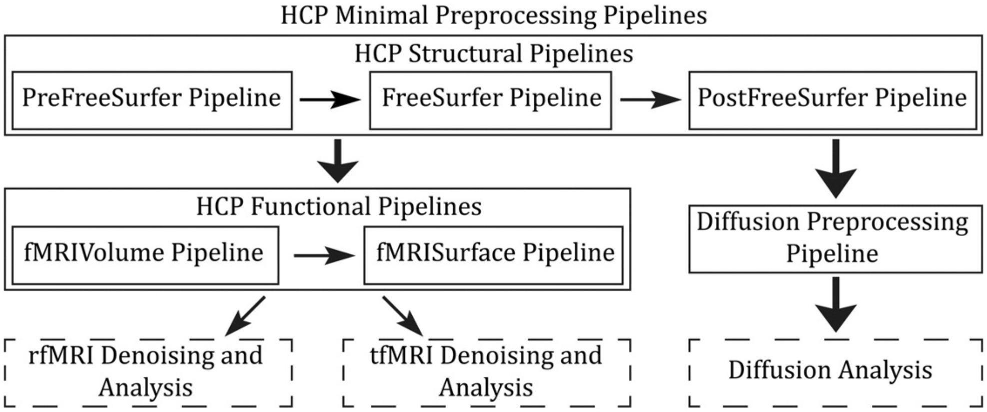


Figure 1: High level overview of the Human Connectome Project Minimal Processing Pipeline. The HCP Structural Pipelines prepare a subject's structural brain map and registers it to a reference. The HCP Functional Pipelines preform many analogous operations on a time series of functional images that indicate changes in the subject's brain activity. This functional data is then registered to the individual's structural brain map. The Diffusion Preprocessing Pipeline does similar things as the HCP Functional Pipelines, but is specialized to prepare the functional data for diffusion analysis.

The use of these data processing pipelines vastly reduces the amount of time that is required to carry out this processing, but running them effectively still demands an intimate knowledge of the algorithms being employed. While it is relatively easy to push one's data through an established pipeline, it is up to the user to ensure that it is being handled in a biologically relevant way. The individual algorithms that make up a pipeline often depend on user-input parameters to guide their execution. Without knowing what these parameters are, and what values are appropriate for the data being processed, it can be easy to end up with data that is sub-optimal, or even unusable. For example, in the minimal processing pipeline, the user is required to input the subject's brain size, which is used when aligning to the reference brain image. If this value is incorrect, or entered using the wrong units, the algorithm will evaluate incorrect transformations necessary to align the images.

Workflows

While a pipeline can be as simple as a few computational operations run in succession from a command line, a workflow simplifies things by encapsulating processing steps into individual components or modules, hiding unnecessary detail from the user, and clearly defining all inputs and outputs. Additionally, they usually provide a GUI interface to make operation more convenient. Workflow systems such as Galaxy[24,25,26] and LONI Pipeline[27] can be used to alleviate many of the difficulties associated with data processing by allowing for parallelization and coordinating the operation of processing steps across multiple environments. They also provide the user with visual representations of the flow of information throughout the various processing steps. Workflows are generally understood to be beneficial and are widespread in medical science. Examples include detecting microRNA[28] and epigenetic DNA modifications[29], to mass spectrometry[30] and drug discovery.[31]

Using a workflow system to operate one's pipeline often introduces features to help provide some user guidance. When setting up a job on Galaxy, the system will enforce that the correct file types are being supplied by the user by only allowing them to select from files that have the correct format. Still, there is no way to guarantee that the information in those files are actually correct and properly formatted. For example, in a .BED file, which is used in genomics to describe the location of features in a genome, each line contains 3 minimally necessary fields, followed by several optional fields that can contain additional information. If an algorithm is being used that depends on that optional information, the user needs to verify that it is present in the input file. Additionally, workflows often require multiple input files that are related, or contain interdependencies between each other. A workflow may require two inputs that are expected to both contain information about an individual that were obtained on the same day, but if the user accidentally specifies inputs containing information from different time points, the workflow will still run, but yield invalid results. If something like this is not caught by the operator, it can have harmful downstream ramifications, possibly causing investigators to draw erroneous conclusions. It is still easy to select non-compatible inputs or specify sub-optimal parameter values, which can lead to issues in the accuracy and reproducibility of results. These workflow systems lack an integrated user guidance system to convey the expectations of a workflow, make suggestions, and help point users in the right direction.

Semantic Workflows

Semantic workflows are a unique class of workflow systems that allow for the introduction of semantic constraints. These constraints can be used to help guide users through a workflow run, thereby helping to enhance both analytical accuracy and reproducibility. Semantic constraints can function as an extra layer of input checking that verifies that the datasets and parameters specified by the user conform to all the expectations of the workflow. These semantic constraints can also be defined to help guide the user by only allowing valid inputs which are compatible with each other, and can provide messages to the user explaining what is expected by the workflow. Semantic constraints can also be defined to suggest input parameters to the user based on their other specified inputs, reducing the amount of expertise required, and easing the burden of executing the workflow.

A semantic workflow implemented to collect and merge annotation information for genetic variants from various sources is one example of how semantics have been used effectively in the field of clinical omics.[32] This workflow, implemented in the WINGS semantic workflow system,[33] takes a list of genetic variants detected in a patient, and compares it to the libraries of genetic variants annotated in the COSMIC and dbSNP databases, as well as any in-house curation of DNA variants, to assemble information about the genetic situation of the patient. For this process, it is assumed, and essential, that the different lists of genetic variants were produced using the same version of the human genome as a reference. Otherwise, the data would be incompatible, though there is no immediately obvious way to see if that were not so. In this case, semantic constraints were used to ensure this compatibility behind the scenes based on the metadata of the input files, and used to verify that everything was in order prior to executing the workflow. In this way, semantics are used to enhance the validity and reproducibility of the analysis.

Semantic workflows have been implemented for many applications, including biomedical imaging,[34] but until now there have not been any implemented that specialize in neuroimaging. We hypothesize that the operation of a semantic workflow for medical image processing will enhance its usability. To test this hypothesis, we developed a semantic implementation of PreFreeSurfer, an established neuroimage processing pipeline, using the WINGS semantic workflow system. The usability of our semantic workflow was then evaluated via user testing and feedback. We expect that this work will help to establish the effectiveness of semantic workflows in enhancing reproducibility and analytical accuracy in medical image processing and begin to make them available to researchers.

Methods

Use Case: The PreFreeSurfer Pipeline

To investigate the potential benefits of semantic workflows in neuroimaging processing, we chose to focus on the PreFreeSurfer[12] pipeline. (Figure 2) PreFreeSurfer is the first step in a series of minimal processing pipelines developed at Washington University as part of the Human Connectome Project. The original code for the PreFreeSurfer pipeline (designed to be run from command line) is available on the Washington University GitHub page: https://github.com/Washington-University/Pipelines/tree/master/PreFreeSurfer

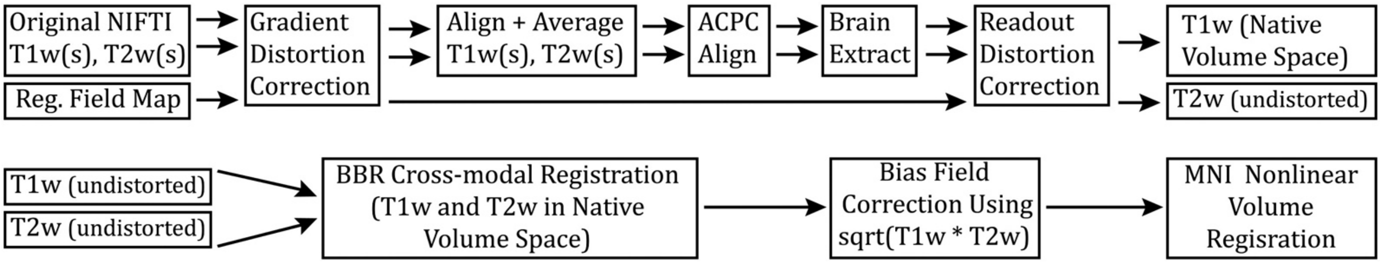
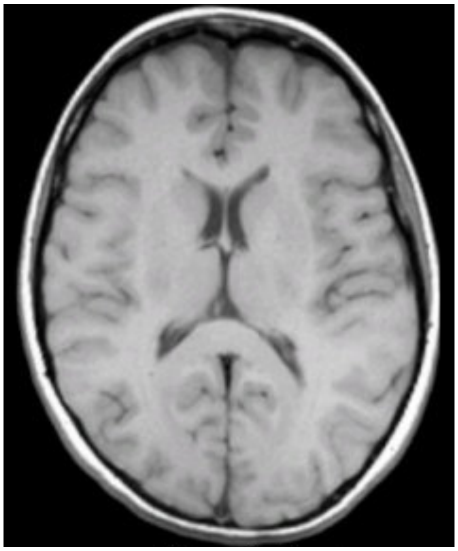
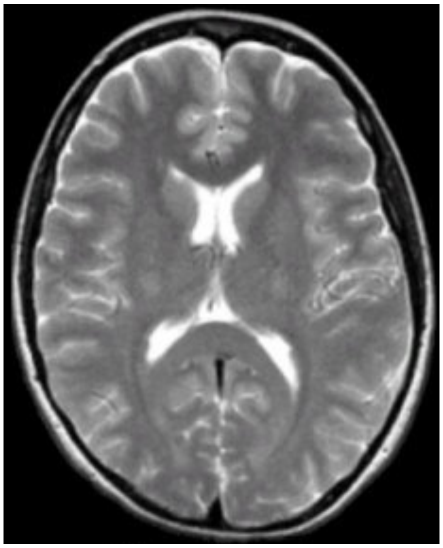

Original NIFTI T1w(s), T2w(s)
Reg. Field Map
Gradient Distortion Correction
Align + Average T1w(s), T2w(s)
ACPC Align
Brain Extract
Readout Distortion Correction
T1w (Native Volume Space)
T2w (undistorted)
T1w (undistorted)
T2w (undistorted)
BBR Cross-modal Registration (T1w and T2w in Native Volume Space)
Bias Field Correction Using sqrt(T1w * T2w)
MNI Nonlinear Volume Regisration

Figure 2: Simplified map of the PreFreeSurfer pipeline. The "native volume space T1w" and "undistorted T2w" outputs at the end of the first row are the "undistorted T1w" and "undistorted T2w" inputs at the start of the second row.

The primary goal of PreFreeSurfer is to produce an undistorted structural volume image of an individual's brain in their native space, which represents the true brain size and shape for that individual, and register it to a reference brain image, so that each point of the individual's brain is matched to the corresponding point in the reference. The pipeline takes one or more replicates of the raw structural images, both T1-weighted and T2-weighted (description in Table 1), from MRI scanning, then aligns and averages them to define the individuals native brain space. The image is corrected for gradient and readout distortion, as well as any bias field inhomogeneity. The pipeline also masks out the skull and other tissue surrounding the brain from the image, so that only the brain remains, and finally registers the individual's native brain image to the reference brain to allow for comparisons to be made across individuals.

Table 1: Comparison of T1 and T2 weighted images.

Scan Parameters	**T1-weighted**	**T2-weighted**
Repetition Time	Short (500ms)	Long (4000ms)
Echo Time	Short (14ms)	Long (90ms)
Example		
Description	Intensity determined by time required for 63% of longitudinal magnetization to decay.	Intensity determined by time required for 63% of transverse magnetization to decay.
Cerebrospinal fluid (CSF)	Dark	Bright
White Matter	Light	Dark Grey
Cortex	Grey	Light Grey
Bone Marrow (fat)	Bright	Light

The WINGS Semantic Workflow System

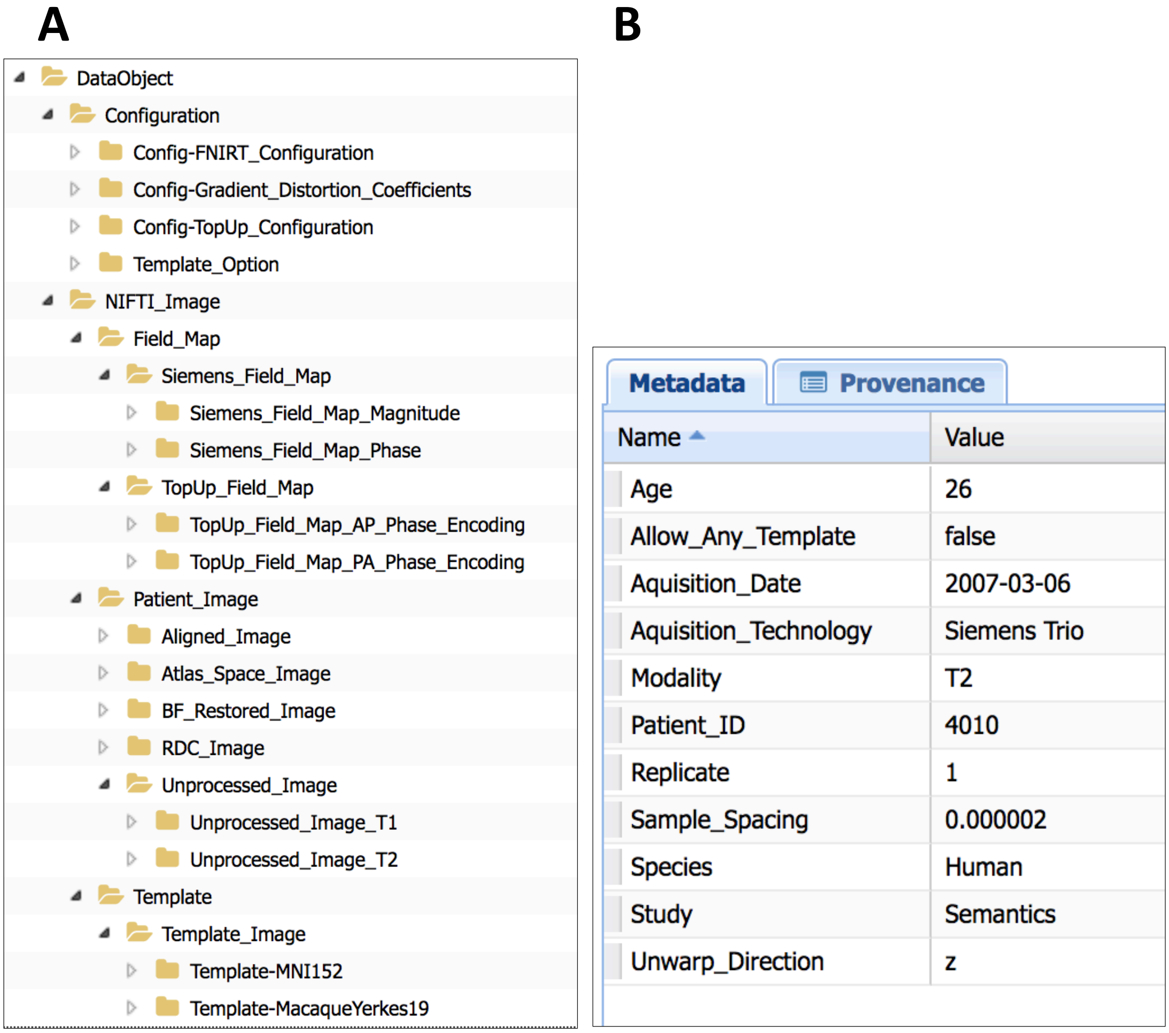
The WINGS Platform

WINGS[30] is a semantic workflow system that allows us to implement the desired semantic constraints in our workflow. It is freely available online at http://www.wings-workflows.org/. Similarly to other workflow management platforms, In WINGS, distinct processing steps are encapsulated into "components", with predefined inputs, outputs, and parameters. These components are then strung together into a comprehensive workflow.

What sets WINGS apart from other platforms is that each workflow run is checked for consistency and coherence, ensuring that only semantically validated runs are performed. Wings enforces these semantic constraints in 2 primary ways: the integration of components with a hierarchy of user-defined data types, and by enforcing a set of user-defined semantic rules that are integrated into each component.

Workflow-Specific Data Types

First, the workflow developer defines data types for the inputs and outputs that will be used by the workflow. (Figure 3A) In our use case, PreFreeSurfer primarily operates on NIFTI images (3-dimensional representations of the volumetric space scanned by an MRI machine) of the brain. However, these images can be semantically separated by their intended purpose in the workflow. In addition to the structural images of a patient's brain, PreFreeSurfer requires supporting patient brain images called fieldmaps, which are used to correct for readout distortion in the structural images. Each patient in a study is also registered to a reference brain template image to allow for comparisons across individuals in downstream analyses. The patient structural images themselves can also be separated into multiple data types to distinguish between raw unprocessed images, and the processed images produced by the workflow and its intermediate steps. When defining the inputs and outputs for a component, the specific data type is specified to ensure that only files that are appropriate for each input can be selected.



- DataObject
  - Configuration
    - Config-FNIRT_Configuration
    - Config-Gradient_Distortion_Coefficients
    - Config-TopUp_Configuration
    - Template_Option
  - NIFTI_Image
    - Field_Map
      - Siemens_Field_Map
        - Siemens_Field_Map_Magnitude
        - Siemens_Field_Map_Phase
      - TopUp_Field_Map
        - TopUp_Field_Map_AP_Phase_Encoding
        - TopUp_Field_Map_PA_Phase_Encoding
    - Patient_Image
      - Aligned_Image
      - Atlas_Space_Image
      - BF_Restored_Image
      - RDC_Image
      - Unprocessed_Image
        - Unprocessed_Image_T1
        - Unprocessed_Image_T2
    - Template
      - Template_Image
        - Template-MNI152
        - Template-MacaqueYerkes19

Name	Value
Age	26
Allow_Any_Template	false
Aquisition_Date	2007-03-06
Aquisition_Technology	Siemens Trio
Modality	T2
Patient_ID	4010
Replicate	1
Sample_Spacing	0.000002
Species	Human
Study	Semantics
Unwarp_Direction	z

Figure 3: A) Data type hierarchy developed for use with our implementation of PreFreeSurfer. B) Metadata defined for the unprocessed patient image data type.

The other advantage to defining custom file types is that the developer can designate metadata fields for each type of file, indicating what additional information is necessary to run the workflow correctly. Having these metadata values assigned to the data files provides users with a way to get a more detailed understanding of their data at a glance. Additionally, a semantic workflow can leverage this information to carry out further semantic reasoning via semantic rules.

Semantic Rules

More complex semantics are defined by the developer via semantic rules (Appendix A.2) encoded into each component. These rules can use the metadata assigned to the input data to restrict the running of a workflow and trigger error messages to the user describing semantic constraints that are being violated. Examples from our use case include a rule that verifies that the patient ID for both the T1 and T2 patient images are identical. Another example is a rule that verifies that the selected template image is appropriate for the patient based on their age and species.

Semantic rules can also be used to completely automate the selection of some inputs and parameters, reducing the number of inputs that must be specified by the workflow user, and completely removing the possibility of incompatible options being selected. PreFreeSurfer accepts 9 different versions of the reference template, since images of different modality (T1 vs T2), resolution, and whether the brain has been isolated from the surrounding skull and head tissue in the image, are used in various processing steps. Rather than having the user select each template file, we created rules to have the workflow automatically choose them based on the template the user chooses use. Given the selected template, the metadata assigned to the available template images describing their resolution, modality, and the imaging information they contain, are used to determine what would be most appropriate for each of the expected template inputs.

Additionally, many of the parameters expected by the PreFreeSurfer pipeline are inherent properties of the patient images (settings such as the dwell time and unwarp direction used during image acquisition). By assigning these values to the patient images as metadata, the workflow can obtain these values from the input files automatically using semantic rules.

Abstract Components

The usefulness of these semantic rules can also be expanded to create “abstract components”. These are components with multiple versions which the workflow should use in different circumstances, such as which readout distortion correction algorithm should be used when running PreFreeSurfer. Rules encoded into the multiple component versions check the metadata on the selected inputs and invalidate any versions of the component that are not viable. The workflow user is only offered the version(s) that is appropriate given their data.

This functionality can also be accomplished by leveraging the integration of the defined data types mentioned earlier. If different versions of a component accept different types of input files, as is the case with the various readout distortion correction algorithms, semantic rules can be used to define what would be appropriate inputs for each version. When WINGS is generating a valid workflow run, the appropriate component version(s) will be determined by what input files are present in WINGS. This allows us to automate the selection of processing steps in workflows with multiple possible processing pathways.

Another key use of semantic rules is the propagation of metadata onto new data objects. Rules can be defined to automatically pass appropriate metadata values from inputs onto the outputs created by the workflow, as well as to assign new metadata values to document the exact processing steps each file has undergone, ensuring data provenance.

Workflow Implementation

Our semantic PreFreeSurfer workflow (Appendix A.1) was implemented in the WINGS semantic workflow system. The data processing steps were organized into 4 components (Figure 4) based on what situational choices must be made by the workflow to execute PreFreeSurfer correctly. This design combines processing steps that are always performed together, simplifying the workflow, while still maintaining as much modularity as possible to facilitate any future updates to the PreFreeSurfer pipeline.

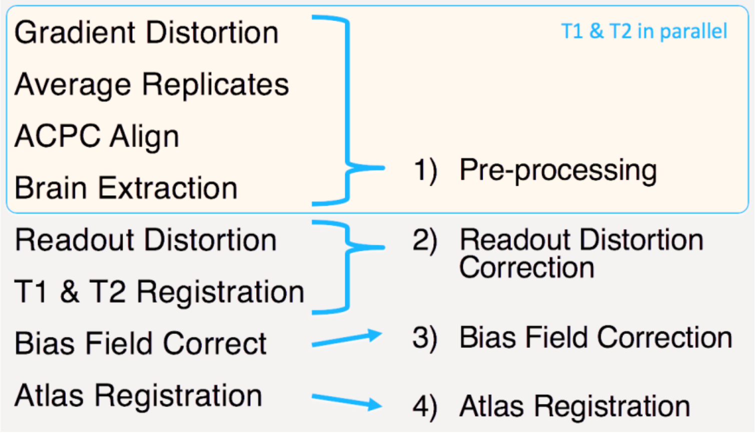


Figure 4: Organization of PreFreeSurfer processing steps into WINGS components.

The first component Incorporates the first several steps that prepare the images for subsequent processing, as well as gradient distortion correction. The only decision here is whether or not to perform the correction. This step is performed for both the patient T1 and T2 inputs in parallel, assuming both are present.

The second component handles both the readout distortion correction, as well as the registration of the T1 and T2 images. The reason for combining these two steps is that computational shortcuts can be taken to significantly reduce the time required to process the data by simultaneously correcting for readout distortion while registering the T2 image to the T1 image. At this step, the type of readout distortion correction algorithm is determined based on what fieldmap files are available for the patient.

The third component handles the bias field correction, assuming that both the T1 and T2 images are available. If there is no T2 image, this step is skipped. There are alternate bias field correction algorithms that can be performed[35] but they have not been implemented into our workflow at this time.

Because the exact size and shape of each individual's brain is unique, the brain images must be warped to match a universal template, or atlas, in a processes called 'registration'. This ensures that the various functional regions of the brain are located in the same place in each individual's brain image. The fourth component in our workflow performs this atlas registration of the processed patient images to the reference template to allow for comparisons to be made across individuals.

The various semantic constraints integrated into the workflow, and how they were implemented, are described in Table 2.

Table 2: Semantic constraints and reasoning implemented in our semantic version of PreFreeSurfer.

Constraint	**Implementation**
All structural patient images and fieldmap files should belong to the same individual.	Semantic rules check the metadata listing the Patient ID, as well as the Study, Patient Age, and Acquisition Date of the patient's unprocessed T1 and T2 images. If any of these do not match, an error message informing the user of the inconsistency is produced. The same metadata should also match between the patient images and fieldmap image files, though this is handled differently due to the presence of specific fieldmap files being used to determine the correct readout distortion algorithm (see below). Similarly though, semantic rules compare the metadata values for the patient images to the metadata on all available fieldmap files, and only valid matches are selected.
The template should be appropriate given the patient age and species.	The metadata describing the patient's species and age (on the unprocessed images) are read, and rules defining which template(s) are appropriate for that individual are used to restrict which templates may be used. If an incompatible selection is made, the user is presented with an error message. We also wanted to allow for some flexibility here, so that researchers could use non-standard templates if desired. Comparing human and primate brains, for example, would necessitate such an option, as at least some of the subjects would have to be registered to a template of a different species. To accomplish this, we included a metadata value for patient images that could be used as a flag to indicate that the use of any template should be allowed.

Constraint	Implementation
Template images are automatically selected.	Instead of requiring the user to individually select all 9 template files expected by PreFreeSurfer, we wanted the user to select from a list of available templates, and let the workflow rules handle the selection of each specific template file. As pre-defined option sets are not, at the time of this publication, available in WINGS, we created a dummy 'Template" data type, where each selectable input just contained the name of an available template. Semantic rules then used the chosen template name, along with the Resolution, Modality, and Image Type metadata on each template file to determine which template file to use for each input.
The necessity of gradient distortion correction, and what correction coefficients file to use is determined by the technology used during acquisition.	Depending on experimental conditions, such as what type of MRI machine was used to acquire the patient's brain images, gradient distortion correction may not be as necessary as it is in other situations. In many labs, this step is skipped to reduce processing time when the downstream analyses will not be significantly affected by the distortion. Our workflow includes semantic rules check the Acquisition Technology of the selected patient images. If they were acquired on a machine that produces significant gradient distortion, such as the customized Siemens Skyra used in the Human Connectome Project, the appropriate gradient distortion coefficients file is selected, and the correction is performed. If the correction is not determined to be necessary, the user will still be able to select from available coefficient files, or choose to skip the correction completely. Additional rules also verify that the selected coefficient file is for the MRI machine that the patient images were acquired from. If this is not the case, an error message is produced.
The readout distortion correction method is determined by available fieldmap files.	There are several readout distortion correction methods that can be used, and which one can be performed depends on what type of supporting fieldmap images were acquired during data acquisition. We implemented an algorithm to perform Siemens Readout Distortion Correction, which relies on a phase encoding fieldmap image and a magnitude encoding fieldmap image of the patient, and an algorithm for TopUp Readout Distortion Correction, which relies on a positive phase encoding fieldmap image and a negative phase encoding fieldmap image. We also wanted to have the option to skip readout distortion correction in the event that none of the necessary fieldmap files were available, or were otherwise unusable, though this would certainly not be desirable. To handle this situational processing, we implemented an abstract component, which is made up of several versions of the same component. The inputs required for each version are different, so WINGS is able to determine which one(s) are possible based on which fieldmap files are present. Semantic rules detect which patient's data is being processed and search for any fieldmap files belonging to that individual. Component versions that require inputs that are not present are invalidated so that the user is only given the option to run the correct algorithm. If n fieldmap files are present, the user is only given the option to register the T1 and T2 images without performing readout distortion correction. This method has the added benefit of automatically selecting the correct fieldmap files for a given patient, reducing the workload of the user.

Constraint	Implementation
Readout distortion correction parameters are automatically determined.	Similarly, all the parameters used by each readout distortion correction algorithm are inherent to the patient images, and documented as metadata. Additional semantic rules read this information from the unprocessed patient brain images and assign the correct values for these parameters automatically. This ensures that the correct parameter values are being used and saves the workflow user from having to look them up.
The bias field correction method is determined by presence of T1 and T2 images.	The bias field correction algorithm implemented in PreFreeSurfer combines information in the T1 and T2 images to calculate the bias field signal and correct for it. In the event that no T2 image is available, an alternate bias field correction algorithm could be employed, though currently the only other algorithm is to skip the bias field correction. This situational handling requires the use of another abstract component, with one version being used if both images are present, and the other being used if they are not. When running a workflow in WINGS, it does not allow you to leave an expected input blank, or enter a null value. Because of this, there is no way to not select an input for the unprocessed T2 image(s). To get around this, we created a dummy data type contained within the Unprocessed T2 data type that allows us to select “no input” as the T2 image. Because WINGS recognizes this “no input” as a kind of Unprocessed T2 image, it is selectable, though it does actually contain any data. The bias field correction component specifically looks for whether the Unprocessed T2 input is dummy data, and invalidates one of the two component versions accordingly. This allows the workflow to select the correct processing method automatically depending on the selected input.

User Testing

To test the usability of our workflow, we recruited 5 experienced users to set up workflow runs for provided sample datasets using our workflow. We intentionally restricted potential test users to individuals with experience working with neuroimaging data processing pipelines, and specifically experience using the PreFreeSurfer pipeline, so as to minimize the effect of PreFreeSurfer’s inherent complexities on the users’ perception of the semantic workflow. We also determined that more experienced users would be more capable of providing useful feedback regarding the workflow, and be more likely to pick up on potential problems with our implementation that may have been overlooked.

Our intention was to evaluate the benefits provided by the use of semantics in the workflow, rather than evaluating the usability of the WINGS platform itself (thought the two are not completely independent, since the way WINGS is designed forces one to incorporate at least some semantics into their workflow). To this end, we also created non-semantic versions of our PreFreeSurfer workflow in WINGS for comparison. The non-semantic workflows were identical to the semantic workflow, except that the semantic rules were removed, and they did not make use of abstract components, requiring us to make a total of 12 separate non-semantic workflows to handle each of the possible situations that a user might encounter (Table 3).

Table 3: List of the 12 non-semantic workflows required to represent all possible data processing scenarios a test user might encounter. In more traditional environments, such as running PreFreeSurfer via command line, options such as what readout distortion correction method to use, and whether or not to use gradient distortion correction, can be specified by entering an appropriate value for certain parameters. In WINGS it is easier to to have users choose between these options by selecting between different versions of a workflow. It should also be noted that PreFreeSurfer only needs to select the correct template files to handle different species or age groups. Subsequent steps in the minimal processing pipelines would require additional versions of the pipeline to handle these situations.

Number	Available Patient Images	Readout Distortion Correction Method	Gradient Distortion Correction Used
1	T1 and T2	None	Yes
2	T1 and T2	None	No
3	T1 and T2	Siemens	Yes
4	T1 and T2	Siemens	No
5	T1 and T2	TopUp	Yes
6	T1 and T2	TopUp	No
7	T1 only	None	Yes
8	T1 only	None	No
9	T1 only	Siemens	Yes
10	T1 only	Siemens	No
11	T1 only	TopUp	Yes
12	T1 only	TopUp	No

The testing participants were provided with the login credentials for a test account containing all necessary example datasets and access to both the semantic and non-semantic workflows. They also received testing instructions (Appendix A.3) to guide them through the testing. Participants were asked to run through 4 separate testing scenarios using both the semantic and non-semantic versions of the pipeline. Each of the test scenarios were designed to each exhibit advantages of different semantic constraints integrated into the workflow, and were as follows:

1) **Standard**: A simple dataset such as would have been used in the Human Connectome Project. All necessary data was present and without issue. This also served to introduce the users to the WINGS platform, and illustrate how it is able to use input file metadata to make recommendations and restrict the options of other selectable inputs to only those that are appropriate.
2) **Alternate**: A dataset acquired using different imaging technology than the “standard” dataset, requiring the use of a different readout distortion correction algorithm. Gradient distortion

correction is also not necessary for this dataset, due to the very minimal gradient distortion introduced by the machine the images were obtained from.

3) **Non-human Primate**: This dataset consists of images obtained from a non-human primate, rather than a human subject. This requires that use of an alternate set of brain image templates that are appropriate for the subject.
4) **Mislabeled\Missing:** A dataset where some of the input data has been mislabeled, causing it to not correspond to the correct individual. When users attempt to run this seemingly straightforward dataset, the inconsistencies in the input files will be detected and a warning will be provided to the user, allowing them to catch the mistake and exclude the incorrect data from the processing.

User Feedback

After completing the testing process, the participants were asked to fill out a short survey (Appendix A.4) about their experience. The survey was comprised of the 10 standard questions used in the System Usability Scale[36], followed by several free-response questions.

The System Usability Scale (SUS) questions ask the user to indicate how strongly they agree with a statement about their experience, ranging from "strongly disagree" to "strongly agree", using a 5-point scale. It alternates between positive and negative statements, encouraging users to score their experience without being biased by the wording of the questions. The responses to these questions can then be used to calculate an overall usability score ranging from 0 to 100. Each item's score contribution ranges from 0 to 4. For positively worded items, the score contribution is the scale position minus 1. For negatively worded items, the contribution is 5 minus the scale position. The sum each item's contribution is then multiplied by 2.5 to obtain the overall system usability score.

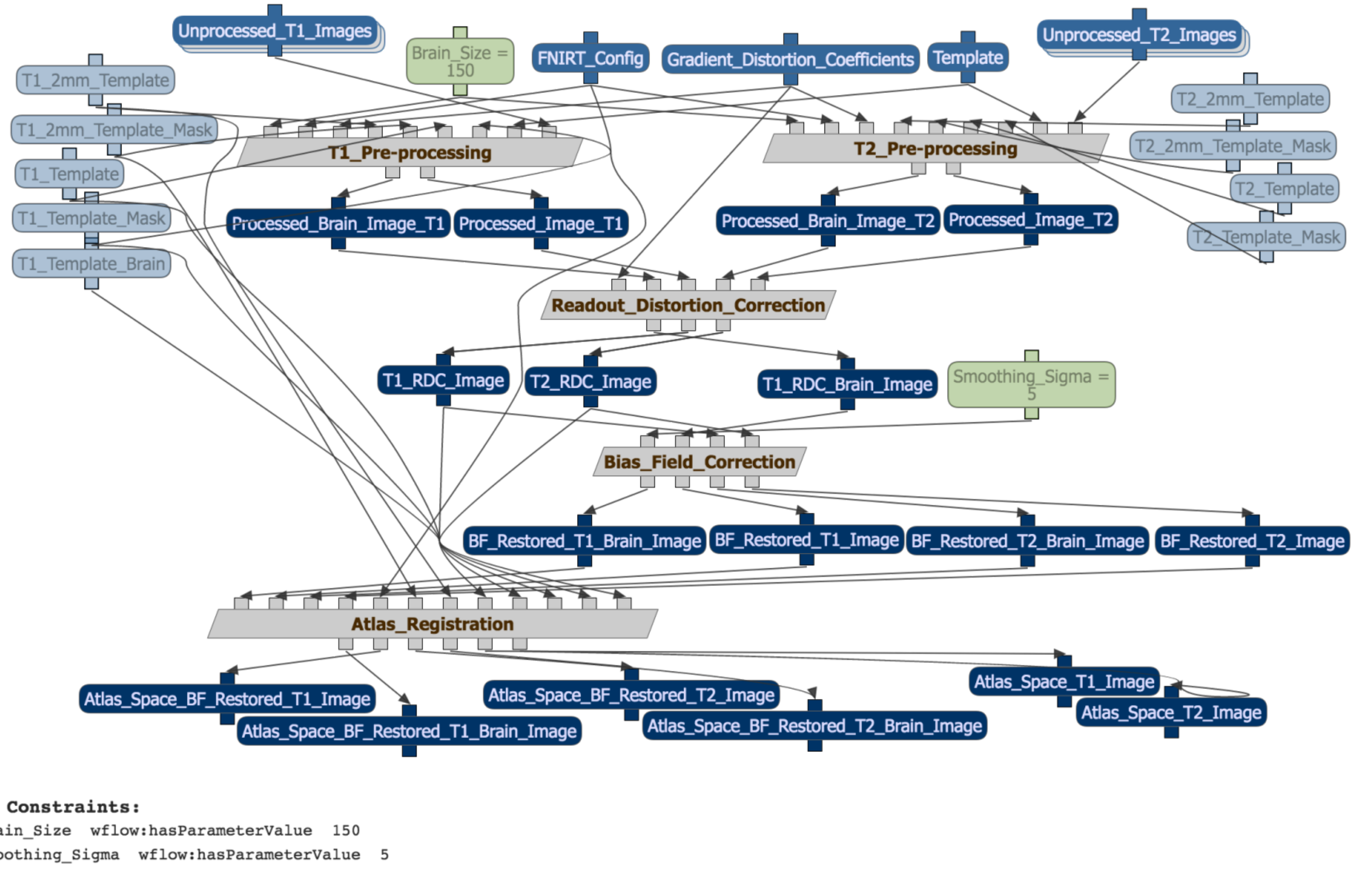
We chose to employ the System Usability Scale because it has been shown to be reliable, even at smaller sample sizes.[37] Additional benefits are that it measures both learnability and usability, and that participants only need short exposure to system in order to provide accurate feedback.

The free-response questions were included to allow participants an opportunity to leave any additional feedback they feel would be appropriate, including the opportunity to raise any concerns they may have about the workflow.

Results

PreFreeSurfer Semantic Workflow

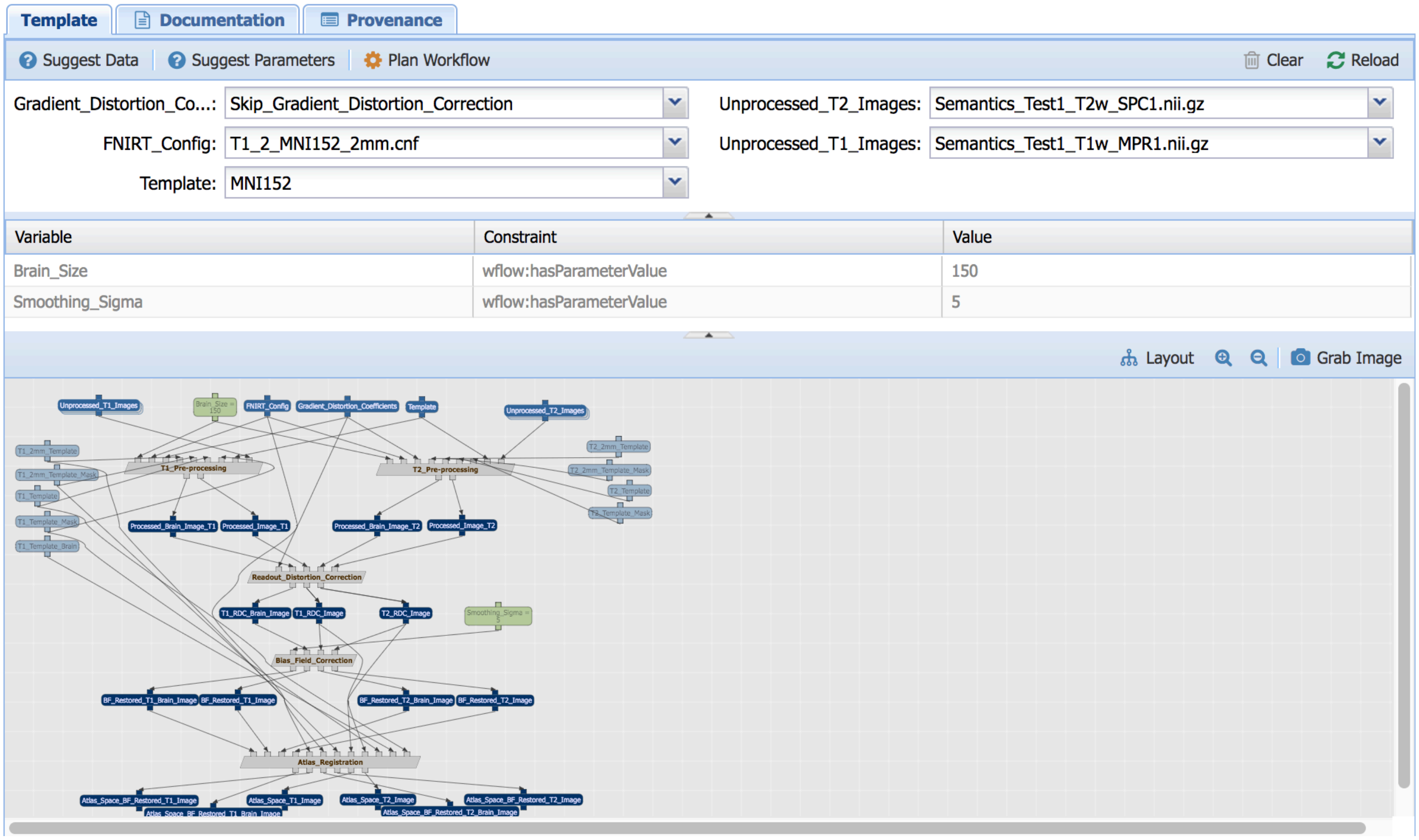
Our implementation of PreFreeSurfer (Figure 5) in WINGS includes all of the functionality of the base PreFreeSurfer pipeline, while benefiting from the semantic validation and user guidance made possible by a semantic workflow. Semantic rules and constraints have been defined to verify the compatibility of several inputs to protect from user error. Selectable inputs are restricted to only those that are appropriate for the intended use of the workflow, and in many cases the inputs and parameters are automatically chosen to facilitate the setting up of workflow runs. Rules were also implemented to propagate metadata onto the workflow outputs, and new metadata variables are created to describe the algorithms that were used, preserving information about the data and how it was processed. Finally, the ability to navigate various situational processing pathways alleviates the need to develop and maintain the multiple workflows normally required to process data under varying circumstances.



```
All Constraints:
  Brain_Size  wflow:hasParameterValue  150
  Smoothing_Sigma  wflow:hasParameterValue  5
```

Figure 5: Map of the semantic PreFreeSurfer workflow in WINGS. The green and blue boxes that are faded-out are automatically assigned by WINGS, and are therefore not shown to the user as selectable inputs. The 5 light-blue boxes at the top are the inputs that must be chosen by the user. Additional inputs for the Readout Distortion Correction component are dependent on which correction method is used, and are all automatically selected, so are not shown in the workflow map.

The command line version of PreFreeSurfer includes 27 inputs and parameters (not including the path to where you want it to save the output files), 13 of which may or may not be used depending on what readout distortion correction method is being employed. In comparison, our semantic PreFreeSurfer workflow in WINGS only requires the selection of 5 inputs. (Figure 6) Furthermore, after selecting just the unprocessed T1 and T2 patient images, a user can take advantage of WINGS's data suggestion feature to assist with selecting the other inputs. All the other inputs and parameters required by PreFreeSurfer are automatically selected, reducing the work required to set up a workflow run and making the workflow more accessible to less experienced users who might not be as familiar with all the complexities of PreFreeSurfer.



Gradient_Distortion_Co...: Skip_Gradient_Distortion_Correction
FNIRT_Config: T1_2_MNI152_2mm.cnf
Template: MNI152
Unprocessed_T2_Images: Semantics_Test1_T2w_SPC1.nii.gz
Unprocessed_T1_Images: Semantics_Test1_T1w_MPR1.nii.gz

Variable	Constraint	Value
Brain_Size	wflow:hasParameterValue	150
Smoothing_Sigma	wflow:hasParameterValue	5

Figure 6: User interface for running our PreFreeSurfer workflow in WINGS.

Usability Testing Results

Test users were directed to set up workflow runs for all four of the testing scenarios described in our methods using both our semantic PreFreeSurfer workflow, and our collection of non-semantic workflows, which were also implemented in WINGS for a more direct comparison of semantic and non-semantic workflows. Following testing, participants filled out and submitted a feedback for consisting of 7 free-response questions plus the 10 questions included in the System Usability Scale (SUS). Responses to the SUS were used to calculate a usability score to evaluate the usability of our semantic workflow, while the free-response questions were used to provide insight into what factors led to the evaluated usability score.

Feedback was collected from a total of 5 advanced users with prior experience running PreFreeSurfer and other neuroimage processing pipelines. The mean System Usability Score for the semantic version of PreFreeSurfer was 61.5 with a standard deviation of 3.05. Though this score is on a 0 – 100 point scale, it is important to note that the System Usability Score is not a percentage, and cannot alone describe the absolute usability of the workflow. For context, approximate comparisons can be made to other scoring systems (Figure 7).

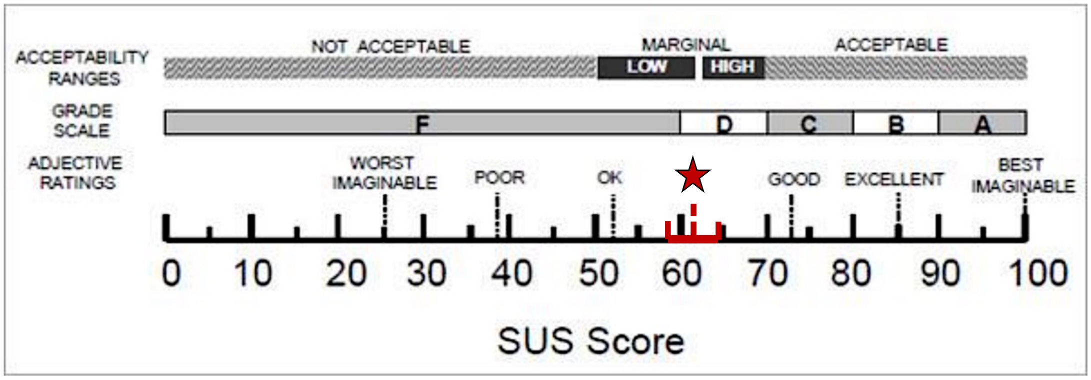


Figure 7: Contextualization of the System Usability Scale as it appears in (Brook 2013).
The results of our usability testing (mean SUS score of 61.5 with standard deviation of 3.05) are shown in red.

The usability score for our workflow is halfway between "OK" and "Good" on the SUS according to the Adjective Rating Scale[38], and would be considered "marginally acceptable" on the Acceptability Ranges spectrum.[39] The System Usability Scale is not diagnostic, and is not designed to be used to determine what aspects of a system contribute to its perceived usability. However, comments from the free-form response questions (Appendix A.4) suggest that contributing factors for this mediocre score could include concerns about the necessary time and expertise required to develop and maintain a semantic workflow, the accessibility of the data within WINGS, and the population of the metadata used to enforce the semantic constraints of the workflow. These concerns will be discussed in greater detail in our conclusions.

The feedback we received indicates that the operation of the semantic version of PreFreeSurfer was preferred to the non-semantic version used for comparison during testing. Specific comments indicate positive reception of the auto-detecting of parameters, as well as the use of metadata and overall flexibility of the workflow.

Discussion

Usability of Semantic Workflows

Our results suggest that the usability of semantic workflows is greater than that of traditional non-semantic workflows. While the concerns expressed by our test users regarding the feasibility of adopting this approach to other neuroimaging pipelines are valid, the ability of semantic workflows to leverage metadata to create a more flexible workflow with greatly simplified requirements for use was appreciated.

The use of semantics in our workflow leads to improved usability by handling all automatable decision making behind the scenes; only requiring a minimal amount of input from the user. Especially useful for large collaborative research efforts, this approach ensures the the same processing steps and parameters will be used regardless of who is running the workflow. This is beneficial to research as it removes sources of bias and enhances reproducibility.

It should be pointed out, however, that the logic governing how to process the data must be encoded into the workflow, which greatly increases the time and expertise required to develop new workflows. When talking about the use of workflows, we should distinguish between the workflow developer (who designs and creates a useable workflow) and the workflow user (who specifies the necessary inputs and runs the workflow). Though both roles can be filled by the same person, a workflow developer usually creates a workflow with the intention that others will be using it as well. When one chooses to develop a semantic workflow, they should keep in mind that the primary beneficiary is the workflow user, as the semantics can create a convenient and helpful environment for running a workflow at the cost of being much more difficult for the workflow developer to make. This necessitates an altruistic approach by the workflow developer, as they are accepting most of the work and responsibility for making sure the user's data is being processed correctly.

These results give us a good idea of how our workflow is received by experienced users of neuroimage processing workflows. Additional testing is required among novice users is required to get a more accurate measure of the true usability of our workflow, especially among users with limited experience with neuroimage processing and the PreFreeSurfer pipeline.

Feasibility of WINGS for Neuroimage Processing

The Importance of Metadata

There are a number of aspects about WINGS that raise concerns about its adoption into neuroimaging research. The most paramount being the population and security of the metadata assigned to the user's data. WINGS, being a web-based platform is designed for users to upload their data, then manually enter the necessary metadata for that data. Because the semantic constraints rely on the data being uploaded as the correct data type in the custom data hierarchy and properly annotated, this becomes the most troublesome aspect of using WINGS, and where it is most vulnerable to user error. Without a means to automate this process, WINGS only relocates the potential source for user error from workflow run setup, to the data upload process.

A python API enabling back-door access to WINGS's data library is currently in development, and would allow a developer to automate this process. External data upload scripts would be a necessary part of any WINGS workflow, further increasing the up-front cost of workflow development. Even then, there is no way to lock or restrict access to the metadata of files stored within WINGS, so it would still be possible for users to tamper with these values, whether they be well-meaning or not.

Data Accessibility

The data being stored within WINGS can be another inconvenience, as it contains no integrated data viewer. For one to inspect their data, they would have to download it and load it into a 3rd party viewer. In the neuroimaging field, researchers are working with several very large 3-dimensional images, so the need to download and possibly re-upload these images becomes time-consuming. The most obvious way around this is to keep a copy of all your data locally, but then you introduce a disconnect between the local data, and that stored within WINGS, leading to consistency issues should any modifications be made to one and not the other.

Unfortunately, accessing the data using 3rd party software while it resides within WINGS's data library is further complicated by the way WINGS stores data. Everything uploaded to WINGS is stored in a single directory, while the elaborate data hierarchy defined by the developer is manifested in the way WINGS indexes the data. Therefore, locating a file through back-end access is very different from locating it within WINGS. Consequently, WINGS would need added functionality to allow users to view their data or inspect results, or some way to quickly provide the file path of a file to 3rd party applications.

Development Costs

The cost for a lab to develop and maintain workflows is already high, as a significant amount of technical and domain expertise is required. Developing a semantic workflow adds to this burden, as it adds the need for developers to understand semantic reasoning and how to encode it into the workflow. As described in our methods, a variety of methods had to be used to implement the semantic constraints for PreFreeSurfer, ranging from defining simple rules to the design of abstract components. Researchers may not be willing to pay these up-front costs, especially when working with large-scale processing pipelines, such as the HCP Minimum Processing Pipeline.

WINGS also demands a considerable amount of foresight, as constructing components requires that all the necessary inputs and outputs first be defined in the data hierarchy. Any later modifications to a workflow could require complete reconstruction of the data type hierarchy, components, and/or the entire workflow. This is also a concern for the maintenance and versioning of a workflow. As new data processing methods are being developed, adding them to an established workflow could be troublesome if it was not designed to be sufficiently modular.

Even simple modifications to a workflow are tedious, since there is no way to duplicate an already established workflow, and requires that the entire thing be re-constructed exactly the same way, save for whatever minor changes are being made. The only way to duplicate something in WINGS is to download the entire domain, which includes not only any components and assembled workflows, but also any data that is being used with them, and re-upload everything as a separate domain.

This integration of workflow and data also has consequences for workflow sharing. Being a web-based platform, it is possible to allow collaborators to create accounts and utilize your workflows over the internet similar to Galaxy. However, doing this requires you to provide the computational capacity for their work and assume the responsibilities of data stewardship. Another option is to provide a copy of your workflows as an entire domain for them to upload onto their own installation of WINGS, though you would have to keep in mind what data that might include.

Conclusion

Our semantic PreFreeSurfer workflow employs semantic constraints to provide user guidance; restricting selectable inputs and semantically validating workflow runs. Through semantic reasoning, it also increases usability; minimizing the number of required inputs to set up a workflow run and automating the selection of all others. The ability to automatically choose between multiple processing paths based on the input allows it to handle a variety of processing situations. Automatic metadata propagation can also serve to track data provenance and enhance reproducibility. User testing has shown the workflow to be an acceptable platform for data processing, though matters of high developmental costs, data accessibility, and the process of uploading both data and metadata must be considered.

There remains work to be done for our semantic workflow to be useful in a laboratory setting. A means of handling the auxiliary tasks mentioned, such as uploading data to WINGS, as well as the extraction and population of metadata values for that data must be developed. There exists a python API currently in development that allows for back-end access to the data library in WINGS, and can be used to perform these tasks, however it is not yet fully developed. Such a tool would be essential to facilitate use of our workflow in any kind of high-throughput setting.

Additionally, the scaling up of our workflow to encompass the entire HCP Minimum Processing Pipeline would be required. PreFreeSurfer itself is only the first step in a larger overall workflow required to prepare raw neuroimaging data for analysis. The granularity at which our PreFreeSurfer workflow has been implemented is more detailed than would be desired if we were to run the entire Minimum Processing Pipeline. In this case, we would want to have the entire PreFreeSurfer step represented as a single component. We have been informed by the WINGS developers about possibility of adding an additional layer of abstraction to WINGS, allowing us to encapsulate our PreFreeSurfer workflow into a single component for use in a larger workflow. Should this functionality not be implemented, our PreFreeSurfer workflow would have to be re-constructed.

Appendix

A.1 Workflow Access

The domain for our semantic PreFreeSurfer workflow, which contains the workflow and sample data, can be downloaded from https://github.com/brandonkeibler/PreFreeSurfer-in-WINGS. Visualization and use of the workflow require the WINGS software, which can be downloaded from http://www.wings-workflows.org/download. Successfully running the semantic PreFreeSurfer pipeline to process data also requires the HCP minimum processing pipelines and their dependencies to be properly installed and configured. Source files and documentation available at https://github.com/Washington-University/Pipelines.

A.2 Semantic Rules

T1 Pre-processing:

```
[ DetermineAppropriateTemplates-Human:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Unprocessed_T1_Images')
(?i1 dcdom:Species ?spec)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Template')
equal('Human' ?spec)
-> (?i2 dcdom:Version 'MNI152')
]

[ DetermineTemplates:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?temp)
(?temp ac:hasArgumentID 'Template')
(?temp dcdom:Version ?ver)
(?c ac:hasInput ?i1)
```

```
(?i1 ac:hasArgumentID 'T1_Template')
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T1_Template_Mask')
(?c ac:hasInput ?i3)
(?i3 ac:hasArgumentID 'T1_2mm_Template')
(?c ac:hasInput ?i4)
(?i4 ac:hasArgumentID 'T1_2mm_Template_Mask')
-> (?i1 dcdom:Modality 'T1')
(?i1 dcdom:Resolution '0.7'^^http://www.w3.org/2001/XMLSchema#float)
(?i1 dcdom:Descriptor 'Full_Image')
(?i1 dcdom:Version ?ver)
(?i2 dcdom:Modality 'T1')
(?i2 dcdom:Resolution '0.7'^^http://www.w3.org/2001/XMLSchema#float)
(?i2 dcdom:Descriptor 'Brain_Mask')
(?i2 dcdom:Version ?ver)
(?i3 dcdom:Modality 'T1')
(?i3 dcdom:Resolution '2'^^http://www.w3.org/2001/XMLSchema#float)
(?i3 dcdom:Descriptor 'Full_Image')
(?i3 dcdom:Version ?ver)
(?i4 dcdom:Modality 'T1')
(?i4 dcdom:Resolution '2'^^http://www.w3.org/2001/XMLSchema#float)
(?i4 dcdom:Descriptor 'Brain_Mask')
(?i4 dcdom:Version ?ver)
]

[ DetermineFNIRTConfig:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?temp)
(?temp ac:hasArgumentID 'Template')
(?temp dcdom:Version ?ver)
(?c ac:hasInput ?config)
(?config ac:hasArgumentID 'FNIRT_Config')
-> (?config dcdom:Version ?ver)
]

[ DetermineGDCCoefficients-HCPSkyra:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Unprocessed_T1_Images')
(?i1 dcdom:Aquisition_Technology ?scanner1)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Gradient_Distortion_Coefficients')
equal(?scanner1 'HCP Custom Siemens Skyra')
-> (?i2 dcdom:Aquisition_Technology ?scanner1)
]

[ DetermineAppropriateTemplates-Macaque:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Unprocessed_T1_Images')
(?i1 dcdom:Species ?spec)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Template')
equal('Rhesus Macaque' ?spec)
-> (?i2 dcdom:Version 'MacaqueYerkes19')
]

[ CheckCollectionMetadata-Study:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T1_Images')
noValue(?i dcdom:Study ?value)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(All unprocessed T1 images must be from the same study)
]

[ CheckCollectionMetadata-PatientID:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T1_Images')
noValue(?i dcdom:Patient_ID ?value)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(All unprocessed T1 images must have the same patient ID)
]

[ CheckCollectionMetadata-Modality:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T1_Images')
noValue(?i dcdom:Modality ?value)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(All unprocessed T1 images must be T1 images)
]

[ EnforceGDCCoefficientAppropriate:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Unprocessed_T1_Images')
(?i1 dcdom:Aquisition_Technology ?scanner1)
(?c ac:hasInput ?i2)
```

```
(?i2 ac:hasArgumentID 'Gradient_Distortion_Coefficients')
(?i2 dcdom:Aquisition_Technology ?scanner2)
bound(?scanner2)
notEqual(?scanner1 '')
notEqual(?scanner1 ?scanner2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Selected gradient distortion coefficients file does not appear to be appropriate for the unprocessed images. Check the 'Acquisition_Technology' metadata of the unprocessed images. Leaving it blank allows for the use of any gradient distortion coefficients file.)
]

[ EnforceFNIRTConfigMatchesTemplate:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Template')
(?i1 dcdom:Version ?ver1)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'FNIRT_Config')
(?i2 dcdom:Version ?ver2)
notEqual(?ver1 ?ver2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : FNIRT_Config is not appropriate for the selected Template)
]

[ EnforceUnprocessedImage=T1:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T1_Images')
-> (?i dcdom:Modality 'T1')
print(Please select all desired 'Unprocessed_T1_Images' and 'Unprocessed_T2_Images' before requesting data suggestions)
]

[ ForwardMetadata:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T1_Images')
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o dcdom:Patient_ID ?id)
(?o dcdom:Study ?study)
(?o dcdom:Species ?species)
(?o dcdom:Modality ?mod)
(?o dcdom:Sample_Spacing ?space)
(?o dcdom:Unwarp_Direction ?dir)
(?o dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
]

[ EnforceCorrectTemplate:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Unprocessed_T1_Images')
(?i1 dcdom:Species ?spec1)
(?i1 dcdom:Allow_Any_Template ?any)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Template')
(?i2 dcdom:Species ?spec2)
notEqual(?any 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
notEqual(?spec1 ?spec2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Selected template is not intended for the species of the selected unprocessed images. Images are ?spec1 while template is ?spec2)
]

[ DetermineAppropriateTemplates-Other:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'Unprocessed_T1_Images')
(?i1 dcdom:Species ?spec)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Template')
notEqual('Rhesus Macaque' ?spec)
notEqual('Human' ?spec)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Unable to find a template that matches the species of the unprocessed images)
]
```

```
[ ForwardMetadata-AquisitionTechnology:
(?c rdf:type acdom:T1_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T1_Images')
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
]
```

When gradient distortion correction is performed:

```
[ SetGDCMethodMetadata:
(?c rdf:type acdom:T1_Pre-processing_WithGDCClass)
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
-> (?o dcdom:Gradient_Distortion_Corrected 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
]
```

When gradient distortion correction is not performed:

```
[ SetGDCMethodMetadata:
(?c rdf:type acdom:T1_Pre-processing_WithGDCClass)
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
-> (?o dcdom:Gradient_Distortion_Corrected ‘false’^^http://www.w3.org/2001/XMLSchema#boolean)
]
```

T2 Pre-processing:

```
[ ForwardMetadata-AquisitionTechnology:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T2_Images')
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
]

[ PrintErrorInstructions:
(?c rdf:type acdom:T2_Pre-processingClass)
-> print(Suggestion : 'Unprocessed_T2_Images' must be T2 images from the same individual as the T1 images. If none are available select 'No_Input')
]

[ CheckCollectionMetadata-PatientID:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T2_Images')
noValue(?i dcdom:Patient_ID ?value)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(All unprocessed T2 images must have the same patient ID)
]

[ CheckCollectionMetadata-Study:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T2_Images')
noValue(?i dcdom:Study ?value)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(All unprocessed images must be from the same study)
]

[ EnforceUnprocessedImage=T2:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T2_Images')
-> (?i dcdom:Modality 'T2')
]

[ ForwardMetadata:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T2_Images')
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o dcdom:Patient_ID ?id)
(?o dcdom:Study ?study)
(?o dcdom:Species ?species)
(?o dcdom:Modality ?mod)
(?o dcdom:Sample_Spacing ?space)
(?o dcdom:Unwarp_Direction ?dir)
(?o dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
]

[ DetermineTemplates:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?temp)
(?temp ac:hasArgumentID 'Template')
(?temp dcdom:Version ?ver)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T2_Template')
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Template_Mask')
(?c ac:hasInput ?i3)
(?i3 ac:hasArgumentID 'T2_2mm_Template')
(?c ac:hasInput ?i4)
(?i4 ac:hasArgumentID 'T2_2mm_Template_Mask')
-> (?i1 dcdom:Modality 'T2')
(?i1 dcdom:Resolution '0.7'^^http://www.w3.org/2001/XMLSchema#float)
(?i1 dcdom:Descriptor 'Full_Image')
(?i1 dcdom:Version ?ver)
(?i2 dcdom:Modality 'T1')
(?i2 dcdom:Resolution '0.7'^^http://www.w3.org/2001/XMLSchema#float)
(?i2 dcdom:Descriptor 'Brain_Mask')
(?i2 dcdom:Version ?ver)
(?i3 dcdom:Modality 'T2')
(?i3 dcdom:Resolution '2'^^http://www.w3.org/2001/XMLSchema#float)
(?i3 dcdom:Descriptor 'Full_Image')
(?i3 dcdom:Version ?ver)
(?i4 dcdom:Modality 'T1')
(?i4 dcdom:Resolution '2'^^http://www.w3.org/2001/XMLSchema#float)
(?i4 dcdom:Descriptor 'Brain_Mask')
(?i4 dcdom:Version ?ver)
]

[ CheckCollectionMetadata-Modality:
(?c rdf:type acdom:T2_Pre-processingClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Unprocessed_T2_Images')
noValue(?i dcdom:Modality ?value)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(All unprocessed T2 images must be T2 images)
]
```

When gradient distortion correction is performed:

```
[ SetGDCMethodMetadata:
(?c rdf:type acdom:T2_Pre-processing_WithGDCClass)
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
-> (?o dcdom:Gradient_Distortion_Corrected 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
]
```

When gradient distortion correction is not performed:

```
[ SetGDCMethodMetadata:
(?c rdf:type acdom:T2_Pre-processing_NoGDCClass)
(?c ac:hasOutput ?o)
(?o ac:hasArgumentID 'Processed_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Processed_Brain_Image')
-> (?o dcdom:Gradient_Distortion_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
]
```

Readout Distortion Correction:

```
[ ForwardT1Metadata:
(?c rdf:type acdom:Readout_Distortion_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Uncorrected_Image')
(?c ac:hasOutput ?o1)
```

```
(?o1 ac:hasArgumentID 'T1_RDC_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'T1_RDC_Brain_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o1 dcdom:Patient_ID ?id)
(?o1 dcdom:Study ?study)
(?o1 dcdom:Species ?species)
(?o1 dcdom:Modality ?mod)
(?o1 dcdom:Sample_Spacing ?space)
(?o1 dcdom:Unwarp_Direction ?dir)
(?o1 dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
]

[ ForwardT2Metadata:
(?c rdf:type acdom:Readout_Distortion_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Uncorrected_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T2_RDC_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o1 dcdom:Patient_ID ?id)
(?o1 dcdom:Study ?study)
(?o1 dcdom:Species ?species)
(?o1 dcdom:Modality ?mod)
(?o1 dcdom:Sample_Spacing ?space)
(?o1 dcdom:Unwarp_Direction ?dir)
(?o1 dcdom:Age ?age)
]

[ ForwardT1GDCMethodMetadata:
(?c rdf:type acdom:Readout_Distortion_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Uncorrected_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T1_RDC_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'T1_RDC_Brain_Image')
(?i dcdom:Gradient_Distortion_Corrected ?GDC)
-> (?o1 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o2 dcdom:Gradient_Distortion_Corrected ?GDC)
]

[ ForwardT1Metadata-AquisitionTechnology:
(?c rdf:type acdom:Readout_Distortion_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Uncorrected_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T1_RDC_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'T1_RDC_Brain_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o1 dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
]

[ ForwardT2Metadata-AquisitionTechnology:
(?c rdf:type acdom:Readout_Distortion_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Uncorrected_Image')
(?c ac:hasOutput ?o1)
(?o2 ac:hasArgumentID 'T2_RDC_Brain_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o1 dcdom:Aquisition_Technology ?scanner)
]

[ ForwardT2GDCMethodMetadata:
(?c rdf:type acdom:Readout_Distortion_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Uncorrected_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T2_RDC_Image')
```

```
(?i dcdom:Gradient_Distortion_Corrected ?GDC)
-> (?o1 dcdom:Gradient_Distortion_Corrected ?GDC)
]
```

When no readout distortion correction is performed:

```
[ SetRDCMethodMetadata:
(?c rdf:type acdom:RDC-NoneClass)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T1_RDC_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'T1_RDC_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'T2_RDC_Image')
-> (?o1 dcdom:Readout_Distortion_Correction_Method 'None')
(?o2 dcdom:Readout_Distortion_Correction_Method 'None')
(?o3 dcdom:Readout_Distortion_Correction_Method 'None')
]
```

When Siemens readout distortion correction is performed:

```
[ DetermineEchoDifference:
(?c rdf:type acdom:RDC-SiemensClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Fieldmap_Magnitude')
(?i dcdom:Echo_Difference ?val)
(?c ac:hasInput ?p)
(?p ac:hasArgumentID 'Echo_Difference')
-> (?p ac:hasValue ?val)
]

[ SetRDCMethodMetadata:
(?c rdf:type acdom:RDC-SiemensClass)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T1_RDC_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'T1_RDC_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'T2_RDC_Image')
-> (?o1 dcdom:Readout_Distortion_Correction_Method 'Siemens')
(?o2 dcdom:Readout_Distortion_Correction_Method 'Siemens')
(?o3 dcdom:Readout_Distortion_Correction_Method 'Siemens')
]

[ DetermineUnwarpDirection:
(?c rdf:type acdom:RDC-SiemensClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Uncorrected_Image')
(?i dcdom:Unwarp_Direction ?val)
(?c ac:hasInput ?p)
(?p ac:hasArgumentID 'Unwarp_Direction')
-> (?p ac:hasValue ?val)
]

[ DetermineT2SampleSpacing:
(?c rdf:type acdom:RDC-SiemensClass)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Uncorrected_Image')
(?i2 dcdom:Sample_Spacing ?val2)
(?c ac:hasInput ?p2)
(?p2 ac:hasArgumentID 'T2_Sample_Spacing')
-> (?p2 ac:hasValue ?val2)
]

[ CheckForSiemensFiles:
(?c rdf:type acdom:RDC-SiemensClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Uncorrected_Image')
(?i1 dcdom:Study ?study)
(?i1 dcdom:Patient_ID ?id)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Fieldmap_Magnitude')
(?c ac:hasInput ?i3)
(?i3 ac:hasArgumentID 'Fieldmap_Phase')
-> (?i2 dcdom:Study ?study)
(?i2 dcdom:Patient_ID ?id)
(?i3 dcdom:Study ?study)
(?i3 dcdom:Patient_ID ?id)
]

[ DetermineT1SampleSpacing:
(?c rdf:type acdom:RDC-SiemensClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Uncorrected_Image')
(?i1 dcdom:Sample_Spacing ?val)
(?c ac:hasInput ?p1)
(?p1 ac:hasArgumentID 'T1_Sample_Spacing')
-> (?p1 ac:hasValue ?val)
]
```

When TopUp readout distortion correction is performed:

```
[ SetRDCMethodMetadata:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'T1_RDC_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'T1_RDC_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'T2_RDC_Image')
-> (?o1 dcdom:Readout_Distortion_Correction_Method 'TopUp')
(?o2 dcdom:Readout_Distortion_Correction_Method 'TopUp')
(?o3 dcdom:Readout_Distortion_Correction_Method 'TopUp')
]

[ DetermineT1SampleSpacing:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Uncorrected_Image')
(?i1 dcdom:Sample_Spacing ?val)
(?c ac:hasInput ?p1)
(?p1 ac:hasArgumentID 'T1_Sample_Spacing')
-> (?p1 ac:hasValue ?val)
]

[ DetermineT2SampleSpacing:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Uncorrected_Image')
(?i2 dcdom:Sample_Spacing ?val2)
(?c ac:hasInput ?p2)
(?p2 ac:hasArgumentID 'T2_Sample_Spacing')
-> (?p2 ac:hasValue ?val2)
]

[ DetermineDwellTime:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Spin_Echo_Phase_Positive')
(?i dcdom:Dwell_Time ?val)
(?c ac:hasInput ?p)
(?p ac:hasArgumentID 'Dwell_Time')
-> (?p ac:hasValue ?val)
]

[ DetermineSEUnwarpDirection:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'Spin_Echo_Phase_Positive')
(?i dcdom:Unwarp_Direction ?val)
(?c ac:hasInput ?p)
(?p ac:hasArgumentID 'SE_Unwarp_Direction')
-> (?p ac:hasValue ?val)
]

[ CheckForTopUpFiles:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Uncorrected_Image')
(?i1 dcdom:Study ?study)
(?i1 dcdom:Patient_ID ?id)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'Spin_Echo_Phase_Positive')
(?c ac:hasInput ?i3)
(?i3 ac:hasArgumentID 'Spin_Echo_Phase_Negative')
-> (?i2 dcdom:Study ?study)
(?i2 dcdom:Patient_ID ?id)
(?i3 dcdom:Study ?study)
(?i3 dcdom:Patient_ID ?id)
]

[ DetermineUnwarpDirection:
(?c rdf:type acdom:RDC-TopUpClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Uncorrected_Image')
(?i dcdom:Unwarp_Direction ?val)
(?c ac:hasInput ?p)
(?p ac:hasArgumentID 'Unwarp_Direction')
-> (?p ac:hasValue ?val)
]
```

Bias Field Correction:

```
[ ForwardT1Metadata:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T1_Brain_Image')
```

```
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o1 dcdom:Patient_ID ?id)
(?o1 dcdom:Study ?study)
(?o1 dcdom:Species ?species)
(?o1 dcdom:Modality ?mod)
(?o1 dcdom:Sample_Spacing ?space)
(?o1 dcdom:Unwarp_Direction ?dir)
(?o1 dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
]

[ ForwardT2Metadata-AquisitionTechnology:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Image_T1-registered')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T2_Brain_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o1 dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
]

[ ForwardT1RDCMethodMetadata:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T1_Brain_Image')
(?i dcdom:Readout_Distortion_Correction_Method ?RDC)
-> (?o1 dcdom:Readout_Distortion_Correction_Method ?RDC)
(?o2 dcdom:Readout_Distortion_Correction_Method ?RDC)
]

[ ForwardT1GDCMethodMetadata:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T1_Brain_Image')
(?i dcdom:Gradient_Distortion_Corrected ?GDC)
-> (?o1 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o2 dcdom:Gradient_Distortion_Corrected ?GDC)
]

[ ForwardT2RDCMethodMetadata:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Image_T1-registered')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T2_Brain_Image')
(?i dcdom:Readout_Distortion_Correction_Method ?RDC)
-> (?o1 dcdom:Readout_Distortion_Correction_Method ?RDC)
(?o2 dcdom:Readout_Distortion_Correction_Method ?RDC)
]

[ ForwardT2Metadata:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Image_T1-registered')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T2_Brain_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
```

```
-> (?o1 dcdom:Patient_ID ?id)
(?o1 dcdom:Study ?study)
(?o1 dcdom:Species ?species)
(?o1 dcdom:Modality ?mod)
(?o1 dcdom:Sample_Spacing ?space)
(?o1 dcdom:Unwarp_Direction ?dir)
(?o1 dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
]

[ ForwardT2GDCMethodMetadata:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T2_Image_T1-registered')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T2_Brain_Image')
(?i dcdom:Gradient_Distortion_Corrected ?GDC)
-> (?o1 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o2 dcdom:Gradient_Distortion_Corrected ?GDC)
]

[ ForwardT1Metadata-AquisitionTechnology:
(?c rdf:type acdom:Bias_Field_CorrectionClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T1_Brain_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o1 dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
]
```

When the sqrt(T1xT2) bias field correction algorithm is used:

```
[ SetBFCorrectionMetadata:
(?c rdf:type acdom:BFC_sqrtT1xT2Class)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T1_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'BF_Restored_T2_Image')
(?c ac:hasOutput ?o4)
(?o4 ac:hasArgumentID 'BF_Restored_T2_Brain_Image')
-> (?o1 dcdom:Bias_Field_Corrected 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
(?o2 dcdom:Bias_Field_Corrected 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
(?o3 dcdom:Bias_Field_Corrected 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
(?o4 dcdom:Bias_Field_Corrected 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
]

[ EnforceInputsHaveSameAquisitionTechnology:
(?c rdf:type acdom:BFC_sqrtT1xT2Class)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Image')
(?i1 dcdom:Aquisition_Technology ?val1)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Image_T1-registered')
(?i2 dcdom:Aquisition_Technology ?val2)
notEqual(?val1 ?val2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Patient T1 and T2 images appear to have been acquired from different scanning technology. Check input file metadata)
]

[ EnforceInputsHaveSameSpecies:
(?c rdf:type acdom:BFC_sqrtT1xT2Class)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Image')
(?i1 dcdom:Species ?val1)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Image_T1-registered')
(?i2 dcdom:Species ?val2)
notEqual(?val1 ?val2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Patient T1 and T2 images appear to be from different species. Check input file metadata)
]

[ EnforceInputsHaveSameAge:
(?c rdf:type acdom:BFC_sqrtT1xT2Class)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Image')
(?i1 dcdom:Age ?val1)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Image_T1-registered')
(?i2 dcdom:Age ?val2)
notEqual(?val1 ?val2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Patient T1 and T2 images appear to have different ages. Check input file metadata)
]

[ EnforceInputsHaveSamePatientID:
(?c rdf:type acdom:BFC_sqrtT1xT2Class)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Image')
(?i1 dcdom:Patient_ID ?val1)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T2_Image_T1-registered')
(?i2 dcdom:Patient_ID ?val2)
notEqual(?val1 ?val2)
-> (?c ac:isInvalid 'true'^^http://www.w3.org/2001/XMLSchema#boolean)
print(Error : Patient T1 and T2 images must have the same patient ID)
]
```

When bias field correction is not performed:

```
[ SetBFCorrectionMetadata:
(?c rdf:type acdom:BFC_sqrtT1xT2Class)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'BF_Restored_T1_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'BF_Restored_T2_Image')
(?c ac:hasOutput ?o4)
(?o4 ac:hasArgumentID 'BF_Restored_T2_Brain_Image')
-> (?o1 dcdom:Bias_Field_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
(?o2 dcdom:Bias_Field_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
(?o3 dcdom:Bias_Field_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
(?o4 dcdom:Bias_Field_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
]
```

Atlas Registration:

```
[ SetT2NoBFCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_T2_Image')
-> (?o1 dcdom:Bias_Field_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
]

[ ForwardT1Metadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T1_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o1 dcdom:Patient_ID ?id)
(?o1 dcdom:Study ?study)
(?o1 dcdom:Species ?species)
(?o1 dcdom:Modality ?mod)
(?o1 dcdom:Sample_Spacing ?space)
(?o1 dcdom:Unwarp_Direction ?dir)
(?o1 dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
(?o3 dcdom:Patient_ID ?id)
(?o3 dcdom:Study ?study)
(?o3 dcdom:Species ?species)
(?o3 dcdom:Modality ?mod)
(?o3 dcdom:Sample_Spacing ?space)
(?o3 dcdom:Unwarp_Direction ?dir)
(?o3 dcdom:Age ?age)
]

[ ForwardT2Metadata:
```

```
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Registered_T2_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T2_Image')
(?i dcdom:Patient_ID ?id)
(?i dcdom:Study ?study)
(?i dcdom:Species ?species)
(?i dcdom:Modality ?mod)
(?i dcdom:Sample_Spacing ?space)
(?i dcdom:Unwarp_Direction ?dir)
(?i dcdom:Age ?age)
-> (?o1 dcdom:Patient_ID ?id)
(?o1 dcdom:Study ?study)
(?o1 dcdom:Species ?species)
(?o1 dcdom:Modality ?mod)
(?o1 dcdom:Sample_Spacing ?space)
(?o1 dcdom:Unwarp_Direction ?dir)
(?o1 dcdom:Age ?age)
(?o2 dcdom:Patient_ID ?id)
(?o2 dcdom:Study ?study)
(?o2 dcdom:Species ?species)
(?o2 dcdom:Modality ?mod)
(?o2 dcdom:Sample_Spacing ?space)
(?o2 dcdom:Unwarp_Direction ?dir)
(?o2 dcdom:Age ?age)
(?o3 dcdom:Patient_ID ?id)
(?o3 dcdom:Study ?study)
(?o3 dcdom:Species ?species)
(?o3 dcdom:Modality ?mod)
(?o3 dcdom:Sample_Spacing ?space)
(?o3 dcdom:Unwarp_Direction ?dir)
(?o3 dcdom:Age ?age)
]

[ ForwardT1BFCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Brain_Image')
(?i dcdom:Bias_Field_Corrected ?BFC)
-> (?o1 dcdom:Bias_Field_Corrected ?BFC)
(?o2 dcdom:Bias_Field_Corrected ?BFC)
]

[ ForwardT1GDCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T1_Image')
(?i dcdom:Gradient_Distortion_Corrected ?GDC)
-> (?o1 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o2 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o3 dcdom:Gradient_Distortion_Corrected ?GDC)
]

[ ForwardT1Metadata-AquisitionTechnology:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T1_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o1 dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
(?o3 dcdom:Aquisition_Technology ?scanner)
]

[ ForwardT2RDCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Registered_T2_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Image')
```

```
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T2_Image')
(?i dcdom:Readout_Distortion_Correction_Method ?RDC)
-> (?o1 dcdom:Readout_Distortion_Correction_Method ?RDC)
(?o2 dcdom:Readout_Distortion_Correction_Method ?RDC)
(?o3 dcdom:Readout_Distortion_Correction_Method ?RDC)
]

[ ForwardT1RDCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T1_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T1_Image')
(?i dcdom:Readout_Distortion_Correction_Method ?RDC)
-> (?o1 dcdom:Readout_Distortion_Correction_Method ?RDC)
(?o2 dcdom:Readout_Distortion_Correction_Method ?RDC)
(?o3 dcdom:Readout_Distortion_Correction_Method ?RDC)
]

[ ForwardT2GDCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Registered_T2_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T2_Image')
(?i dcdom:Gradient_Distortion_Corrected ?GDC)
-> (?o1 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o2 dcdom:Gradient_Distortion_Corrected ?GDC)
(?o3 dcdom:Gradient_Distortion_Corrected ?GDC)
]

[ SetT1NoBFCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_T1_Image')
-> (?o1 dcdom:Bias_Field_Corrected 'false'^^http://www.w3.org/2001/XMLSchema#boolean)
]

[ ForwardT2BFCMethodMetadata:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Registered_T2_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Brain_Image')
(?i dcdom:Bias_Field_Corrected ?BFC)
-> (?o1 dcdom:Bias_Field_Corrected ?BFC)
(?o2 dcdom:Bias_Field_Corrected ?BFC)
]

[ ForwardT2Metadata-AquisitionTechnology:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i)
(?i ac:hasArgumentID 'T1_Registered_T2_Image')
(?c ac:hasOutput ?o1)
(?o1 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Image')
(?c ac:hasOutput ?o2)
(?o2 ac:hasArgumentID 'Atlas_Space_BF_Restored_T2_Brain_Image')
(?c ac:hasOutput ?o3)
(?o3 ac:hasArgumentID 'Atlas_Space_T2_Image')
(?i dcdom:Aquisition_Technology ?scanner)
-> (?o1 dcdom:Aquisition_Technology ?scanner)
(?o2 dcdom:Aquisition_Technology ?scanner)
(?o3 dcdom:Aquisition_Technology ?scanner)
]

[ DetermineTemplateBrain:
(?c rdf:type acdom:Atlas_RegistrationClass)
(?c ac:hasInput ?i1)
(?i1 ac:hasArgumentID 'T1_Template')
(?i1 dcdom:Version ?ver)
(?c ac:hasInput ?i2)
(?i2 ac:hasArgumentID 'T1_Template_Brain')
-> (?i2 dcdom:Version ?ver)
(?i2 dcdom:Resolution '0.7'^^http://www.w3.org/2001/XMLSchema#float)
(?i2 dcdom:Descriptor 'Brain_Image')
(?i2 dcdom:Modality 'T1')
]
```

A.3 Testing Instructions

The instructions provided to test users to guide them through the testing process can be found at https://github.com/brandonkeibler/PreFreeSurfer-in-WINGS.

A.4 Testing Feedback

The completed post-test questionnaires we received as feedback from our testing participants can be found at https://github.com/brandonkeibler/PreFreeSurfer-in-WINGS.

References

[1] Ai T, Morelli JN, Hu X, Hao D, Goerner FL, Ager B, Runge VM. A historical overview of magnetic resonance imaging, focusing on technological innovations. Invest Radiol. 2012 Dec;47(12):725-41.

[2] Hart MG, Housden CR, Suckling J, Tait R, Young A, Müller U, Newcombe VFJ, Jalloh I, Pearson B, Cross J, Trivedi RA, Pickard JD, Sahakian BJ, Hutchinson PJ. Advanced magnetic resonance imaging and neuropsychological assessment for detecting brain injury in a prospective cohort of university amateur boxers. Neuroimage Clin. 2017 Apr 26;15:194-199.

[3] Bruzzone MG, D'Incerti L, Farina LL, Cuccarini V, Finocchiaro G. CT and MRI of brain tumors. Q J Nucl Med Mol Imaging. 2012 Apr;56(2):112-37.

[4] Szwarc P, Kawa J, Rudzki M, Pietka E. Automatic brain tumour detection and neovasculature assessment with multiseries MRI analysis. Comput Med Imaging Graph. 2015 Dec;46 Pt 2:178-90.

[5] Bahadure NB, Ray AK, Thethi HP. Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM. Int J Biomed Imaging. 2017;2017:9749108. doi: 10.1155/2017/9749108. Epub 2017 Mar 6.

[6] Dubois J, Dehaene-Lambertz G, Kulikova S, Poupon C, Hüppi PS, Hertz-Pannier L. The early development of brain white matter: a review of imaging studies in fetuses, newborns and infants. Neuroscience. 2014 Sep 12;276:48-71.

[7] Benkarim OM, Sanroma G, Zimmer VA, Muñoz-Moreno E, Hahner N, Eixarch E, Camara O, González Ballester MA, Piella G. Toward the automatic quantification of in utero brain development in 3D structural MRI: A review. Hum Brain Mapp. 2017 May;38(5):2772-2787. doi: 10.1002/hbm.23536. Epub 2017 Feb 14.

[8] Yamasue H. Function and structure in social brain regions can link oxytocin-receptor genes with autistic social behavior. Brain Dev. 2013 Feb;35(2):111-8.

[9] Blumenfeld H. Impaired consciousness in epilepsy. Lancet Neurol. 2012 Sep;11(9):814-26.

[10] Costa Dias TG, Iyer SP, Carpenter SD, Cary RP, Wilson VB, Mitchell SH, Nigg JT, Fair DA. Characterizing heterogeneity in children with and without ADHD based on reward system connectivity. Dev Cogn Neurosci. 2015 Feb;11:155-74.

[11] Van Essen DC, Smith SM, Barch DM, Behrens TE, Yacoub E, Ugurbil K; WU-Minn HCP Consortium. The WU-Minn Human Connectome Project: an overview. Neuroimage. 2013 Oct 15;80:62-79.

[12] Pezoulas VC, Zervakis M, Michelogiannis S, Klados MA. Resting-State Functional Connectivity and Network Analysis of Cerebellum with Respect to Crystallized IQ and Gender. Front Hum Neurosci. 2017 Apr 26;11:189.

[13] Burks JD, Boettcher LB, Conner AK, Glenn CA, Bonney PA, Baker CM, Briggs RG, Pittman NA, O'Donoghue DL, Wu DH, Sughrue ME. White matter connections of the inferior parietal lobule: A study of surgical anatomy. Brain Behav. 2017 Mar 8;7(4):e00640.

[14] Glasser MF, Sotiropoulos SN, Wilson JA, Coalson TS, Fischl B, Andersson JL, Xu J, Jbabdi S, Webster M, Polimeni JR, Van Essen DC, Jenkinson M; WU-Minn HCP Consortium. The minimal preprocessing pipelines for the Human Connectome Project. Neuroimage. 2013 Oct 15;80:105-24.

[15] Peechatka AL, Janes AC. Association Between Reward Reactivity and Drug Use Severity is Substance Dependent: Preliminary Evidence From the Human Connectome Project. Nicotine Tob Res. 2017 Jun 1;19(6):710-715.

[16] Pujol S, Cabeen R, Sébille SB, Yelnik J, François C, Fernandez Vidal S, Karachi C, Zhao Y, Cosgrove GR, Jannin P, Kikinis R, Bardinet E. In vivo Exploration of the Connectivity between the Subthalamic Nucleus and the Globus Pallidus in the Human Brain Using Multi-Fiber Tractography. Front Neuroanat. 2017 Jan 19;10:119.

[17] Riccelli R, Toschi N, Nigro S, Terracciano A, Passamonti L. Surface-based morphometry reveals the neuroanatomical basis of the five-factor model of personality. Soc Cogn Affect Neurosci. 2017 Jan 24. pii: nsw175.

[18] Kochunov P, Thompson PM, Winkler A, Morrissey M, Fu M, Coyle TR, Du X, Muellerklein F, Savransky A, Gaudiot C, Sampath H, Eskandar G, Jahanshad N, Patel B, Rowland L, Nichols TE, O'Connell JR, Shuldiner AR, Mitchell BD, Hong LE. The common genetic influence over processing speed and white matter microstructure: Evidence from the Old Order Amish and Human Connectome Projects. Neuroimage. 2016 Jan 15;125:189-97.

[19] Abdollahi RO, Kolster H, Glasser MF, Robinson EC, Coalson TS, Dierker D, Jenkinson M, Van Essen DC, Orban GA. Correspondences between retinotopic areas and myelin maps in human visual cortex. Neuroimage. 2014 Oct 1;99:509-24.

[20] Poldrack RA, Laumann TO, Koyejo O, Gregory B, Hover A, Chen MY, Gorgolewski KJ, Luci J, Joo SJ, Boyd RL, Hunicke-Smith S, Simpson ZB, Caven T, Sochat V, Shine JM, Gordon E, Snyder AZ, Adeyemo B, Petersen SE, Glahn DC, Reese Mckay D, Curran JE, Göring HH,Carless MA, Blangero J, Dougherty R, Leemans A, Handwerker DA, Frick L, Marcotte EM, Mumford JA. Long-term neural and physiological phenotyping of a single human. Nat Commun. 2015 Dec 9;6:8885.

[21] Landman BA, Huang AJ, Gifford A, Vikram DS, Lim IA, Farrell JA, Bogovic JA, Hua J, Chen M, Jarso S, Smith SA, Joel S, Mori S, Pekar JJ, Barker PB, Prince JL, van Zijl PC. Multi-parametric neuroimaging reproducibility: a 3-T resource study. Neuroimage. 2011 Feb 14;54(4):2854-66.

[22] Bonilha L, Gleichgerrcht E, Fridriksson J, Rorden C, Breedlove JL, Nesland T, Paulus W, Helms G, Focke NK. Reproducibility of the Structural Brain Connectome Derived from Diffusion Tensor Imaging. PLoS One. 2015 Sep 2;10(8):e0135247.

[23] Guberman L. Irreproducible life sciences research in U.S. costs $28 billion. ScienceNews. 2015 Jul 11:188(1):5.

[24] Goecks J, Nekrutenko A, Taylor J, and The Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 2010 Aug 25;11(8):R86.

[25] Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J. "Galaxy: a web-based genome analysis tool for experimentalists. Current Protocols in Molecular Biology. 2010 Jan; Chapter 19:Unit 19.10.1-21.

[26] Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A. "Galaxy: a platform for interactive large-scale genome analysis. Genome Research. 2005 Oct; 15(10):1451-5.

[27] Rex DE, Ma JQ, Toga AW. he LONI Pipeline Processing Environment. Neuroimage. 2003 Jul;19(3):1033-48.

[28] Majer A, Caligiuri KA, Booth SA. A user-friendly computational workflow for the analysis of microRNA deep sequencing data. Methods Mol Biol. 2013;936:35-45.

[29] Wei X, Wang X. A computational workflow to identify allele-specific expression and epigenetic modification in maize. Genomics Proteomics Bioinformatics. 2013 Aug;11(4):247-52.

[30] Aiche S, Sachsenberg T, Kenar E, Walzer M, Wiswedel B, Kristl T, Boyles M, Duschl A, Huber CG, Berthold MR, Reinert K, Kohlbacher O. Workflows for automated downstream data analysis and visualization in large-scale computational mass spectrometry. Proteomics. 2015 Apr;15(8):1443-7.

[31] Gilad Y, Nadassy K, Senderowitz H. A reliable computational workflow for the selection of optimal screening libraries. J Cheminform. 2015 Dec 11;7:61.

[32] Zheng CL, Ratnakar V, Gil Y, McWeeney SK. Use of semantic workflows to enhance transparency and reproducibility in clinical omics. Genome Med. 2015 Jul 25;7(1):73.

[33] Gil Y, Ratnakar V, Kim J, Gonzalez-Calero PA, Groth P, Moody J, Deelman E. Wings: Intelligent Workflow-Based Design of Computational Experiments. IEEE Intelligent Systems. 2011; 26(1):62-72.

[34] Kumar VS, Sadayappan P, Mehta G, Vahi K, Deelman E, Ratnakar V, Kim J, Gil Y, Hall M, Kurc T, Saltz J. An integrated framework for parameter-based optimization of scientific workflows. Proc Int Symp High Perform Distrib Comput. 2009:177-186.

[35] Juntu J., Sijbers J., Van Dyck D., Gielen J. Bias Field Correction for MRI Images. In: Computer Recognition Systems: Advances in Soft Computing. 2005: 30:543-551

[36] Brooke J. SUS: a retrospective. Journal of Usability Studies. 2013 Feb: 8(2):29-40.

[37] Tullis TS, Stetson JN. A comparison of questionnaires for assessing website usability. Proceedings of UPA 2004 Conference. Minneapolis, Minnesota. 2004 June 7-11.

[38] Bangor A, Kortum PT, Miller JT. Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of Usability Studies. 2009;4(3):114-123

[39] Bangor A, Kortum P, Miller JA. The System Usability Scale (SUS): An Empirical Evaluation. International Journal of Human-Computer Interaction. 2008;24(6)