Sampled Iterative Local Approximation (SILA) Quick Guide

Written By: Tobey J Betthauser, PhD
University of Wisconsin-Madison

Contributors:
Tobey J Betthauser
Murat Bilgel
Rebecca Langhough Koscik
Bruno M Jedynak
Yang An
Kristina A Kellett
Abhay Moghekar
Erin M Jonaitis
Charles K Stone
Corinne D Engelman
Sanjay Asthana
Bradley T Christian
Dean F Wong
Marilyn Albert
Susan M Resnick
Sterling C Johnson

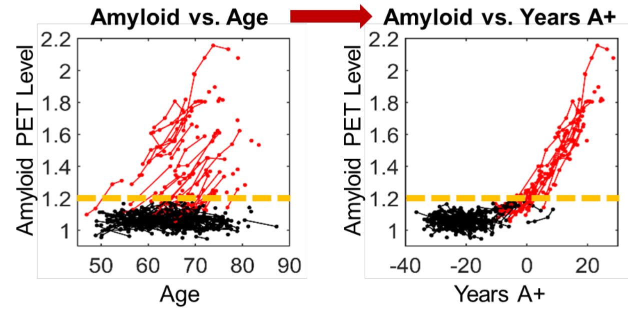
Betthauser Neuroimaging Laboratory
Wisconsin Alzheimer's Disease Research Center
University of Wisconsin-Madison
600 Highland Avenue, CSC K6/428
Madison, WI 53792
https://github.com/Betthauser-Neuro-Lab

Table of Contents

1 Introduction to SILA

Sampled Iterative Local Approximation (SILA) was developed by Dr. Tobey Betthauser at the University of Wisconsin-Madison to model the longitudinal time course of beta-amyloid accumulation in Alzheimer's Disease measured using positron emission tomography imaging. The idea is that the process of amyloid accumulation occurs over two or more decades, but shorter longitudinal observations (e.g., 5-7 years) from several different research participants at different stages of disease can be used to piece together the entire 20+ year-long process of amyloid accumulation. SILA assumes that this process follows first-order kinetics such that the rate of change of amyloid is a function of the amount of amyloid, and further assumes that amyloid accumulation over time is monotonic (creating a unique pairing between amyloid level and A+ time). In this application the goal was to model the relationship between amyloid accumulation over time, and then use this information to reorder observations as a function of how long individuals had amyloid in their brain. This enables estimation of the age at which an individual had enough amyloid to be detected by PET imaging (i.e., an estimated amyloid onset age). See Betthauser, et al., *BRAIN*, 2022 for details on this specific application and model validation.

Figure 1: Conceptualization of SILA applied to amyloid PET



The SILA algorithm is applied to longitudinal observations from several participants in different stages of amyloid accumulation spanning ~3-5 years to generate an amyloid vs. time curve. This curve can be used to accurate impute amyloid levels at earlier observations, estimate time to/from the A+ threshold, and estimate A+ onset age for individual participants.

Since the validation of SILA in amyloid PET imaging, the algorithm appears to be able to model other longitudinal observations like tau PET imaging, plasma $pTau_{217}$, and white matter hyperintensities from T2 FLAIR. While studies are ongoing to validate SILA in these applications, initial results appear promising that many neurodegenerative processes and potentially other biological processes can be modeled with this approach, and that this modeled population-level pattern can be used to estimate the time from a particular operating value (e.g., time from a biomarker positivity threshold) and estimate the age at which an individual would have reached this value.

The main outputs from the SILA algorithm are a 1) non-parametric function describing the relationship between the values being modeled and time, and 2) individual estimates of the time from the threshold and the age the person would have hit that threshold. These outputs are generated using the SILA.m and SILA_estimate functions described below. There are also some utility functions included that can be used useful for related applications like estimating the time between two different thresholds or imputing biomarker levels and times from thresholds for observations that are not concurrent with biomarker observations (described below).

2 SILA Functions

2.1 SILA.m

SILA.m is the main function used to train the SILA model on new longitudinal data. The inputs for SILA.m are described in Table 1 below, but generally SILA.m requires longitudinal observations for several participants in tall format (i.e., values to be modeled, age at those observations, and a participant/subject identifier). SILA.m calls ILLA.m and SILA_estimate.m (see below) to optimize the smoothing kernel applied prior to numeric integration and to generate the final *value vs. time* curve, whereas ILLA.m will model the data with a prespecified smoothing kernel. It is also possible to specify a smoothing kernel to SILA.m, in which case all inputs from SILA.m are passed to ILLA.m and both functions give the same output. The optimal smoothing kernel for SILA.m is determined by choosing the robust LOESS smoothing kernel that minimizes weighted residuals for backwards prediction of values (i.e., predicting the value at the first observation referencing the last observation). Because there could be an imbalance of subjects above and below the threshold (val0), residuals are weighted such that observations above and below the user-specified threshold value receive equal weight in the optimization. Once the optimal smoothing kernel is identified, the final SILA model is fitted to produce a nonparametric *value vs. time* curve, which is output as a numeric table (see Table 2). SILA also outputs a second optional table with the smoothed discrete rate sampling output describing the relationship between the annual rate of change in the value as a function of the value. Note that any subjects with a single timepoint are excluded from model training since they do not provide information about longitudinal change.

Table 1: SILA Input Data Variables

Variable	Description
age	age in years at each observation
value	observed values to be modeled with SILA
subid	a numeric subject identifier
dt	the step size for numeric integration (0.25 years works well for many biological processes)
val0	the value corresponding to the positivity threshold for the input value. This is used as an initial condition such that time = 0 years corresponds to the threshold value (val0) on the value vs. time curve.
maxi	a number indicating the maximum iterations used during numeric integration. The value to use here will depend on the step size indicated by dt (200 works well for most things; dt*maxi is the maximum duration in years that the algorithm will model)
sk (optional)	By default, SILA will optimize the smoothing kernel by minimizing the sum of squared residuals for backwards prediction of the value. The user can optionally specify a predefined smoothing kernel rather than allowing SILA to optimize this parameter. sk is a number between 0 and 1 representing the fraction of data used to smooth the rate vs. value function prior to numeric integration. Inputting this optional argument will dramatically speed up the algorithm since the smoothing kernel optimization step is not performed. If sk = 0, then no smoothing is applied.

Table 2: SILA Output Variables

Variable	Description
tsila	A table with the value vs. time curve and some additional information about the modeled curve.
tsila.val	the SILA-modeled value
tsila.time	the SILA-modeled time resulting from numeric integration without the initial condition of t=0 corresponds to a given threshold.
tsila.adtime	the SILA-modeled time resulting from numeric integration applying the initial condition that t=0 corresponds to val0 in the input.
tsila.mrate	the mean sampled rate through the value
tsila.sdrate	the standard deviation of the sampled rate through the value
tsila.nsubs	the number of subjects with observations that intersect the modeled value
tsila.sdval	an approximation of the standard deviation of the value. This is calculated by propagating the rate error through the calculation of the value.
tsila.ci95	an approximation of the 95% confidence interval of the value using tsila.sdval. Note this is likely an

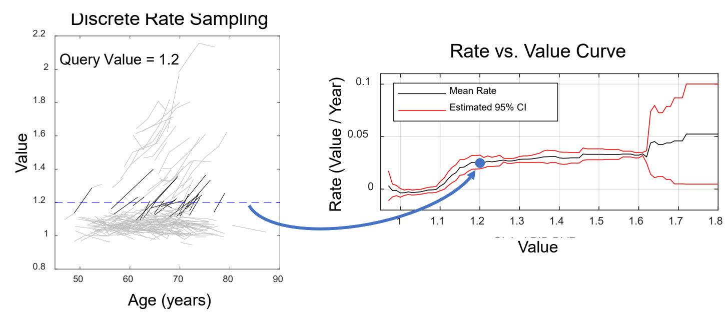
	underestimation of the model error since it does not account for temporal covariance between observations.
tdrs	A table with information about discrete rate sampling
tdrs.val	query values used for discrete rate sampling
tdrs.rate	mean rate through each query value
tdrs.ratestd	standard deviation of the rate at each query value
tdrs.npos	number of participants with positive longitudinal slopes for a given query value
tdrs.tot	number of participants with observations that intersect each query value
tdrs.ci	an approximation of the 95% confidence interval of the rate for each query value
tdrs.skern	the optimal smoothing kernel (or user defined kernel) for the rate vs. value curve. The optimization finds the smoothing kernel that minimizes sum of squared residuals for backwards prediction of values.

2.2 ILLA.m

ILLA.m is a subfunction of SILA.m that is used to generate nonparametric models of the input value vs. age data using iterative local linear approximation (ILLA). This is accomplished in two general steps: discrete rate sampling and numeric integration (i.e., Euler's Method). The inputs and outputs of ILLA.m are the same as SILA.m shown in Table 1 and Table 2. The main difference between these functions is that SILA.m performs and optimization of the smoothing kernel whereas ILLA.m will not perform this optimization and will apply whatever smoothing kernel is input. Accordingly, skern is an optional input in SILA.m with the default set such that SILA.m will automatically optimize the kernel, whereas skern is a required input for ILLA.m. Inputting a user specified smoothing kernel to SILA.m effectively bypasses this optimization and will simply pass the inputs from SILA.m to ILLA.m to train the model. Entering a smoothing kernel of 0 will result in no smoothing being applied prior to numeric integration.

Discrete Rate Sampling is used to generate the function describing the first-order relationship between the annualized rate of change as a function of the observed values. To accomplish this, 150 evenly spaced query values are generated throughout the range of observed values in the input data. For each query value, the algorithm determines which subjects' observations intersect that query value. The mean rate through the query value is then calculated across this subset of subjects by taking the mean of within-person longitudinal slopes. The mean rate at each query value is weighted such that subjects with more longitudinal observations receive a higher weight in the average rate calculation. These values and some additional outputs are stored in a table (tdrs) that is returned by the ILLA.m function. Variables in the tdrs table are described above in Table 2.

Figure 2: Discrete Rate Sampling

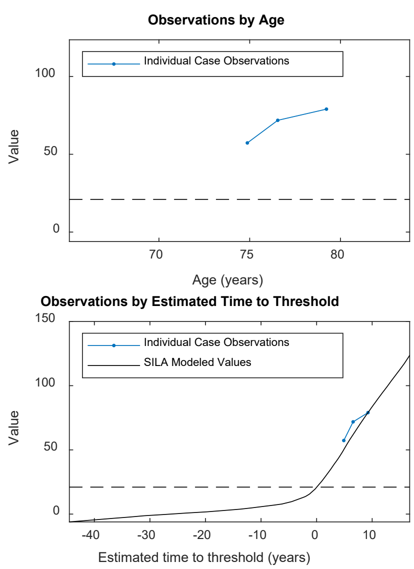


SILA uses a discrete sampling approach to generate the numeric function describing the relationship between the rate of change of the value and the value itself. In the left panel, subjects that have observations that intersect the query value of 1.2 are identified and used to calculate the mean rate through that value. This rate vs. value information is stored in a table. This process is repeated throughout the range of the observed values to form the discrete rate vs. value function that is subsequently numerically integrated.

Once the discrete rate sampling table is generated, Euler's iterative method is applied to numerically integrate the data and produce a *value vs. time* curve. This is accomplished by using the discrete rate sampling table as a lookup table at each iteration of Euler's method. If 0 < skern < 1, a Robust Loess smoothing kernel is applied to the rate vs. value curve using the MATLAB built-in smooth function prior to numeric integration. Numeric integration is initiated beginning from the mean value of the tdrs table and proceeds to integrate through time forward and backwards from this point. For each loop iteration, the current value is stored in the tsila table (along with additional information) and the current value is used in conjunction with the discrete rate sampling table to lookup the mean rate through that value. The value for the next iteration is then determined by multiplying this rate by the step size (dt) and adding this to the value of the current iteration. The iterative integration continues until one of three conditions is met: 1) the maximum number of iterations (maxi) specified by the user is exceeded, 2) there is only one subject with longitudinal data for the current iteration's value, 3) or the rate for the current iteration changes sign (i.e., a curve with a positive slope goes negative or a curve with an overall negative slope goes positive). Once numeric integration has completed, the time variable is rescaled such that 0 time corresponds to the user-specific threshold value(val0). The integrated, discrete, nonparametric function describing *value vs. time* is then output by ILLA.m as a table (tsila) with the variables described in Table 2. This table is used as the input to other functions below.

2.3 SILA_estimate.m

Figure 3: Depiction of SILA Estimate for a single subject



The SILA_estimate.m function estimates the time from threshold and related variables for individual subjects. In the case shown above, the top plot show three longitudinal observations for a single subject with the bottom plot showing that subject's observations aligned to the modeled *value vs. time* curve output from SILA. In this case, the last observation is the reference observation used to align this subject to the modeled curve and estimate time from threshold, age from threshold and impute values for previous observations.

SILA_estimate.m is used to generate subject-level estimates of time to threshold (estdtt0) and the age the subject crossed the threshold (estaget0) based on the modeled value vs. time function output by SILA.m (see Figure 3 for an example). SILA_estimate.m inputs the tsila table output from SILA.m or ILLA.m as well as subject-level data in tall format (age, value, subid; see Table 3 for details). In addition, there are optional input parameters that can be specified as name-value pairs (see Table 4). Subject-level estimates are obtained by solving the SILA-modeled value vs. time curve for time given an observed value at a reference observation or set of observations for each subject. The outputs of SILA_estimate.m are shown in Table 5. If the first or last within-subject observations are chosen as the align event (i.e., the reference observation), the value vs. time curve is solved for time given the value at the first or last observation. The estimated age at threshold (estaget0) is then calculated as age at reference observation + the estimated time duration from the threshold (estdtt0). If the 'all' option is specified for align_event and the subjecet has more than one observation, then basis functions are used to optimize the within person time-shift by finding the time shift that minimizes the within-person sum of squares to get that subject's value vs. age data to fit the modeled value vs. time curve. In addition to these time estimates, SILA_estimate.m will also impute the value at each time point (estval) based on the modeled value vs. time curve and the calculated time from threshold at each observation within a subject.

In addition to the estimated value, time from threshold, and age at threshold, additional variables are output that can be used to inform study design and analysis planning. These additional variables are described further in Table 5.

Table 3: SILA Estimate Input Variables

Variable	Description
tsila	table with the output from SILA.m containing the modeled value vs. time function
age	age of the subject at each observation in years
val	the value at each observation
subid	a numeric subject ID

optional arguments	Optional parameters can be specified as name-value pairs. These optional parameters are described below in Table 4.

Table 4: Optional name-value pairs for SILA_estimate.m

Parameter	Options and Description
align_event	This parameter allows the user to specify which observation or observations to use within a person to estimate person-level age of onset and duration of positivity. The user can enter 'first', 'last', or 'all.' The default is 'last.'
extrap_years	This parameter is used to determine how much data is used to extrapolate SILA estimates when observations fall beyond the modeled range. Linear regression is applied to the modeled relationship between the value and time, with the time duration specified by extrap_years determining how much data is used for fitting this model. Individual observations are outside of the modeled range use this linear model to extrapolate age at threshold and time from threshold. The default value is 3 years.
truncate_aget0	For data wherein rates approach zero when moving backwards in time, spurious aget0 estimates can arise with non-realistic interpretation (e.g., threshold onset age of 200 years for a human). In this case, the person-level estimates can be truncated such that the lowest individual estimate of duration from threshold is set to the earliest modeled time point on the SILA curve. The user can specify 'yes' or 'no' as a string with the default option being 'yes' to apply truncation to time to threshold estimates.

Table 5: SILA_estimate Output Variables

Variable	Description
subid	a numeric subject identifier
age	age in years at each observation
val	observed values to be modeled with SILA
minage	the minimum observed age for a subject
maxage	the maximum observed age for a subject
valt0	the time point on the value vs. time curve used to define zero time
ageref	the age at the reference observation for each subject
dtageref	the time from a given observation relative to the reference observation used for that subject. (e.g., -7 years would indicate that a particular longitudinal observation occurred 7 years before the reference observation for that subject).
estval	the SILA estimated value at each observation. Note that by definition this value is equal to the observed value at the time of the reference observation.
estaget0	the SILA estimated age the subject will cross the threshold value. (estaget0 = age at reference observation – estdtt0)

estdtt0	the SILA estimated time from the threshold for each observation for each person. estdtt0 is calculated by solving the SILA modeled value vs. time curve for adtime at the reference observation given the value at that reference observation. As such, the person-level estdtt0 equals the modeled adtime on the SILA value vs. time curve at the observed value for the reference observation.
estresid	residuals for estval at each observation. By definition, estresid = 0 at the reference observation.
estpos	a boolean determining if the SILA-estimated value for that observation is above the specified threshold valt0.
aevent	a variable indicating the alignment event specified by in the inputs in Table 4 above.
extrapyrs	a variable indicating the number of extrapolation years specified in Table 4 above.
truncated	a variable indicating whether or not estaget0 and estdtt0 were truncated for that subject.

2.4 SILA_estimate_other.m

SILA_estimate_other can be used to impute time from threshold and values for individual subjects at timepoints wherein the value being modeled may not have been observed. An example from Alzheimer's disease studies would be wanting to know how much amyloid someone had five years before the first amyloid PET scan was available. This can arise for example when a study wasn't able to collect a certain type of data from baseline and incorporated this new procedure later in the study.

SILA_estimate_other.m inputs the tsila table and the individualized sila estimates table from SILA.m and SILA_estimate.m, respectively, along with the age at which to estimate values and the subject id for each timepoint and subject (see Table 6). Age and subject ids are input in tall format if multiple timepoints are being imputed for each subject. The list of output variables for SILA_estimate_other.m is described in Table 7.

Table 6: Input Variables for SILA_estimate_other.m

Variable	Description
tsila	table with the output from SILA.m containing the modeled value vs. time function
test	table with the output from SILA_estimate.m containing the estimated ages at threshold for each subject
age	age in years at each timepoint that SILA will impute value and time from threshold
subid	a numeric subject identifier

Table 7: Output table variables for SILA_estimate_other.m

Variable	Description
subid	a numeric subject identifier
age	age in years at each observation
estdtt0	the SILA estimated time from the threshold at each observation for each person.
estval	the SILA estimated value at each observation.

dtalign	the time between the age at observation and the age at the reference observation (age_{obs} – age_{ref})
estextrap	an indicator variable that is true if the time being modeled was outside of the modeled range of the data and extrapoloation was used to estimate values and times for that observation
obsrange	A boolean indicating if the age being modeled falls within the observed ages for that subject. True = in range
dtminage	the time from the age of imputation and the minimum observed age for that subject. (age at imputation – min(observed ages))
dtmaxage	the time from the age of imputation and the maximum observed age for that subject. (age at imputation – max(observed ages))

2.5 SILA_estimate_time2val.m

SILA_estimate_time2val.m is a utility function that can be used to determine the value of the modeled time vs. value function for a user specified input time. For example, one might want to know what the modeled value is five years after the threshold is crossed. In this case, entering SILA_estimate_time2val(tsila,5) would give the modeled value five years after the threshold value was crossed.

2.6 SILA_estimate_val2time.m

SILA_estimate_val2time.m is a utility function that can be used to extract time from threshold for input values. For example, one might want to know the time interval corresponding to two different thresholds or modeled values. In this case, the user can input the modeled tsila table from SILA.m along with the two values to determine the time from threshold from each value and then subtract these time estimates to obtain the time between these operating values. (e.g., SILA_estimate_time2val(tsila,threshold1) - SILA_estimate_time2val(tsila,threshold2) = time between threshold 1 and threshold 2).

3 SILA Demo

Included in the SILA Git repository is a demonstration of a common application of SILA. This demo generates simulated longitudinal data and uses these data to show how to train a SILA model using SILA.m, and then get individualized estimates for each subject using SILA_estimate.m. To run the demo, download the code from the git repository to your local machine. Launch MATLAB and navigate to the demo directory. Open the sila_demo.m file to view the demo. Run the demo by executing sila_demo in the command window. This will add the dependent path for the SILA code and will run through the steps of simulating data, training the SILA model, and obtaining individualized time estimates.

4 Required Software and Packages

SILA currently runs in MATLAB only and has been tested in version 2021a. In addition to MATLAB and the included dependent functions, the Statistics and Machine Learning Toolbox and Curve Fitting Toolbox are also required. This code was tested running Windows 10 Professional but should also work in Mac OS and Linux MATLAB installations.

5 Citation and Funding Information

This algorithm was developed at the University of Wisconsin-Madison by Dr. Tobey Betthauser. Several individuals, grants, and data sources contributed to my ability to develop, test, and implement this algorithm. In order to ensure future development of this and other algorithms, I ask that you please include the following information in any published works that use or further develop this method:

5.1 Citation for Conference Abstracts

Abstract Text: As space and formatting permits, please cite the original publication in abstract text.
Tobey J Betthauser, Murat Bilgel, Rebecca L Koscik, Bruno M Jedynak, Yang An, Kristina A Kellett, Abhay Moghekar, Erin M Jonaitis, Charles K Stone, Corinne D Engelman, Sanjay Asthana, Bradley T Christian, Dean F Wong, Marilyn Albert, Susan M Resnick, Sterling C Johnson, for the Alzheimer's Disease Neuroimaging Initiative, Multi-method investigation of factors influencing amyloid onset and impairment in three cohorts, *Brain*, 2022;, awac213, https://doi.org/10.1093/brain/awac213

Slide/Oral presentations: Please include an in-slide callout to the published paper as space permits (e.g., "Betthauser, et al., Brain. 2022" or "Betthauser, et al., Multi-method investigation of factors influencing amyloid onset and impairment in three cohorts. Brain. 2022").

5.2 Citation for Peer-Reviewed Publication

Please include a link to the git repository in the methods section of the main body text (e.g., *"Sampled iterative local approximation (SILA; https://github.com/Betthauser-Neuro-Lab/SILA-AD-Biomarker) was used to model…"*).

Please cite the following paper in the methods section as appropriate:

Tobey J Betthauser, Murat Bilgel, Rebecca L Koscik, Bruno M Jedynak, Yang An, Kristina A Kellett, Abhay Moghekar, Erin M Jonaitis, Charles K Stone, Corinne D Engelman, Sanjay Asthana, Bradley T Christian, Dean F Wong, Marilyn Albert, Susan M Resnick, Sterling C Johnson, for the Alzheimer's Disease Neuroimaging Initiative, Multi-method investigation of factors influencing amyloid onset and impairment in three cohorts, *Brain*, 2022;, awac213, https://doi.org/10.1093/brain/awac213

5.3 Study and Funding Acknowledgements

This algorithm was developed and validated for amyloid PET imaging using data from the Wisconsin Registry for Alzheimer's Prevention (WRAP; PI: Sterling Johnson) and the Baltimore Longitudinal Study of Aging (BLSA; PI: Susan Resnick) under the Preclinical Alzheimer's Consortium (PAC; PI: Marilyn Albert), and data from the Alzheimer's Disease Neuroimaging Initiative (ADNI; PI: Michael Weiner).

The following funding sources contributed to the development of this algorithm: NIH R01 AG021155, R01 AG027161, P50 AG033514, U54 HD090256, S10 OD025245, R01 AG054047, RF1 AG059869; Alzheimer's Association AARF-19-614533.

The following are acknowledgements from the ADNI data use agreement: This work was supported in part by the Intramural Research Program of the National Institute on Aging, National Institutes of Health. Data collection and sharing for this project was funded by the ADNI (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.;

Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.