Europe PMC Funders Group
Author Manuscript
Nat Protoc. Author manuscript; available in PMC 2022 September 27.

Published in final edited form as:
Nat Protoc. 2022 July 01; 17(7): 1711–1734. doi:10.1101/2021.08.08.455583.

The Normative Modeling Framework for Computational Psychiatry

Saige Rutherford[1,2,3], **Seyed Mostafa Kia**[1,2,4], **Thomas Wolfers**[5,6], **Charlotte Fraza**[1,2], **Mariam Zabihi**[1,2], **Richard Dinga**[1,2], **Pierre Berthet**[5,6], **Amanda Worker**[7], **Serena Verdi**[8,9], **Henricus G. Ruhe**[10,*], **Christian F. Beckmann**[1,2,11,*], **Andre F. Marquand**[1,2,7,*]

[1]Donders Institute for Brain, Cognition, and Behavior, Radboud University, Nijmegen, the Netherlands

[2]Department of Cognitive Neuroscience, Radboud University Medical Center, Nijmegen, the Netherlands

[3]Department of Psychiatry, University of Michigan, Ann Arbor, MI, United States

[4]Department of Psychiatry, Utrecht University Medical Center, Utrecht, the Netherlands

[5]Department of Psychology, University of Oslo, Oslo, Norway

[6]Norwegian Center for Mental Disorders Research, University of Oslo, Oslo, Norway

[7]Department of Psychological Medicine, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, United Kingdom

[8]Centre for Medical Image Computing, Medical Physics and Biomedical Engineering, University College London, London, UK

[9]Dementia Research Centre, UCL Queen Square Institute of Neurology, London, United Kingdom

[10]Department of Psychiatry, Radboud University Medical Center, Nijmegen, the Netherlands

[11]Centre for Functional MRI of the Brain, University of Oxford, Oxford, United Kingdom

Abstract

Normative modeling is an emerging and innovative framework for mapping individual differences at the level of a single subject or observation in relation to a reference model. It involves charting

This work is licensed under a CC BY 4.0 International license.

Correspondence to: Saige Rutherford.

Corresponding author: Saige Rutherford saige.rutherford@donders.ru.nl.

*Contributed equally to senior author

Tweet New protocol by @being_saige for normative modeling analysis using the Predictive Clinical Neuroscience toolkit (PCNtoolkit) - an emerging and innovative framework for mapping individual differences at the level of a single subject or observation in relation to a reference model, moving computational psychiatry analyses away from noisy case-control comparisons.

Author Contributions
Conceptualization: SR, SMK, TW, CF, MZ, RD, PB, AW, SV, HGR, CFB, AFM; Methodology: SR, SMK, TW, CF, MZ, RD, AFM; Data Curation: SR, AFM; Writing – Original Draft: SR; Writing – Reviewing and Editing: SR, SMK, TW, CF, MZ, RD, PB, AW, SV, HGR, CFB, AFM; Visualization: SR; Supervision: HR, CFB, AFM; Funding Acquisition: HR, CFB, AFM

Conflict of Interests
CFB is director and shareholder of SBGNeuro Ltd. HGR received speaker's honorarium from Lundbeck and Janssen. The other authors report no conflicts of interest.

Europe PMC Funders Author Manuscripts
Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

centiles of variation across a population in terms of mappings between biology and behavior which can then be used to make statistical inferences at the level of the individual. The fields of computational psychiatry and clinical neuroscience have been slow to transition away from patient versus “healthy” control analytic approaches, likely due to a lack of tools designed to properly model biological heterogeneity of mental disorders. Normative modeling provides a solution to address this issue and moves analysis away from case-control comparisons that rely on potentially noisy clinical labels. In this article, we define a standardized protocol to guide users through, from start to finish, normative modeling analysis using the Predictive Clinical Neuroscience toolkit (PCNtoolkit). We describe the input data selection process, provide intuition behind the various modeling choices, and conclude by demonstrating several examples of down-stream analyses the normative model results may facilitate, such as stratification of high-risk individuals, subtyping, and behavioral predictive modeling. The protocol takes approximately 1-3 hours to complete.

Keywords

normative modeling; computational psychiatry; individual differences; precision medicine; software tutorial; lifespan neuroscience; brain growth charting

Introduction

Clinical neuroscientists have recently acknowledged two realities that have disrupted the way research is conducted: first, that to understand individual differences it is necessary to move away from group average statistics [1–7] and, second, that the classical diagnostic labels of psychiatric disorders are not clearly represented in the underlying biology[8–11]. Initiatives such as RDoC[9,12,13], HiTOP[10,14,15], and ROAMER[16,17] were established in response and seek to refine the nosology of mental disorders by mapping biobehavioral dimensions that cut across heterogeneous disorder categories. Despite this awareness and an increasing interest in quantifying individual differences, the field has still been slow to transition away from case-control comparisons that aim to contrast patient versus healthy control groups and assume that clinical groups are distinct and homogenous. A key barrier that has impeded progress is a lack of alternative analysis methods, designed to model variation across individuals, also known as heterogeneity[18]. Nearly all existing techniques for connecting the brain to behavior operate at the group-level and provide no path to individual-level inference[19–21]. Normative modeling is a framework for understanding differences at the level of a single subject or observation while mapping these differences in relation to a reference model (Figure 1). It involves charting centiles of variation across a population in terms of mappings between biology and behavior, which can then be used to make statistical inferences at the level of the individual, akin to the use of growth charts in pediatric medicine (Figure 1A). The practice of normative modeling in clinical neuroscience was developed to provide additional information beyond what can be learned from case-control modeling approaches (see ‘Development of the Protocol’ section below for further information). Case-control thinking assumes that the mean is representative of the population, when it may not be (e.g., if the clinical population is diffuse or comprised of multiple sub-populations). Therefore, normative modeling has become a leading tool for

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

precision medicine research programs and has been used in many clinical contexts[22] (see 'Applications' section below for further examples).

Neuroscience has historically brought together scientists from diverse educations, for example, some from a clinical background and others having a mathematics background. The interdisciplinary nature introduces a challenge in bridging the gap between technical and clinical perspectives. This is a key challenge that aligns with the aims of the open-science movement and brain-hack community[23], in other words, to distill the essential components of the analytic workflow into a consistent and widely applicable protocol. This helps to avoid 'research debt', i.e., a lack of ideas being digested[24]. This distiller mindset is crucial for confronting research debt and embracing paradigm shifts in thinking, such as moving from case-control comparisons to the normative modeling framework.

The purpose of this work is to distill the methods of normative modeling, an advanced analysis technique, into an actionable protocol that addresses these challenges in that it is accessible to researchers within the diverse field of clinical neuroscience. We distill the essential components of a normative modeling analysis and provide a demonstrative analysis from start to finish using the Predictive Clinical Neuroscience Toolkit software. We describe the input data selection process, give an overview of the various modeling choices, and conclude by demonstrating several examples of downstream analyses the normative model results may facilitate, such as stratification of high-risk individuals, subtyping, and behavioral predictive modeling.

Development of the protocol

Normative modeling has a long history that relates to statistics and measurement theory and has many applications from medicine to economics to neuroscience. Familiar use cases of normative modeling include growth charting in pediatrics, neurocognitive tests, and interpreting graduate school test score percentiles (i.e., scoring 90th percentile on the MCAT). The mathematical and computational development of normative modeling has been fine-tuned[25–28] and currently exists as an open-source software python package, the Predictive Clinical Neuroscience toolkit (PCNtoolkit), which we focus on in this manuscript. This toolkit implements many commonly used algorithms for normative modelling and supports multiple industry standard data formats (e.g., NIFTI, CIFTI, text formats). Extensive documentation has been written to accompany this protocol and is available online through read the docs. This includes tutorials with sample data for all algorithm implementations, a glossary to help new users understand the jargon associated with the software, and a frequently asked questions page. An online forum for communicating questions, bugs, feature requests, *etc*. to the core team of PCNtoolkit developers is also available. We have developed these open-source resources to promote and encourage individual differences research in computational psychiatry using normative modeling.

Applications and comparison with other methods

Normative modeling has been applied to many research questions in computational psychiatry and other fields, including in autism spectrum disorder[29–31], attention deficit

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

hyperactive disorder[32,33], Alzheimer's disease[34], bipolar disorder, and schizophrenia[35–37]. Crucially, these applications have shown that normative modelling can detect individual differences both in the presence of strong case-control differences (observed in schizophrenia)[36] and in their absence (observed in autism spectrum disorder)[30]. This highlights the value and complementary nature of understanding individual variation relative to group means. These applications have primarily focused on predicting regional structural or functional neuroimaging data (i.e., biological response variables) from phenotypic variables (i.e., clinically relevant covariates) such as age and sex. Age creates a natural, time-varying dimension for mapping normative trajectories and is well suited to applications in which deviations of an individual manifest from a typical trajectory of brain development or ageing. However, other phenotypes that have been used in neuroimaging predictive modeling studies such as general cognitive ability[38,39], social cognition, or sustained attention[40,41] are also attractive possibilities to use as covariates, thereby defining axes for observing deviation patterns. Normative modeling has also been used to learn mappings between reward sensitivity and reward related brain activity[42].

It is important to emphasize that normative modeling is a general regression framework for mapping sources of heterogeneity, refocusing attention on individual predictions rather than group means (e.g., diagnostic labels), and detecting individuals who deviate from the norm. Therefore, it is not limited to a specific algorithm or mathematical model, although we recommend certain algorithms based on the research question and available input data. The algorithms in the PCNtoolkit tend to favor Bayesian over frequentist statistics, as there are certain features of Bayesian approaches that facilitate better normative modeling estimation. For example, having a posterior distribution over the parameters help to better separate different sources of uncertainty, e.g., separating variation ('aleatoric uncertainty' – cannot be reduced by adding more data) from modeling (or 'epistemic') uncertainty which can be reduced by adding more data. These different use cases of normative modeling (algorithm selection, predicting brain from behavior or behavior from the brain) are explained in-depth in the 'Experimental Design' section, below.

There is a long history of using regression methods to learn mappings between brain and behavior[43,44]. Principle Component Regression (PCR)[20,45,46], Connectome predictive modeling (CPM)[19,47], and canonical correlation analysis (CCA)[48,49] have become mainstream methods for linking brain and behavior. These methods have demonstrated the feasibility of brain-behavior mapping and laid the foundation for individual differences research to thrive. While these approaches have generated much curiosity and excitement, they are limited in their ability to provide inference at the level of the individual, providing only point estimates (i.e., without associated centiles of variation). Most papers using these tools only report the mean predictive model performance, collapsing information across hundreds or thousands of people into a single number (e.g. model accuracy or regression performance)[46,47,50,51]. The normative modeling framework takes these ideas a step further to quantify and describe how individuals differ statistically, with respect to an expected pattern. In this way, normative modelling breaks the symmetry inherent in the case control paradigm. In more detail, PCR and CPM differ from normative modeling in terms of how the prediction model is formulated. PCR and CPM setup the regression model such that, Y, a n_subjects x 1 vector (i.e., age or fluid intelligence), is predicted from X, a matrix

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

with n_subjects x n_brain dimensions, where n_brain is typically a reduced feature space selected via a regularization step. This setup makes interpretating which brain features are related to the behavior very challenging. Studies using PCR or CPM attempt to interpret the brain feature weights, but as these methods typically use fMRI connectomic data, consisting of connections and nodes, interpretation often yields a complex whole-brain visualization that is not very informative. The individual-level output of these models is a single point estimate, a predicted behavior score for each subject. These individual point estimates are then summarized by correlating the predicted and true behavior scores, reporting explained variance (R_2), and calculating accuracy (mean squared error). Compared to PCR and CPM, normative modeling inverts the regression setup around to predict brain region Y, a n_subjects x 1 vector from X, a matrix with n_subjects x n_covariates (i.e., age, sex, fluid intelligence, site, data quality metric). There is a separate regression model for each brain region. The individual-level outputs of normative modeling are the predicted brain score, the predictive variance (separated into modeling and noise components), a deviation score (Z-score, how much each subject deviate from the normative range). The overall performance is evaluated by correlating predicted and true values, calculating explained variance, standardized mean squared error, and mean standardized log loss. In contrast, CCA estimates a doubly multivariate relationship in that both X and Y are matrices (X is n_subjects x n_brain matrix and Y is n_subjects x n_behavior). Whilst CCA is well suited to detecting that a mapping exists, this still leads to difficult interpretation of feature importance and, moreover, CCA is highly prone to overfitting and requires careful assessment of out of sample metrics with respect to an appropriate null distribution, which is not always done in practice. Like PCR and CPM, CCA also does not provide individual measure of uncertainty or deviation scores.

Case-control inference (e.g., mass univariate group t-testing and classification of patient vs. control) examples are perhaps the most interesting comparison to the normative modeling framework. Case-control methods typically require there to be a homogeneous within-group spatial signature and their success relies on obtaining statistical significance (p-value < 0.05). We clarify this point with an example of the assumptions of case-control inference. To detect a group difference in amygdala activation between a control group and a group of individuals with post-traumatic stress disorder during a fMRI task, all individuals in the control group need a similar value of amygdala activation and all individuals in the PTSD group need a similar value of amygdala activation. Then, the mean amygdala activation signal of the control group must be statistically different from the mean amygdala activation signal of the PTSD group after stringent multiple comparison testing correction. These assumptions ignore the fact that different biological processes (i.e., some people have increased activation and others decreased) can lead to similar external behavior. Normative models reveal a different side of the data -- that the classical diagnostic labels of psychiatric disorders are not clearly represented in the underlying biology, meaning patient groups are not well defined by a unifying neurosignature -- and provide clear evidence for the limitations of case-control paradigms. Brain age models are also in the same family as normative models but generally have a narrower focus on interpreting accelerated/ decelerated aging[52,53] or improving prediction accuracy[54]. Brain age models only allow for

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

interpreting centiles of variation in terms of age, which is limited and does not have a clear interpretation in terms of biological variation across individuals.

“All models are wrong, but some are useful” -- George E.P. Box.

There is not one ‘best’ modeling approach and many of the methods presented in this section can be complimentary in that they investigate different questions. Before embarking on a computational modeling journey, it is always important to ask questions such as: What are the assumptions made by this model? What type of inference do you want to make (group-level, individual-level)? What aspect of the predictive model is most important (accuracy, quantifying uncertainty, statistical significance)? Allow your research question to guide the answers and model selection.

Expertise needed to implement the protocol

We aimed to make this protocol user-friendly to the diverse community of neuroscience, including those with a non-technical background. The fundamental objective of this protocol is to learn how to implement the normative modeling framework via the PCNtoolkit software without being an expert in statistics and machine learning. You will be given enough knowledge to set up training and test sets, understand what data should be going into the model, interpret results, and make inferences based on the results. Prerequisites of this protocol are basic familiarity with the Python programming language and a computer with a stable internet connection. Complete code, example data, and extensive documentation accompany this protocol; thus, writing code from scratch is unnecessary. Of course, it is our intention for readers to be inspired by this protocol and to use the normative modeling framework in more ways than presented here. If you wish to use the framework presented in this protocol beyond the provided code, familiarity with the Linux command line, bash scripting, setting up virtual environments, and submitting jobs to high-performance clusters would also be helpful.

Limitations

Big data requires automated QC

As datasets grow to meet the requirements of becoming population-level or big data, there is typically a need to rely on automated quality control metrics[55]. This means there is potential to unintentionally include poor quality data, which could, in turn, affect the results. The training and test dataset used in this protocol has been manually quality checked by visualizing every subject’s raw T1w volume with their corresponding Freesurfer brain-mask as an overlay using an online (JavaScript-based) image viewer. Quality checking code and further instructions for use is made available on GitHub. These images were inspected for obvious quality issues, such as excess field-of-view cut-off, motion artifacts, or signal drop-out. Subjects that were flagged as having quality issues were excluded from the sample. Users should consider manually quality checking their own data if they wish to add on additional samples to the dataset.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Multi-site confounds and data availability

Pooling data from multiple sites is often a necessary step to create diverse datasets and reach sufficient sample sizes for machine learning analyses. When combining data from different studies, several challenges arise. First, there are often different MRI scanners at each site that also have different acquisition parameters. These MRI hardware and software divergences give rise to substantial nuisance variance that must be properly accounted for when modeling the data. Second, there may be sampling differences, for example due to different inclusion criteria and definitions of diagnostic labels at each site. For example, if one site uses the Structured Clinical Interview for DSM-5 (SCID-5) administered by a trained mental health professional who is familiar with the DSM-5 diagnostic criteria, while another site relies on self-report questionnaire data to define clinical labels. This increases the heterogeneity within the clinical groups (e.g., by mixing inclusion criteria across cohorts) and could also add noise to the diagnostic labels (e.g., if diagnostic assessments have different reliability across studies). This is important to consider if clinical labels are used to separate data into training and testing sets (i.e., controls only in the training set) and when comparing the outputs of normative modeling across patient groups from different sites.

Furthermore, there is likely to be dissimilarities in the available demographic, cognitive, and clinical questionnaire data across sites as well which needs to be considered when deciding which studies to include and which covariates to use in modeling. If the goal is to share the model, allowing transfer to new samples, using unique covariates that are specific to your sample (and not commonly collected) will hinder the ability of others to use your model on their own data. There is a careful balance that should be considered regarding the benefits gained from a new site joining the sample versus the site related nuisance variance that accompanies the addition of new sites.

Overview of the procedure

Experimental design

There are many choices and considerations that should be carefully planned before embarking on a normative modeling analysis – the decision points can be grouped into the following stages: data selection, data preparation, algorithm/modeling, and evaluation/interpretation. These stages, and the corresponding step numbers of the procedure are summarized in Figure 2. There are additional resources and support for running normative modeling analysis that are summarized in Figure 3.

Data selection – reference cohort inclusion criteria—Creating the training dataset that will serve as the “normative” reference cohort is the first important decision. Ideally, the training dataset will be a large and representative sample, and the included subjects should not be missing vital demographic (age, sex) or biological (neuroimaging) data. However, data imputation may be used if necessary but should be used cautiously. In most research studies, data are missing not at random, and we interpret more than just mean effects. In this case, mean imputation may bias results and other forms of imputation

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

should be considered[56,57]. It is important that the reference cohort provides good coverage (complementary covariates) of the test set (e.g., clinical) population.

The sample size of the reference cohort (training set) is important to consider in normative modeling, although we emphasize that the focus is different to classical power calculations, which target a fundamentally different question (i.e., determining a required sample size to detect a group level comparison of a specified effect size at a given significance level). In contrast, in normative modeling, the focus is usually on quantifying deviations from a reference model at the individual level. In this context, the size of the reference cohort primarily influences the test set deviation scores by influencing the accuracy and precision with which the target phenotype (i.e., response variable) can be predicted. As the sample sizes increase, the predictive intervals will shrink, which results in an increased sensitivity to detect individual differences. However, there is not a specific cutoff that represents an ideal sample size, and we emphasize that context is key. For instance, you could build a clinical normative model for a sample of individuals with major depression disorder (MDD) for the purpose of stratification or detecting subgroups (individuals who have recurrent episodes or individuals who do not respond to medication). In this case, the reference cohort might consist of individuals that have experienced single MDD episodes and those that have responded well to medication. The sample size for this normative modeling research question would likely be relatively small due to data availability (e.g., clinical datasets typically have stricter data sharing requirements). The main takeaway from this MDD example is that sample size is highly dependent on the research question which in turn guides the inclusion criteria for the reference cohort you want to measure deviations from. If you are modeling “healthy” lifespan populations, the sample size will likely be large (on the order of thousands) because of the plethora of publicly shared data that can be leveraged. On the other hand, if you want to model a specific clinical population or a specific functional task, the sample size will be smaller due to availability of data. A smaller dataset that properly addresses the research question at hand is completely acceptable.

Data selection - covariate selection—The next choice should be regarding which covariates to include. One of the main criteria to include a covariate is the relevance to the posed research question. In normative modeling, usually we are interested in studying the deviations from the norm of the population, in other words, we are more interested in residuals. Thus, when we include a covariate in the design matrix for estimating the normative model, we are mainly interested in removing its effect from the residuals (thus deviations) than investigating its effect on the neuroimaging variable. Normative modeling is a tool to study unknowns (that are encoded in the deviations). To do so, we need to first account for known variation in the data by regressing them out of the data (thus we include the knowns in the covariates), and then we interpret the residual variation in the deviation scores. For example, if you want to know the effect of smoking on the ventral striatum, that is not confounded by other substance use, you should include substance use variables (e.g., drinks per week, etc.) in the covariate matrix, estimate the normative model, and then correlate the ventral striatum deviation score (that has the effect of drinking removed from it) with smoking frequency. When pooling data from multiple sites, the available measures across sites may influence the selection of covariates because ideally,

Europe PMC Funders Author Manuscripts

the variables should be consistent across sites. For example, you should not use different versions of a cognitive test, as they could test for different dimensions of general cognitive ability. For neurodevelopmental or lifespan model, the suggested minimum covariates to include are age, sex, site (using random- or fixed-effects), and optionally a metric of data quality (i.e., mean framewise displacement or Freesurfer Euler number). Modeling site is very important; however, an exhaustive explanation is outside the scope of this protocol but see [20,21] for an in-depth account of modeling site variation. Diagnostic labels could also be included as covariates to utilize the variance explained by these labels without constraining the mapping to only reflect case-control differences. Furthermore, additional biological covariates could also be included, for example blood biomarkers, or structural brain measures if predicting functional brain measures. Additional or alternative covariates may include other demographics (race, ethnicity, gender, education level, marital status, household income) and cognitive variables.

Data Selection – MRI modality and spatial resolution of brain data—Next, it is necessary to decide on the modality of brain imaging to model. In this protocol, we use cortical thickness and subcortical volume measurements from structural MRI (T1-weighted) images. However, other modalities such as resting-state and task-based functional MRI or diffusion weighted MRI could also be selected in this step (data for these modalities is not provided with this protocol). The resolution of brain data is important to consider while keeping in mind the increasing computational complexity with modeling smaller units. Vertex or voxel-level modeling of brain data provides high-resolution deviation maps. Still, region of interest (ROI) level modeling may allow for easier interpretation/visualization of the output deviation maps and will have a lower penalty in multiple comparison correction (if doing post-hoc analysis) on the deviation maps. The PCNtoolkit can run models in parallel to speed up computation time; however, there is still a univariate nature, meaning a separate model is fit for each brain region. This univariate approach does not address the spatial autocorrelation[58–65] or functional heterogeneity (functional mis-registration) present in (f)MRI data[66]. Spatial autocorrelation refers to the complex spatial correlation patterns present in MRI data. Nearby regions are often more correlated than distant regions, thus they are not statistically independent. Spatial correlations are difficult to model due to their heterogeneity, complexity and high dimensionality with a limited sample size. Techniques such as Markov Random Fields[62,63], network/graph theory (topology)[58,59,64,65], and spatial Bayesian latent factor methods[61,67] have been applied to address the problem of spatial autocorrelation in raw or preprocessed MRI data. Progress in addressing spatial autocorrelation in the context of normative modeling has also been made in which Kronecker algebra and low rank approximations are used to build multivariate normative models[68,69]. In the context of normative modeling, we recommend paying extra attention to image registration in order to properly model the functional regions, as the spatial overlap of regions across individuals is not guaranteed with functional areas. In addition to taking extra measures to align the fMRI data, rather than modeling single voxels or parcels (as is often done in structural MRI), it may be beneficial to model brain networks as these features better capture the spatial patterns of functional units.

Europe PMC Funders Author Manuscripts

Data preparation – Preprocessing and quality checking—Example data has been curated and shared for the purposes of this protocol. As mentioned in the data selection – MRI modality section above, we use structural MRI and have run Freesurfer to extract cortical thickness and subcortical volume measures. If using other data than the provided protocol data (i.e., your own data) then you will be to preprocess it accordingly and quality check the data to ensure only high-quality data is included. If you are new to working with MRI data, we recommend Andy's Brain Book[70] that includes videos and code tutorials for most neuroimaging software (i.e., Freesurfer, FSL, SPM).

Data preparation – Setup computational environment—In this stage you will create a python virtual environment and install the required python packages. Then you will clone the GitHub repository which contains all the code and data required to follow along with the procedure section. You can run the entire protocol in the cloud using Google Colab or chose to run the code on your own computer or server.

Data preparation – Format design matrix (site effects)—It is rare for a single scanning site to acquire large enough samples that are an accurate representation of the general population. Therefore, it is common to pool data obtained across multiple MRI centers. Some projects, such as the ABCD study[71], have begun to harmonize scanning protocols because multi-site pooling was planned prior to data collection. In contrast, other projects, such as ENIGMA[72], combine data post-collection and not have harmonized scanning sessions prior to data collection. If possible, to eliminate additional sources of variance, multi-site pooled data should be preprocessed using identical pipelines and software versions. However, due to data sharing restrictions and privacy concerns regarding health data, raw data may be unavailable, making pre or post data collection harmonization efforts impossible. Data harmonization techniques, such as COMBAT[73–76], aim to remove site-related variance from the data as a preprocessing step before further analyses are run. There are some issues with harmonization, principally that all sources of variance that are correlated with the batch-effects (i.e., site-related variance) are removed which can unintentionally remove important, unknown, clinically relevant variance from the data. COMBAT also requires that the user have access to all the data when harmonizing which may have implications for data privacy. We therefore do not recommend users focus on data harmonization techniques when preparing their data sets for normative modeling. Hierarchical Bayesian Regression (HBR)[27,28] implemented in the PCNtoolkit has been thoroughly developed and tested to address these challenges when using multi-site data in normative modeling. HBR estimates site-specific mean effects and variations in the normative model estimation stage using a Bayesian hierarchical model, which produces site-agnostic deviation scores (z-statistics). This distinction between harmonization techniques (i.e., COMBAT) and HBR-normative modeling is very important when using deviation scores as features in subsequent interpretation analyses, as harmonization has been shown to overexaggerate confidence in downstream analyses[77].

Data preparation - Train-test split—Whilst there are no hard rules for selecting the relative proportion of training and test data, some general guidelines that may help this decision can be considered. On the one hand, it is important to ensure the training set be

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

sufficiently large to model the target phenotype with sufficient accuracy and precision. On the other hand, ensuring the test set is not too small is also important to provide sufficient sensitivity to detect downstream differences (which may depend on the expected frequency of clinically relevant deviations in the test set). In practice a 70% train, 30% test or 80% train, 20% test split often provides a reasonable balance between these competing objectives, but in certain applications it may be necessary to deviate from these recommendations. The main purpose of the train-test split is to establish out of sample generalizability and whether there is over (or under) fitting occurring. More important than the exact ratio of the train/test split, we believe it is critical to focus on preserving the sample characteristics across the train/test split. For example, it would not be sensible to model age ranges of childhood and adolescence in the reference cohort and have the test cohort consist of late adulthood ages. This scenario would detect high deviations in this test set due to not properly modeling the target population. If you want to investigate the hypothesis that a certain clinical group (e.g., individuals with a psychosis diagnosis) have more extreme deviation patterns than a control group (individuals with no psychiatric diagnosis), you need to verify that it is because they are patients not because they are in the test set. In order to verify this, it is important to also include some controls (from the same imaging site as the patients) in the test set. In other words, you cannot separate site variation from diagnostic variation if you do not have control reference data.

The train-test ratio decision naturally relates to the sample size requirements of the reference cohort mentioned in the data selection – reference cohort inclusion criteria section above, and the same consideration of the context needs to be taken when creating the train-test split. Does this split align with the research question being asked? More specifically, does the training set adequately match the reference ("normative") cohort and does the testing set represent the target cohort in which deviations (from the reference cohort) will be interpreted? We discourage cross validation, or iteratively resampling of the data set into train and test sets, unless the dataset is very small, and if it is used then practitioners should be aware of the problems it introduces. Ideally, the train/test split of the dataset will only be done once. While cross validation is useful for testing stability and sensitivity of models to perturbations, it also leads to having multiple models which are not easy to combine and interpret and it induces dependence between folds which violates most parametric statistical tests[78].

Algorithm & Modeling - algorithm selection—After the data have been carefully chosen and curated, it is time to move onto the normative modeling implementation. There are several algorithms for implementing a normative model including Gaussian process regression[79], Bayesian linear regression[25,67], hierarchical Bayesian regression[27,28], generalized additive models of location, scale, and shape[26], neural processes[68], random feature approximation[80], quantile regression[81] and many of these are implemented in the PCNtoolkit software package (Table 1). The algorithms have different properties depending on their ability to model non-linear effects, scaling to large data sets (in terms of computation time), handling of random or fixed effects (e.g., to model site effects), their ability to model heteroscedastic or non-Gaussian noise distributions and their suitability for use in a federated or decentralized learning environment. An overview of these algorithm

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

implementations is covered in Table 1. Gaussian process regression (GPR) was widely used in the beginning phases of normative modeling, which can flexibly model non-linear effects but does not computationally scale well when the training data increases (i.e., beyond a few thousand data points). In this work, we focus on Bayesian linear regression (BLR), which is highly scalable (fast compute time with large samples) and flexible (can be transferred to new sites not included in the training sample and can be combined with likelihood warping to model non-Gaussian effects). Hierarchical Bayesian regression (HBR) is another appealing choice as it has been used to better address multi-site datasets and allows for transfer learning (e.g., prediction for unseen sites) and can be estimated in a federated learning framework which is useful if there are privacy concerns and/or sharing restrictions meaning data cannot easily be pooled at a single computing site.

The remaining steps including estimating the normative model, evaluation the model performance, interpretating the model fit, and ideas for post-hoc analysis of the normative modeling outputs are covered in more detail in the protocol section.

Materials

Equipment

- Computing infrastructure: a Linux computer or HPC (SLURM or Torque) with enough space to store the imaging data of the train and test set.
  - If a Linux computer or server is unavailable, this protocol can also be run in Google Colab (for free). If using Google Colab, only a computer with an internet connection and modern internet browser (e.g., Chrome or Firefox) installed is necessary.
- Python installation (https://www.python.org/downloads/).
  - Recommended: Anaconda or virtual environment to manage the required python packages (https://www.anaconda.com/ or https://virtualenv.pypa.io/en/latest/).
- PCNtoolkit python package version 0.20 (and dependencies) installed via pip (https://pcntoolkit.readthedocs.io/en/latest/pages/installation.html).
- Covariates and response variables. Examples of these are provided with this protocol.
  - Demographic and behavioral data used as predictor variables
    - ♦ Age, sex/gender, site/scanner ID, race/ethnicity, cognition, data quality metric (Euler number if structural, mean framewise displacement if functional)
  - Biological data to be modeled. An example structural MRI dataset is provided with this protocol.
    - ♦ Structural MRI: cortical thickness, surface area, subcortical volume

- Functional MRI: parcellated task activation maps, resting-state networks

Procedure

The data selection stage (Figure 2, panel 1) does not require code, as it is more of a research question formulation stage (i.e., choosing inclusion criteria and what type of imaging modality to model). Data preprocessing (running Freesurfer) and quality checking have also already been performed, and code for running Freesurfer or other preprocessing is not included in this protocol. Thus, for this protocol, the procedure begins at the *data preparation – setting up computational environment stage*. See the experimental design section for guidance on the data selection stage and preprocessing if using different data than what is provided with the protocol.

Data Preparation: Prepare computational environment

Timing: 1-3 minutes—1. Begin by cloning the GitHub repository using the following commands. This repository contains the necessary code and example data. Then install the python packages using pip and import them into the python environment (either Google Colab or using a local python installation on your computer), as follows:

```
git clone https://github.com/predictive-clinical-neuroscience/PCNtoolkit-demo.git
# set this path to the git cloned PCNtoolkit-demo repository --> Uncomment whichever line you need for either running on your own computer or on Google Colab.
#os.chdir('/Users/saigerutherford/repos/PCNtoolkit-demo/') # if running on your own computer, use this line (change the path to match where you cloned the repository)
#os.chdir('PCNtoolkit-demo/') # if running on Google Colab, use this line
import os
pip install -r requirements.txt
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import joypy
from sklearn.model_selection import train_test_split
from pcntoolkit.normative import estimate, evaluate
from pcntoolkit.utils import create_bspline_basis, compute_MSLL
```

Data Preparation: Prepare covariate data

Timing: 5-8 minutes—2. The data set (downloaded in Step 1) includes a multi-site dataset from the Human Connectome Project Young Adult study, CAMCAN, and IXI. It is also possible to use different datasets (i.e., your own data or additional public datasets) in this step. If using your own data here, it is recommended to load the example data to view

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

the column names in order to match your data to this format. Read in the data files using pandas, then merge the covariate (age & sex) data from each site into a single data frame (named cov). The columns of this covariate data frame represent the predictor variables. Additional columns may be added here, depending on the research question.

```
hcp = pd.read csv('data/HCP1200 age gender.csv')
cam = pd.read_csv('data/cam_age_gender.csv')
ixi = pd.read csv('data/IXI age gender.csv')
cam hcp = pd.merge(hcp, cam, how='outer')
cov = pd.merge(cam_hcp, ixi, how='outer')
sns.set(font scale=1.5, style='darkgrid')
sns.displot(cov, x="age", hue="site", multiple="stack", height=6)
cov.groupby(['site']).describe()
```

Data Preparation: Prepare brain data

Timing: 10-15 minutes—3. Next, format and combine the MRI data using the following commands. The example data contains cortical thickness maps estimated by running recon-all from Freesurfer (version 6.0). The dimensionality of the data was reduced by using ROIs from the Desikan-Killiany atlas. Including the Euler number as a covariate is also recommended, as this is a proxy metric for data quality. The Euler number from each subject's recon-all output folder was extracted into a text file and is merged into the cortical thickness data frame. The Euler number is site-specific, thus, to use the same exclusion threshold across sites it is important to center the site by subtracting the site median from all subjects at a site. Then take the square root and multiply by negative one and exclude any subjects with a square root above 10.

```
cam = pd.read_csv('data/CAMCAN_aparc_thickness.csv')
hcpya = pd.read_csv('data/HCP1200_aparc_thickness.csv')
ixi = pd.read_csv('data/IXI_aparc_thickness.csv')
hcpya_cam = pd.merge(hcpya, cam, how='outer')
brain_all = pd.merge(ixi, hcpya_cam, how='outer')
hcp_euler = pd.read_csv('data/hcp-ya_euler.csv')
cam_euler = pd.read_csv('data/cam_euler.csv')
ixi_euler = pd.read_csv('data/ixi_euler.csv')
hcp_euler['site'] ='hcp'
cam_euler['site'] ='cam'
ixi_euler['site'] = 'ixi'
hcp_euler.dropna(inplace=True)
cam_euler.dropna(inplace=True)
ixi_euler.dropna(inplace=True)
hcp_euler['rh_euler'] = hcp_euler['rh_euler'].astype(int)
hcp_euler['lh_euler'] = hcp_euler['lh_euler'].astype(int)
cam_euler['rh_euler'] = cam_euler['rh_euler'].astype(int)
cam_euler['lh_euler'] = cam_euler['lh_euler'].astype(int)
ixi_euler['rh_euler'] = ixi_euler['rh_euler'].astype(int)
```

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

```
ixi_euler['lh_euler'] = ixi_euler['lh_euler'].astype(int)
hcp_cam_euler = pd.merge(hcp_euler, cam_euler, how='outer')
df_euler = pd.merge(ixi_euler, hcp_cam_euler, how='outer')
df_euler['avg_euler'] = df_euler[['lh_euler','rh_euler']].mean(axis=1)
df_euler.groupby(by='site').median()
df_euler['site_median'] = df_euler['site']
df_euler['site_median'] = df_euler['site_median'].replace({'hcp':-43, 'cam'
:-61, 'ixi' :-56})
df_euler['avg_euler_centered'] = df_euler['avg_euler'] -
df_euler['site_median']
df_euler['avg_euler_centered_neg'] = df_euler['avg_euler_centered']*-1
df_euler['avg_euler_centered_neg_sqrt'] =
np.sqrt(np.absolute(df_euler['avg_euler_centered_neg']))

brain = pd.merge(df_euler, brain_all, how='inner')
brain_good = brain.query('avg_euler_centered_neg_sqrt < 10')
```

Critical Step: If possible, data should be visually inspected to verify that the data inclusion is not too strict or too lenient. Subjects above the Euler number threshold should be manually checked to verify and justify their exclusion due to poor data quality. This is just one approach for automated QC used by the developers of the PCNtoolkit. Other approaches such as the ENIGMA QC pipeline or UK Biobank's QC pipeline [55] are also viable options for automated QC.

Data Preparation: Check that subjects (rows) align across covariate and brain dataframes

Timing: 3-5 minutes—4. The normative modeling function requires the covariate predictors and brain features to be in separate text files. However, it is important to first (inner) merge them together, using the following commands, to confirm that the same subjects are in each file and that the rows (representing subjects) align. This requires that both data frames have 'subject_id' as a column name. Once this is confirmed, exclude rows with NaN values and separate the brain features and covariate predictors into their own dataframes, using the commands below.

```
# make sure to use how="inner" so that we only include subjects that have
data in both the covariate and the cortical thickness files
all_data = pd.merge(brain_good, cov, how='inner')
# Create a list of all the ROIs you want to run a normative model for
roi_ids = ['lh_MeanThickness_thickness',
            'rh_MeanThickness_thickness',
            'lh_bankssts_thickness',
            'lh_caudalanteriorcingulate_thickness',
            'lh_superiorfrontal_thickness',
            'rh_superiorfrontal_thickness']
from sklearn.model_selection import train_test_split
all_data = all_data.dropna()
```

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

```
all_data_features = all_data[[subset=roi_ids]]
all data covariates = all data[['age','sex','site']]
```

Critical Step: roi_ids is a variable that represents which brain areas will be modeled and can be used to select subsets of the data frame if you do not wish to run models for the whole brain.

Data Preparation: Add variable to model site/scanner effects

Timing: 3-5 minutes—5. Currently, the different sites are coded in a single column (named ‘site’) and are represented as a string data type. However, the PCNtoolkit requires binary variables. Use the pandas package as follows to address this, which has a built-in function, pd.get_dummies, that takes in the string ‘site’ column and dummy encodes the site variable so that there is now a column for each site and the columns contain binary variables (0=not in this site, 1=present in this site).

```
all_data_covariates = pd.get_dummies(all_data_covariates, columns=['site'])
all_data['Average_Thickness'] =
all data[['lh MeanThickness thickness','rh MeanThickness
thickness']].mean(axis=1)
```

Data Preparation: Train/Test split

Timing: 5-10 minutes—6. In this example, we use 80% of the data for training and 20% for testing. Please carefully read the experimental design section on train/test split considerations when using your own data in this step. Using a function from scikit-learn (**train_test_split**), stratify the train/test split using the site variable to make sure that the train/test sets both contain data from all sites, using the following commands. Next, confirm that your train and test arrays are the same size (rows), using the following commands. You do not need the same size columns (subjects) in the train and test arrays, but the rows represent the covariate and responses which should be the same across train and test arrays.

```
X_train, X_test, y_train, y_test = train_test_split(all_data_covariates,
all_data_features,
stratify=all_data['site'], test_size=0.2, random_state=42)
tr_cov_size = X_train.shape
tr_resp_size = y_train.shape
te_cov_size = X_test.shape
te_resp_size = y_test.shape
print("Train covariate size is: ", tr_cov_size)
print("Test covariate size is: ", te_cov_size)
print("Train response size is: ", tr_resp_size)
print("Test response size is: ", te resp size)
```

Critical Step: The model would not learn the site effects if all the data from one site was only in the test set. Therefore, we stratify the train/test split using the site variable.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

7. When the data were split into train and test sets, the row index was not reset. This means that the row index in the train and test data frames still correspond to the full data frame (before splitting the data occurred). The test set row index informs which subjects belong to which site, and this information is needed to evaluate per site performance metrics. Resetting the row index of the train/test data frames fixes this issue. Then extract the site row indices to a list (one list per site) and create a list called site_names that is used to decide which sites to evaluate model performance for, as follows:

```
x_col_names = ['age','sex','site_cam','site_hcp','site_ixi']
X_train = pd.read_csv('data/covariate_files/cov_tr.txt', sep='\t',
header=None, names=x_col_names)
X_test = pd.read_csv('data/covariate_files/cov_te.txt', sep='\t',
header=None, names=x_col_names)
y_train = pd.read_csv('data/response_files/resp_tr.txt', sep='\t',
header=None)
y_test = pd.read_csv('data/response_files/resp_te.txt', sep='\t',
header=None)
X_train.reset_index(drop=True, inplace=True)
X_test.reset_index(drop=True, inplace=True)
y_train.reset_index(drop=True, inplace=True)
y_test.reset_index(drop=True, inplace=True)
cam_idx = X_test.index[X_test['site_cam']== 1].to_list()
hcp_idx = X_test.index[X_test['site_hcp'] == 1].to_list()
ixi_idx = X_test.index[X_test['site_ixi'] == 1].to_list()
# Save the site indices into a single list
sites = [cam_idx, hcp_idx, ixi_idx]
# Create a list with sites names to use in evaluating per-site metrics
site_names = ['cam', 'hcp',    'ixi']
```

Data Preparation: Setup output directories

Timing: 1-3 minutes—8. Save each brain region to its own text file (organized in separate directories) using the following commands, because for each response variable, **Y** (e.g., brain region) we fit a separate normative model.

```
for c in y_train.columns:
        y_train[c].to_csv('resp_tr_' + c + '.txt' , header=False,
index=False)
        X_train.to_csv('cov_tr.txt', sep = '\t', header=False, index = False)
        y_train.to_csv('resp_tr.txt', sep = '\t', header=False, index =False)
forc iny_test.columns:
        y_test[c].to_csv('resp_te_'+ c +'.txt', header=False, index=False)
        X_test.to_csv('cov_te.txt', sep = '\t', header=False, index = False)
        y_test.to_csv('resp_te.txt', sep = '\t', header=False, index = False)
! if [[! -e data/ROI_models/]]; then mkdir data/ROI_models; fi
! if [[! -e data/covariate_files/]]; then mkdir data/covariate_files; fi
```

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

```
! if [[! -e data/response_files/]]; then mkdir data/response_files; fi
! for i in 'cat data/roi_dir_names'; do cd data/ROI_models; mkdir ${i};
cd../../; cp
resp_tr_${i}.txt data/ROI_models/${i}/resp_tr.txt; cp resp_te_${i}.txt
data/ROI_models/${i}/resp_te.txt; cp cov_tr.txt data/ROI_models/${i}/
cov_tr.txt; cp cov_te.txt
data/ROI_models/${i}/cov_te.txt; done
! mv resp_*.txt data/response_files/
! mv cov_t*.txt data/covariate_files/
```

Algorithm & Modeling: Basis expansion using B-splines

Timing: 1-3 minutes—9. Now, set up a B-spline basis set that allows us to perform nonlinear regression using a linear model, using the following commands. This basis is deliberately chosen to not to be too flexible so that it can only model relatively slowly varying trends. To increase the flexibility of the model you can change the parameterization (e.g., by adding knot points to the B-spline basis or increasing the order of the interpolating polynomial). Note that in the neuroimaging literature, it is more common to use a polynomial basis expansion for this. Piecewise polynomials like B-splines are superior to polynomial basis expansions because they do not introduce a global curvature. For further details on the use of B-splines see Fraza et al[25].

```
# Create a cubic B-spline basis (used for regression)
xmin = 10#16 # xmin & xmax are the boundaries for ages of participants in
the dataset
xmax = 95#90
B = create_bspline_basis(xmin, xmax)
# create the basis expansion for the covariates for each of the
for roi in roi_ids:
    print('Creating basis expansion for ROI:', roi)
    roi_dir = os.path.join(data_dir, roi)
    os.chdir(roi_dir)
    #    create output dir
    os.makedirs(os.path.join(roi_dir,'blr'), exist_ok=True)
    #    load train & test covariate data matrices
    X_tr = np.loadtxt(os.path.join(roi_dir, 'cov_tr.txt'))
    X_te = np.loadtxt(os.path.join(roi_dir, 'cov_te.txt'))
    #    add intercept column
    X_tr = np.concatenate((X_tr, np.ones((X_tr.shape[0],1))), axis=1)
    X_te = np.concatenate((X_te, np.ones((X_te.shape[0],1))), axis=1)
    np.savetxt(os.path.join(roi_dir, 'cov_int_tr.txt'), X_tr)
    np.savetxt(os.path.join(roi_dir, 'cov_int_te.txt'), X_te)
    #    create Bspline basis set
    Phi = np.array([B(i) for i in X_tr[:,0]])
    Phis = np.array([B(i) for i in X_te[:,0]])
```

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

```
X_tr = np.concatenate((X_tr, Phi), axis=1)
X_te = np.concatenate((X_te, Phis), axis=1)
np.savetxt(os.path.join(roi_dir, 'cov_bspline_tr.txt'), X_tr)
np.savetxt(os.path.join(roi dir, 'cov bspline te.txt'), X te)
```

Algorithm & Modeling - estimate normative model

Timing: 3-5 minutes per model (multiply by number of ROIs/models)—10. Set up a variable (`data_dir`) that specifies the path to the ROI directories that were created in Step 7. Initiate two empty pandas data frames where the evaluation metrics are the column names, as follows; one will be used for overall test set evaluation (`blr_metrics`) and one will be used for site-specific test set evaluation (`blr_site_metrics`). After the normative model has been estimated, these data frames will be saved as individual csv files.

```
# set this path to wherever your ROI_models folder is located (where you
copied all of the covariate & response text files to in Step 4)
data dir = '/Users/saigerutherford/repos/PCNToolkit-demo/data/ROI models/'
# Create pandas dataframes with header names to save out the overall and
per-site model evaluation metrics
blr metrics = pd.DataFrame(columns = [ 'ROI',    'MSLL', 'EV',    'SMSE',
'RMSE', 'Rho'])
blr_site_metrics = pd.DataFrame(columns = [ 'ROI', 'site',    ' y mean', 'y
var',    ' yhat mean',
                                         'yhat_var', 'MSLL', 'EV', 'SMSE',
'RMSE', 'Rho'])
```

11. Estimate the normative models using a for loop to iterate over brain regions. The estimate function uses a few specific arguments that are worthy of commenting on:

- alg = 'blr': specifies we should use Bayesian Linear Regression. See Table 1 for other available algorithms.
- optimizer = 'powell': use Powell's derivative-free optimization method (faster in this case than L-BFGS)
- savemodel = False: do not write out the final estimated model to disk
- saveoutput = False: return the outputs directly rather than writing them to disk
- standardize = False: Do not standardize the covariates or response variables

An important consideration is whether to re-scale or standardize the covariates or responses. Whilst this generally only has a minor effect on the final model accuracy, it has implications for the interpretation of models and how they are configured. If the covariates and responses are both standardized (standardize = True), the model will return standardized coefficients. If (as in this case) the response variables are not standardized (standardized = False), then the scaling both covariates and responses will be reflected in the estimated coefficients. Also, under the linear modeling approach employed here, if the coefficients are unstandardized

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

and do not have a zero mean, it is necessary to add an intercept column to the design matrix (this is done above in step 9 (B-spline)).

Critical Step: This code fragment will loop through each region of interest in the roi_ids list (created in step 4) using Bayesian Linear Regression and evaluate the model on the independent test set. In principle, we could estimate the normative models on the whole data matrix at once (e.g., with the response variables stored in a n_subjects by n_brain_measures NumPy array or a text file instead of saved out into separate directories). However, running the models iteratively gives some extra flexibility in that it does not require that the included subjects are the same for each of the brain measures.

```
# Loop through ROIs
for roi in roi_ids:
    print(‘Running ROI:', roi)
    roi_dir = os.path.join(data_dir, roi)
    os.chdir(roi_dir)
    # configure the covariates to use. Change *_bspline_* to *_int_* to
    cov_file_tr = os.path.join(roi_dir, ‘cov_bspline_tr.txt')
    cov_file_te = os.path.join(roi_dir, ‘cov_bspline_te.txt')
    # load train & test response files
    resp_file_tr = os.path.join(roi_dir, ‘resp_tr.txt')
    resp_file_te = os.path.join(roi_dir, ‘resp_te.txt')
    # run a basic model
    yhat_te, s2_te, nm, Z, metrics_te = estimate(cov_file_tr,
                                                         resp_file_tr,

testresp=resp_file_te,
                                                         testcov=cov_file_te,
                                                         alg = ‘blr',
                                                         optimizer = ‘powell',
                                                         savemodel = False,
                                                         saveoutput = False,
                                                         standardize = False)
 # display and save metrics
      print(‘EV=‘, metrics_te[‘EXPV'][0])
      print(‘RHO=‘, metrics_te[‘Rho'][0])
      print(‘MSLL=‘, metrics_te[‘MSLL'][0])
      blr_metrics.loc[len(blr_metrics)] = [roi, metrics_te[‘MSLL'][0],
      metrics_te[‘EXPV'][0],
      metrics_te[‘SMSE'][0], metrics_te[‘RMSE'][0],
      metrics_te[‘Rho'][0]]
      # Compute metrics per site in test set, save to pandas df
      # load true test data
      X_te = np.loadtxt(cov_file_te)
      y_te = np.loadtxt(resp_file_te)
```

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

```
      y_te = y_te[:, np.newaxis] # make sure it is a 2-d array
      # load training data (required to compute the MSLL)
      y_tr = np.loadtxt(resp_file_tr)
      y_tr = y_tr[:, np.newaxis]
      for num, site in enumerate(sites):
           y_mean_te_site = np.array([[np.mean(y_te[site])]])
           y_var_te_site = np.array([[np.var(y_te[site])]])
           yhat_mean_te_site = np.array([[np.mean(yhat_te[site])]])
           yhat_var_te_site = np.array([[np.var(yhat_te[site])]])
           metrics_te_site = evaluate(y_te[site], yhat_te[site],
s2_te[site],
           y_mean_te_site, y_var_te_site)
           site_name = site_names[num]
           blr_site_metrics.loc[len(blr_site_metrics)] = [roi,
site_names[num],

y_mean_te_site[0],

y_var_te_site[0],

yhat_mean_te_site[0],

yhat_var_te_site[0],

metrics_te_site['MSLL'][0],

metrics_te_site['EXPV'][0],

metrics_te_site['SMSE'][0],

metrics_te_site['RMSE'][0],

metrics_te_site['Rho'][0]]
```

Evaluation & Interpretation - evaluate normative model performance

Timing: 5-10 minutes—12. In step 11, when we looped over each region of interest in the roi_ids list (created in step 4) and evaluated the normative model on the independent test set, it also computed the evaluation metrics such as the explained variance, mean standardized log-loss and Pearson correlation between true and predicted test responses. The evaluation metrics were calculated for the full test set and calculated separately for each scanning site. The metrics were saved out to a csv file. In this step we load the evaluation metrics into a panads data frame and use the describe function to show the range, mean, and standard deviation of each of the evaluation metrics. Table 2 shows how to interpret the ranges/directions of good model fit.

Europe PMC Funders Author Manuscripts

```
metrics_te([‘EXPV']).describe()
metrics_te([‘MSLL']).describe()
metrics_te([‘SMSE']).describe()
metrics_te([‘Rho']).describe()
metrics_te_site([‘EV']).describe()
metrics_te_site([‘MSLL']).describe()
metrics_te_site([‘SMSE']).describe()
metrics_te_site([‘Rho']).describe()
```

Evaluation & Interpretation - visualize normative model outputs

Timing: 15-20 minutes—13. In this step we look at different ways of visualizing the evaluation metrics from step 12. There are typically many models fit across the different brain regions and it can be a lot of effort to keep track of the performance across all the brain regions. Data visualization will help to understand if there are any emerging patterns and find if there are any brain areas (or certain sites) where the model does not fit well. We summarize the deviation scores in the test set by counting how many subjects have an ‘extreme’ deviation (either positive or negative) and visualize the count of extreme negative and positive deviations by plotting them on a 3D brain plot. This step requires using a separate python notebook (https://github.com/predictive-clinical-neuroscience/PCNtoolkit-demo/blob/main/tutorials/BLR_protocol/visualizations.ipynb)

Evaluation & Interpretation – post-hoc analysis ideas using normative modeling outputs

Timing: 1-2 hours—14. There are many interesting analyses that can be conducted using the outputs of normative modeling (deviation scores). An in-depth tutorial on each of these analyses is outside the scope of this protocol. However, on GitHub, we include code examples (python notebooks that can be run via Colab) of the following post-hoc analysis: (https://github.com/predictive-clinical-neuroscience/PCNtoolkit-demo/blob/main/tutorials/BLR_protocol/post_hoc_analysis.ipynb)

- Using deviation scores as predictors in a regression and classification and comparing the performance to using the true data as predictors. Code for implementing several common predictive modeling frameworks (that are mentioned in comparison to other methods section) is provided. Deviation scores from normative modeling could be used as input features to any of these predictive modeling frameworks.
- Using a pre-trained normative model and transferring it to a new, unseen data set.
- Classical case-control testing (univariate t-tests) on deviation maps compared to univariate t-tests on the true data.

Timing

The normative modeling portion of this protocol (including evaluation and visualization) can be completed in approximately 57-72 minutes. If using the additional code for post-hoc analysis of the normative modeling outputs, you would add approximately 1-2 hours to the estimated normative modeling time. These timing estimates are if using the Google Colab

Europe PMC Funders Author Manuscripts

platform to run the code. If running this protocol on your own computer (where you need to install python and dependencies), this will add extra time to the protocol.

Anticipated Results

There are multiple end products created from running a normative model analysis. First, the evaluation metrics for each model (brain region) are saved to a file. In this protocol, we saved the metrics to a CSV file format, however, in the pcn.estimate() function you could set the argument 'binary = True' which would save the metrics in pickle (.pkl) format. Pickle format is good to use if you are estimating many models in parallel on a large dataset, as it is faster because it avoids reading/writing intermediate text files. These metrics are further summarized into per site metrics to check model fit for each site included in the test set. The short and full names of the evaluation metrics and a brief interpretation guide is summarized in Table 2. The evaluation metrics can be visualized in numerous formats, histograms/density plots, scatter plots with fitted centiles, or brain-space visualizations. Several examples of these visualizations are shown in Figure 4 and code for creating these plots is shared on GitHub. Quality checking the normative model evaluation metrics should be done to ensure proper model estimation. If a model fits well to the data, the evaluation metrics should follow a Gaussian distribution. The model estimation (Procedure step 11) should properly handle confounding site effects, nevertheless, it is also a good idea to check per site metrics to make sure the model is fitting all sites equally well and that there are no obvious site outliers. In addition to the summary level evaluation metrics, there are also many individual metrics (one value per subject for each model/brain region). These individual-level outputs can be very helpful for interpretation because they precisely quantify the uncertainty of each individual predicted value at every location across the brain. If a given individual is identified as having an 'extreme' deviation, and there is low uncertainty you can be confident this is a biologically valid finding and not due to modeling errors. Vice versa, if there are extreme deviations and high levels of uncertainty, more caution should be given to interpretating these results and the deviations may be due to modeling errors rather than true biological variation. The uncertainty estimates are separated into two components (noise and modeling, described in Table 2) to help pinpoint the sources of uncertainty.

A benefit of the PCNtoolkit software for normative modeling, that sets our approach apart from other normative modeling implementations[84], is the fine-scale resolution allowed by the model. Other normative modeling work[84] has focused on modeling gross features such as total brain volume or gray matter volume, which is not adequate for normative modeling applied to mental health conditions and neurodevelopmental disorders, where the effects are subtle and widespread (individuals within a patient group tend to deviate in different regions, see Figure 4E-F) across the cortex and subcortex and averaging over large brain areas usually overlooks these elusive psychiatric effects. This resolution also allows for a better mechanistic understanding because you can quantify the deviation and associated uncertainty for each individual with high spatial precision.

Reliability (the extent to which a measurement gives results that are very consistent) and validity (the degree to which a measurement measures what it is supposed to measure) are

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

important constructs to keep in mind when interpreting results. In recent work, reliability of normative modeling in schizophrenia and bipolar disorder using structural MRI measures was established via replication[37]. Validity is arguably more challenging to assess but should be established by means of out of sample model fit. In other recent work, normative models were fit using a lifespan (age 3-100) big data sample (N=58,836) and carefully tested out-of-sample (variance explained, skewness, kurtosis, and standardized mean squared error) showing excellent model fit (12-68% variance explained) in an independent test set from a sample (and site) that was not included in the training set[85]. This work suggests validity, but this is an on-going evaluation and out of sample model fit must always be considered and reported.

Troubleshooting

We re-iterate that there is additional documentation available online through read the docs including additional tutorials for other algorithm implementations (Gaussian Process Regression and Hierarchical Bayesian Regression), a glossary to clarify the jargon associated with the software, a reference guide with links to normative modeling publications, and a frequently asked questions page where many common errors (and their solutions) are discussed in detail. The problems encountered when troubleshooting a normative modeling analysis can fall into three categories: computing errors, data issues, and misunderstanding or misinterpreting the outputs.

Computing errors

The computing errors might involve python or the computer hardware. Potential python errors may include installation of python or installation of the necessary packages and their dependencies. We recommend using Anaconda to install python 3.8 (required for this protocol) on your system, and the use of a virtual environment for the PCNtoolkit to ensure that the packages required for normative modeling do not interfere with other python versions and packages you may have installed on your system. In general, it is good to have a virtual environment setup for each project or analysis. If you are unfamiliar with setting up virtual environments, and run into issues with python, it is always an option to run the analysis in the cloud via Colab which eliminates the need to setup python on your own system. Hardware problems might include lack of memory to store the data or models running very slowly due to outdated hardware. These hardware errors do not have an easy solution, and we recommend using Google Colab to run normative modeling analysis if your personal computer or server is very slow or lacks the storage space.

Data issues

Data issues that may be encountered are data missing not at random (see Experimental Design section regarding caution using data imputation), improperly coded data (i.e., strings instead of integers or floats, NaN values coded incorrectly), collinearity of columns in the covariate design matrix, or outlier data that does not make biological sense (i.e., negative cortical thickness values, negative age values). While these data errors can be incredibly frustrating to troubleshoot, they can typically be fixed by careful quality checking of the input data and removal of bad ROIs or subjects as needed.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Interpretation confusion

Finally, an example of interpretation confusion may be poor model performance on a certain brain region or site. This can usually be addressed by returning to the input data for additional quality checking to confirm that the poor performance is not due to data quality issues. If there are no data quality issues, then it may be the reality that the model does not fit well some brain regions, and you may want to consider including additional covariates in the model to help explain more variance. Another interpretation confusion may arise when seeing negative explained variance values. When testing out of sample, the explained variance is not restricted to be positive, if it is negative this mean that the model fit is very poor (it is worse than an intercept-only model).

Acknowledgements

This research was supported by grants from the European Research Council (ERC, grant “MENTALPRECISION” 10100118 and “BRAINMINT” 802998), the Wellcome Trust under an Innovator award (“BRAINCHART”, 215698/Z/19/Z) and a Strategic Award (098369/Z/12/Z), the Dutch Organisation for Scientific Research (VIDI grant 016.156.415). TW also gratefully acknowledges the Niels Stensen Fellowship as well as the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant agreement No. 895011.

Data Availability Statement

All data used in this protocol are available on GitHub and Zenodo[86] in csv files. We also include a template csv file to help format user’s own data into the correct form for running the protocol using their own data set.

Code Availability Statement

All code is available on GitHub in the format of python notebooks that can be run in the cloud (for free) using Google Colab. We have also shared the GitHub repository on Zenodo to create a citable DOI for this software that also allows versions which are necessary as additional code and tutorials may be added over time[86].

References

1. Wang D, et al. Parcellating cortical functional networks in individuals. Nat Neurosci. 2015; 18: 1853–1860. [PubMed: 26551545]
2. Finn ES, Constable RT. Individual variation in functional brain connectivity: implications for personalized approaches to psychiatric disease. Dialogues Clin Neurosci. 2016; 18: 277–287. [PubMed: 27757062]
3. Braga RM, Buckner RL. Parallel Interdigitated Distributed Networks within the Individual Estimated by Intrinsic Functional Connectivity. Neuron. 2017; 95: 457–471. e5 [PubMed: 28728026]
4. Poldrack RA. Precision Neuroscience: Dense Sampling of Individual Brains. Neuron. 2017; 95: 727–729. [PubMed: 28817793]
5. Vanderwal T, et al. Individual differences in functional connectivity during naturalistic viewing conditions. NeuroImage. 2017; 157: 521–530. [PubMed: 28625875]
6. Braun U, et al. From Maps to Multi-dimensional Network Mechanisms of Mental Disorders. Neuron. 2018; 97: 14–31. [PubMed: 29301099]
7. Gratton C, et al. Defining Individual-Specific Functional Neuroanatomy for Precision Psychiatry. Biological Psychiatry. 2020; 88: 28–39. [PubMed: 31916942]

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

8. Hyman SE. Can neuroscience be integrated into the DSM-V? Nat Rev Neurosci. 2007; 8: 725–732. [PubMed: 17704814]

9. Insel T, et al. Research Domain Criteria (RDoC): Toward a New Classification Framework for Research on Mental Disorders. AJP. 2010; 167: 748–751.

10. Michelini G, Palumbo IM, DeYoung CG, Latzman RD, Kotov R. Linking RDoC and HiTOP: A new interface for advancing psychiatric nosology and neuroscience. Clinical Psychology Review. 2021; 86 102025 [PubMed: 33798996]

11. Narrow WE, Kuhl EA. Dimensional approaches to psychiatric diagnosis in DSM-5. J Ment Health Policy Econ. 2011; 14: 197–200. [PubMed: 22345361]

12. Cuthbert BN, Insel TR. Toward the future of psychiatric diagnosis: the seven pillars of RDoC. BMC Med. 2013; 11: 126. [PubMed: 23672542]

13. Sanislow CA. RDoC at 10: changing the discourse for psychopathology. World Psychiatry. 2020; 19: 311–312. [PubMed: 32931117]

14. Kotov R, et al. The Hierarchical Taxonomy of Psychopathology (HiTOP): A dimensional alternative to traditional nosologies. Journal of Abnormal Psychology. 2017; 126: 454–477. [PubMed: 28333488]

15. Kotov R, et al. The Hierarchical Taxonomy of Psychopathology (HiTOP): A Quantitative Nosology Based on Consensus of Evidence. Annu Rev Clin Psychol. 2021; 17

16. Haro JM, et al. ROAMER: roadmap for mental health research in Europe. Int J Methods Psychiatr Res. 2014; 23 (Suppl 1) 1–14.

17. Schumann G, et al. Stratified medicine for mental disorders. Eur Neuropsychopharmacol. 2014; 24: 5–50. [PubMed: 24176673]

18. Feczko E, et al. The Heterogeneity Problem: Approaches to Identify Psychiatric Subtypes. Trends in Cognitive Sciences. 2019; 23: 584–601. [PubMed: 31153774]

19. Shen X, et al. Using connectome-based predictive modeling to predict individual behavior from brain connectivity. Nature Protocols. 2017; 12: 506–518. [PubMed: 28182017]

20. Sripada C, et al. Basic Units of Inter-Individual Variation in Resting State Connectomes. Scientific Reports. 2019; 9: 1900. [PubMed: 30760808]

21. Woo C-W, Chang LJ, Lindquist MA, Wager TD. Building better biomarkers: brain models in translational neuroimaging. Nat Neurosci. 2017; 20: 365–377. [PubMed: 28230847]

22. Marquand AF, et al. Conceptualizing mental disorders as deviations from normative functioning. Molecular Psychiatry. 2019; 24: 1415–1424. [PubMed: 31201374]

23. Gau R, et al. Brainhack: Developing a culture of open, inclusive, community-driven neuroscience. Neuron. 2021; 109: 1769–1775. [PubMed: 33932337]

24. Olah C, Carter S. Research Debt. Distill. 2017; 2: e5.

25. Fraza CJ, Dinga R, Beckmann CF, Marquand AF. Warped Bayesian Linear Regression for Normative Modelling of Big Data. bioRxiv. 2021; 2021.04.05.438429 doi: 10.1101/2021.04.05.438429

26. Dinga R, et al. Normative modeling of neuroimaging data using generalized additive models of location scale and shape. 2021; doi: 10.1101/2021.06.14.448106

27. Kia, SM, , et al. Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. Martel, AL, , et al., editors. Springer International Publishing; 2020. 699–709.

28. Kia SM, et al. Federated Multi-Site Normative Modeling using Hierarchical Bayesian Regression. bioRxiv. 2021; 2021.05.28.446120 doi: 10.1101/2021.05.28.446120

29. Floris DL, et al. Atypical Brain Asymmetry in Autism—A Candidate for Clinically Meaningful Stratification. Biological Psychiatry: Cognitive Neuroscience and Neuroimaging. 2020; doi: 10.1016/j.bpsc.2020.08.008

30. Zabihi M, et al. Dissecting the Heterogeneous Cortical Anatomy of Autism Spectrum Disorder Using Normative Models. Biological Psychiatry: Cognitive Neuroscience and Neuroimaging. 2019; 4: 567–578. [PubMed: 30799285]

31. Zabihi M, et al. Fractionating autism based on neuroanatomical normative modeling. Translational Psychiatry. 2020; 10: 1–10. [PubMed: 32066695]

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

32. Wolfers T, et al. Individual differences v. the average patient: mapping the heterogeneity in ADHD using normative models. Psychological Medicine. 2020; 50: 314–323. [PubMed: 30782224]

33. Wolfers T, et al. Refinement by integration: aggregated effects of multimodal imaging markers on adult ADHD. J Psychiatry Neurosci. 2017; 42: 386–394. [PubMed: 28832320]

34. Verdi S, Marquand AF, Schott JM, Cole JH. Beyond the average patient: how neuroimaging models can address heterogeneity in dementia. Brain. 2021; doi: 10.1093/brain/awab165

35. Wolfers T, et al. Extensive brain structural heterogeneity in individuals with schizophrenia and bipolar disorder. 2020; doi: 10.1101/2020.05.08.20095091

36. Wolfers T, et al. Mapping the Heterogeneous Phenotype of Schizophrenia and Bipolar Disorder Using Normative Models. JAMA Psychiatry. 2018; 75: 1146–1155. [PubMed: 30304337]

37. Wolfers T, et al. Replicating extensive brain structural heterogeneity in individuals with schizophrenia and bipolar disorder. Human Brain Mapping. 2021; 42: 2546–2555. [PubMed: 33638594]

38. Sripada C, Angstadt M, Rutherford S, Taxali A. Brain Network Mechanisms of General Intelligence. bioRxiv. 2019; 657205 doi: 10.1101/657205

39. Sripada C, et al. Brain Connectivity Patterns in Children Linked to Neurocognitive Abilities. 2020; doi: 10.1101/2020.09.10.291500

40. Rosenberg MD, et al. A neuromarker of sustained attention from whole-brain functional connectivity. Nature Neuroscience. 2015; 19: 165–171. [PubMed: 26595653]

41. Rosenberg MD, et al. Functional connectivity predicts changes in attention observed across minutes, days, and months. PNAS. 2020; 117: 3797–3807. [PubMed: 32019892]

42. Marquand AF, Haak KV, Beckmann CF. Functional corticostriatal connection topographies predict goal directed behaviour in humans. Nat Hum Behav. 2017; 1

43. Marquand A, et al. Quantitative prediction of subjective pain intensity from whole-brain fMRI data using Gaussian processes. NeuroImage. 2010; 49: 2178–2189. [PubMed: 19879364]

44. Wager TD, et al. An fMRI-Based Neurologic Signature of Physical Pain. New England Journal of Medicine. 2013; 368: 1388–1397. [PubMed: 23574118]

45. Sripada C, Angstadt M, Rutherford S, Taxali A, Shedden K. Toward a “treadmill test” for cognition: Improved prediction of general cognitive ability from the task activated brain. Human Brain Mapping. 2020; 41: 3186–3197. [PubMed: 32364670]

46. Sripada C, Taxali A, Angstadt M, Rutherford S. Boost in Test-Retest Reliability in Resting State fMRI with Predictive Modeling. 2019; doi: 10.1101/796714

47. Finn ES, et al. Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. Nature Neuroscience. 2015; 18: 1664–1671. [PubMed: 26457551]

48. Wang H-T, et al. Finding the needle in a high-dimensional haystack: Canonical correlation analysis for neuroscientists. NeuroImage. 2020; 216 116745 [PubMed: 32278095]

49. Smith SM, et al. A positive-negative mode of population covariation links brain connectivity, demographics and behavior. Nature Neuroscience. 2015; 18: 1565–1567. [PubMed: 26414616]

50. Dadi K, et al. Benchmarking functional connectome-based predictive models for resting-state fMRI. NeuroImage. 2019; 192: 115–134. [PubMed: 30836146]

51. Lake EMR, et al. The Functional Brain Organization of an Individual Allows Prediction of Measures of Social Abilities Transdiagnostically in Autism and Attention-Deficit/Hyperactivity Disorder. Biological Psychiatry. 2019; 86: 315–326. [PubMed: 31010580]

52. Cole JH, Franke K. Predicting Age Using Neuroimaging: Innovative Brain Ageing Biomarkers. Trends in Neurosciences. 2017; 40: 681–690. [PubMed: 29074032]

53. Han LKM, et al. Brain aging in major depressive disorder: results from the ENIGMA major depressive disorder working group. Molecular Psychiatry. 2020; 1–16. DOI: 10.1038/s41380-020-0754-0

54. Sturmfels P, et al. A Domain Guided CNN Architecture for Predicting Age from Structural Brain Images. 2018. arXiv:1808.04362 [cs, stat]

55. Alfaro-Almagro F, et al. Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank. Neuroimage. 2018; 166: 400–424. [PubMed: 29079522]

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

56. Donders ART, van der Heijden GJMG, Stijnen T, Moons KGM. Review: A gentle introduction to imputation of missing values. Journal of Clinical Epidemiology. 2006; 59: 1087–1091. [PubMed: 16980149]

57. Madley-Dowd P, Hughes R, Tilling K, Heron J. The proportion of missing data should not be used to guide decisions on multiple imputation. Journal of Clinical Epidemiology. 2019; 110: 63–73. [PubMed: 30878639]

58. Burt JB, Helmer M, Shinn M, Anticevic A, Murray JD. Generative modeling of brain maps with spatial autocorrelation. NeuroImage. 2020; 220 117038 [PubMed: 32585343]

59. Shinn M, et al. Spatial and temporal autocorrelation weave human brain networks. bioRxiv. 2021; 2021.06.01.446561 doi: 10.1101/2021.06.01.446561

60. Smith SM, Nichols TE. Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. Neuroimage. 2009; 44: 83–98. [PubMed: 18501637]

61. Guo C, Kang J, Johnson TD. A spatial Bayesian latent factor model for image-on-image regression. Biometrics. 2020.

62. Woolrich MW, Behrens TEJ, Smith SM. Constrained linear basis sets for HRF modelling using Variational Bayes. NeuroImage. 2004; 21: 1748–1761. [PubMed: 15050595]

63. Liu W, Zhu P, Anderson JS, Yurgelun-Todd D, Fletcher PT. Spatial Regularization of Functional Connectivity Using High-Dimensional Markov Random Fields. Med Image Comput Comput Assist Interv. 2010; 13: 363–370. [PubMed: 20879336]

64. Song HF, Kennedy H, Wang X-J. Spatial embedding of structural similarity in the cerebral cortex. Proc Natl Acad Sci U S A. 2014; 111: 16580–16585. [PubMed: 25368200]

65. Roberts JA, et al. The contribution of geometry to the human connectome. NeuroImage. 2016; 124: 379–393. [PubMed: 26364864]

66. Bijsterbosch J, et al. Challenges and future directions for representations of functional brain organization. Nature Neuroscience. 2020; 23: 1484–1495. [PubMed: 33106677]

67. Huertas I, et al. A Bayesian spatial model for neuroimaging data based on biologically informed basis functions. NeuroImage. 2017; 161: 134–148. [PubMed: 28782681]

68. Kia SM, Marquand A. Normative Modeling of Neuroimaging Data using Scalable Multi-Task Gaussian Processes. 2018. arXiv:1806.01047 [cs, stat]

69. Kia SM, Beckmann CF, Marquand AF. Scalable Multi-Task Gaussian Process Tensor Regression for Normative Modeling of Structured Variation in Neuroimaging Data. 2018. arXiv:1808.00036 [cs, stat]

70. Jahn, A, , et al. andrewjahn/AndysBrainBook. Zenodo; 2022.

71. Casey BJ, et al. The Adolescent Brain Cognitive Development (ABCD) study: Imaging acquisition across 21 sites. Developmental Cognitive Neuroscience. 2018; 32: 43–54. [PubMed: 29567376]

72. Thompson PM, et al. ENIGMA and global neuroscience: A decade of large-scale studies of the brain in health and disease across more than 40 countries. Transl Psychiatry. 2020; 10: 100. [PubMed: 32198361]

73. Beer JC, et al. Longitudinal ComBat: A method for harmonizing longitudinal multiscanner imaging data. NeuroImage. 2020; 220 117129 [PubMed: 32640273]

74. Fortin J-P, et al. Harmonization of multi-site diffusion tensor imaging data. NeuroImage. 2017; 161: 149–170. [PubMed: 28826946]

75. Fortin J-P, et al. Harmonization of cortical thickness measurements across scanners and sites. NeuroImage. 2018; 167: 104–120. [PubMed: 29155184]

76. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics. 2007; 8: 118–127. [PubMed: 16632515]

77. Nygaard V, Rødland EA, Hovig E. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. Biostatistics. 2016; 17: 29–39. [PubMed: 26272994]

78. Noirhomme Q, et al. Biased binomial assessment of cross-validated estimation of classification accuracies illustrated in diagnosis predictions. Neuroimage Clin. 2014; 4: 687–694. [PubMed: 24936420]

79. Marquand AF, Wolfers T, Mennes M, Buitelaar J, Beckmann CF. Beyond Lumping and Splitting: A Review of Computational Approaches for Stratifying Psychiatric Disorders. Biological Psychiatry: Cognitive Neuroscience and Neuroimaging. 2016; 1: 433–447. [PubMed: 27642641]

80. Rahimi A, Recht B. Random Features for Large-Scale Kernel Machines. 8.

81. Lv J, et al. Individual deviations from normative models of brain structure in a large cross-sectional schizophrenia cohort. Molecular Psychiatry. 2020; 1–12. DOI: 10.1038/s41380-020-00882-5

82. Snelson, E, Ghahramani, Z, Rasmussen, C. Advances in Neural Information Processing Systems. Vol. 16. MIT Press; 2004.

83. Hensman J, Fusi N, Lawrence ND. Gaussian Processes for Big Data. 2013.

84. Bethlehem, RaI; , et al. Brain charts for the human lifespan. bioRxiv. 2021; 2021.06.08.447489 doi: 10.1101/2021.06.08.447489

85. Rutherford S, et al. Charting Brain Growth and Aging at High Spatial Precision. 2021; 2021.08.08.455487 doi: 10.1101/2021.08.08.455487

86. Rutherford, S, , et al. The Normative Modeling Framework for Computational Psychiatry. Zenodo; 2021.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Editorial summary

This protocol guides the user through normative modeling analysis using the Predictive Clinical Neuroscience toolkit (PCNtoolkit), enabling individual differences to be mapped at the level of a single subject or observation in relation to a reference model.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

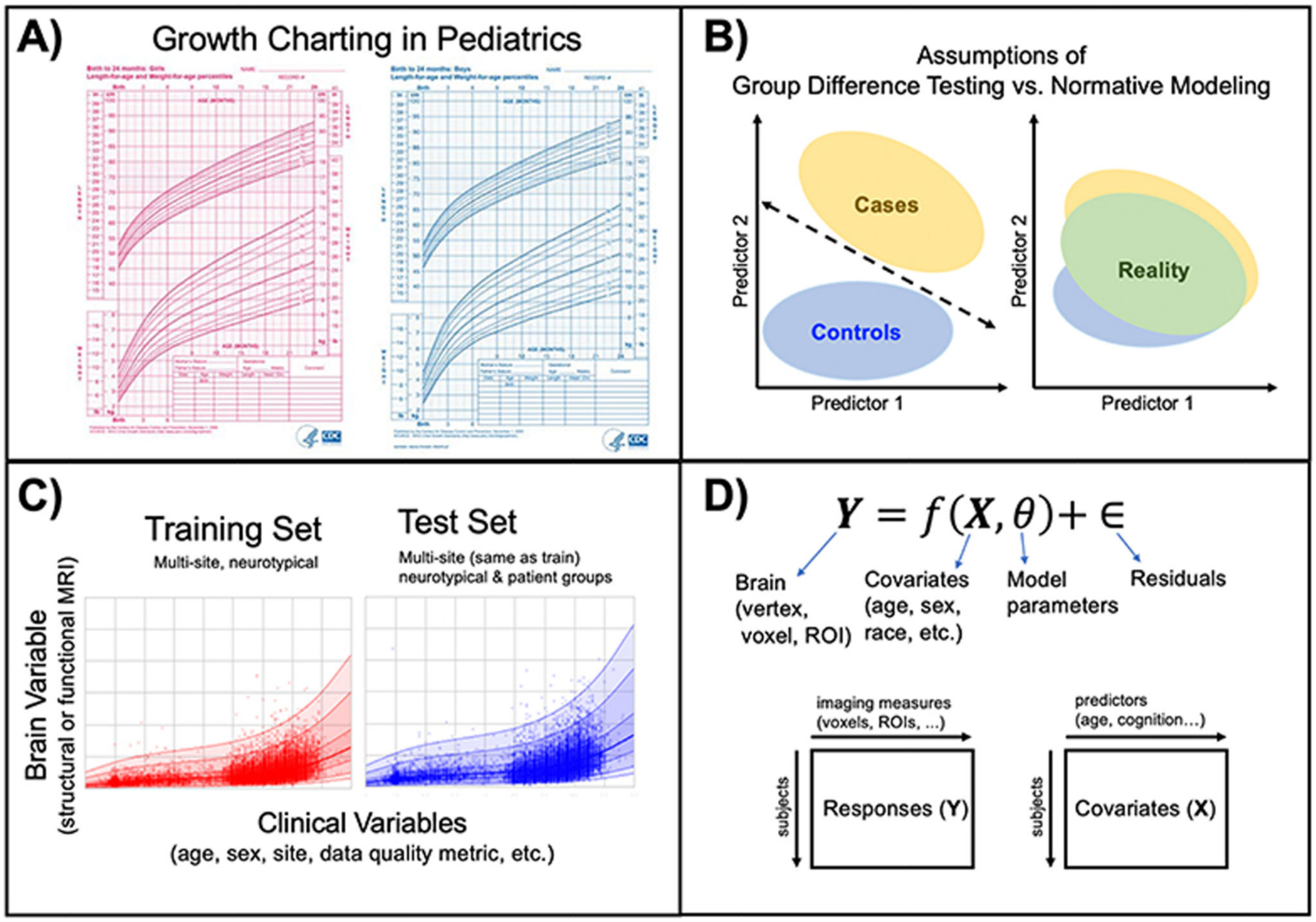
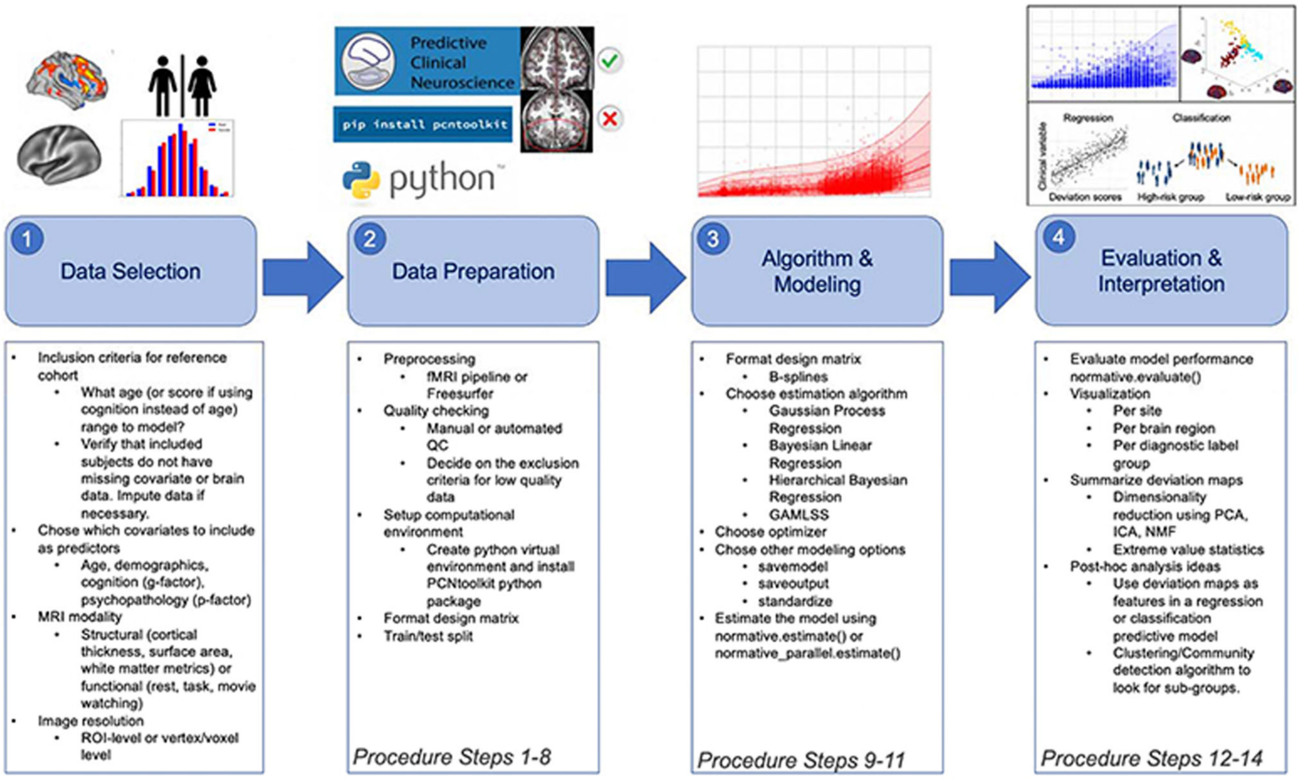


Figure 1. Conceptual Overview of Normative Modeling.

A) Classical example of normative modeling: the use of height and weight growth charting in pediatrics. **B)** Case-control models (left) theoretically make assumptions that there is a boundary that can separate groups and that there is within-group homogeneity. In reality (right), there is nested variation across controls and patient groups and within-group heterogeneity, resulting in unclear separation boundaries. Normative modeling is well equipped to handle this reality. **C)** An example application of normative modeling in computational psychiatry using neuroimaging data. Mean cortical thickness (y-axis) is predicted from age (x-axis) using a training set consisting of multi-site structural MRI from neurotypical controls and a test set consisting of neurotypical controls and patient groups. Every dot indicates the deviation score for a single individual from normal development. **D)** Regression model equation and design matrix setup for the model shown in panel C.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts



- Inclusion criteria for reference cohort
  - What age (or score if using cognition instead of age) range to model?
  - Verify that included subjects do not have missing covariate or brain data. Impute data if necessary.
- Chose which covariates to include as predictors
  - Age, demographics, cognition (g-factor), psychopathology (p-factor)
- MRI modality
  - Structural (cortical thickness, surface area, white matter metrics) or functional (rest, task, movie watching)
- Image resolution
  - ROI-level or vertex/voxel level

- Preprocessing
  - fMRI pipeline or Freesurfer
- Quality checking
  - Manual or automated QC
  - Decide on the exclusion criteria for low quality data
- Setup computational environment
  - Create python virtual environment and install PCNtoolkit python package
- Format design matrix
- Train/test split

Procedure Steps 1-8

- Format design matrix
  - B-splines
- Choose estimation algorithm
  - Gaussian Process Regression
  - Bayesian Linear Regression
  - Hierarchical Bayesian Regression
  - GAMLSS
- Choose optimizer
- Chose other modeling options
  - savemodel
  - saveoutput
  - standardize
- Estimate the model using normative.estimate() or normative_parallel.estimate()

Procedure Steps 9-11

- Evaluate model performance normative.evaluate()
- Visualization
  - Per site
  - Per brain region
  - Per diagnostic label group
- Summarize deviation maps
  - Dimensionality reduction using PCA, ICA, NMF
  - Extreme value statistics
- Post-hoc analysis ideas
  - Use deviation maps as features in a regression or classification predictive model
  - Clustering/Community detection algorithm to look for sub-groups.

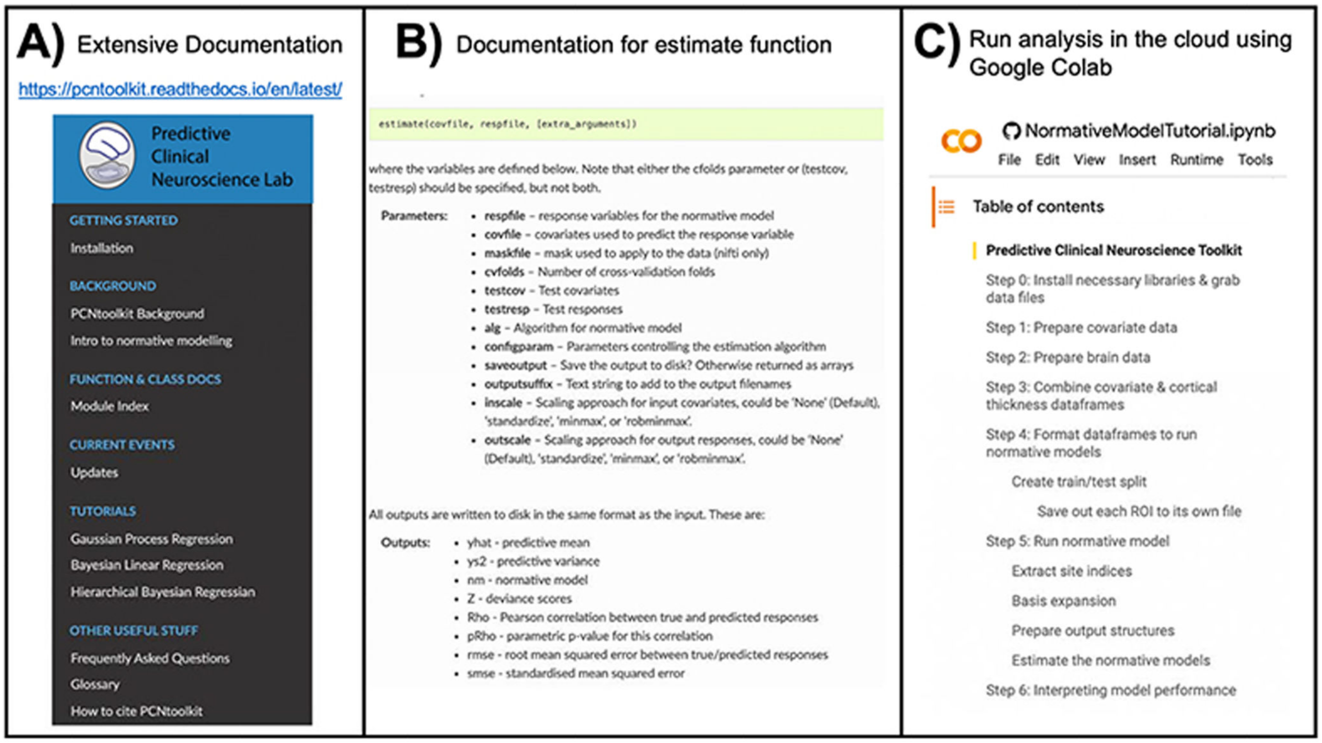
Procedure Steps 12-14

Figure 2. Practical Overview of Normative Modeling Framework.

The workflow consists of four stages: data selection, data preparation, algorithm & modeling, and evaluation & interpretation, which are visualized by the numbered shaded blue boxes. The steps involved at each of these stages are summarized in the box below and highlighted in the images above.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts


A) Extensive Documentation
https://pcntoolkit.readthedocs.io/en/latest/
Predictive Clinical Neuroscience Lab
GETTING STARTED
Installation
BACKGROUND
PCNtoolkit Background
Intro to normative modelling
FUNCTION & CLASS DOCS
Module Index
CURRENT EVENTS
Updates
TUTORIALS
Gaussian Process Regression
Bayesian Linear Regression
Hierarchical Bayesian Regression
OTHER USEFUL STUFF
Frequently Asked Questions
Glossary
How to cite PCNtoolkit
B) Documentation for estimate function
C) Run analysis in the cloud using Google Colab
NormativeModelTutorial.ipynb
File
Edit
View
Insert
Runtime
Tools
Table of contents
Predictive Clinical Neuroscience Toolkit
Step 0: Install necessary libraries & grab data files
Step 1: Prepare covariate data
Step 2: Prepare brain data
Step 3: Combine covariate & cortical thickness dataframes
Step 4: Format dataframes to run normative models
Create train/test split
Save out each ROI to its own file
Step 5: Run normative model
Extract site indices
Basis expansion
Prepare output structures
Estimate the normative models
Step 6: Interpreting model performance

```
estimate(covfile, respfile, [extra_arguments])
```

where the variables are defined below. Note that either the cfolds parameter or (testcov, testresp) should be specified, but not both.

Parameters:

- **respfile** – response variables for the normative model
- **covfile** – covariates used to predict the response variable
- **maskfile** – mask used to apply to the data (nifti only)
- **cvfolds** – Number of cross-validation folds
- **testcov** – Test covariates
- **testresp** – Test responses
- **alg** – Algorithm for normative model
- **configparam** – Parameters controlling the estimation algorithm
- **saveoutput** – Save the output to disk? Otherwise returned as arrays
- **outputsuffix** – Text string to add to the output filenames
- **inscale** – Scaling approach for input covariates, could be 'None' (Default), 'standardize', 'minmax', or 'robminmax'.
- **outscale** – Scaling approach for output responses, could be 'None' (Default), 'standardize', 'minmax', or 'robminmax'.

All outputs are written to disk in the same format as the input. These are:

Outputs:

- yhat - predictive mean
- ys2 - predictive variance
- nm - normative model
- Z - deviance scores
- Rho - Pearson correlation between true and predicted responses
- pRho - parametric p-value for this correlation
- rmse - root mean squared error between true/predicted responses
- smse - standardised mean squared error

Figure 3. Overview of Resources for Running a Normative Modeling Analysis.

A) Detailed documentation, including installation instructions, input/output descriptions of all classes and functions implemented in the python package, tutorials for all algorithms, frequently asked questions, a glossary explaining acronyms and other jargon, references to existing normative modeling literature, and a citation guide, is available online. **B)** Example of the documentation showing the required input, expected output of the main function used in the pcntoolkit software, the estimate function. **C)** All of the code and data used in this protocol is available to run in the cloud via Google Colab. Additional tutorials (shown under the tutorials header in panel A) are also available to run in Google Colab.

Europe PMC Funders Author Manuscripts
Europe PMC Funders Author Manuscripts

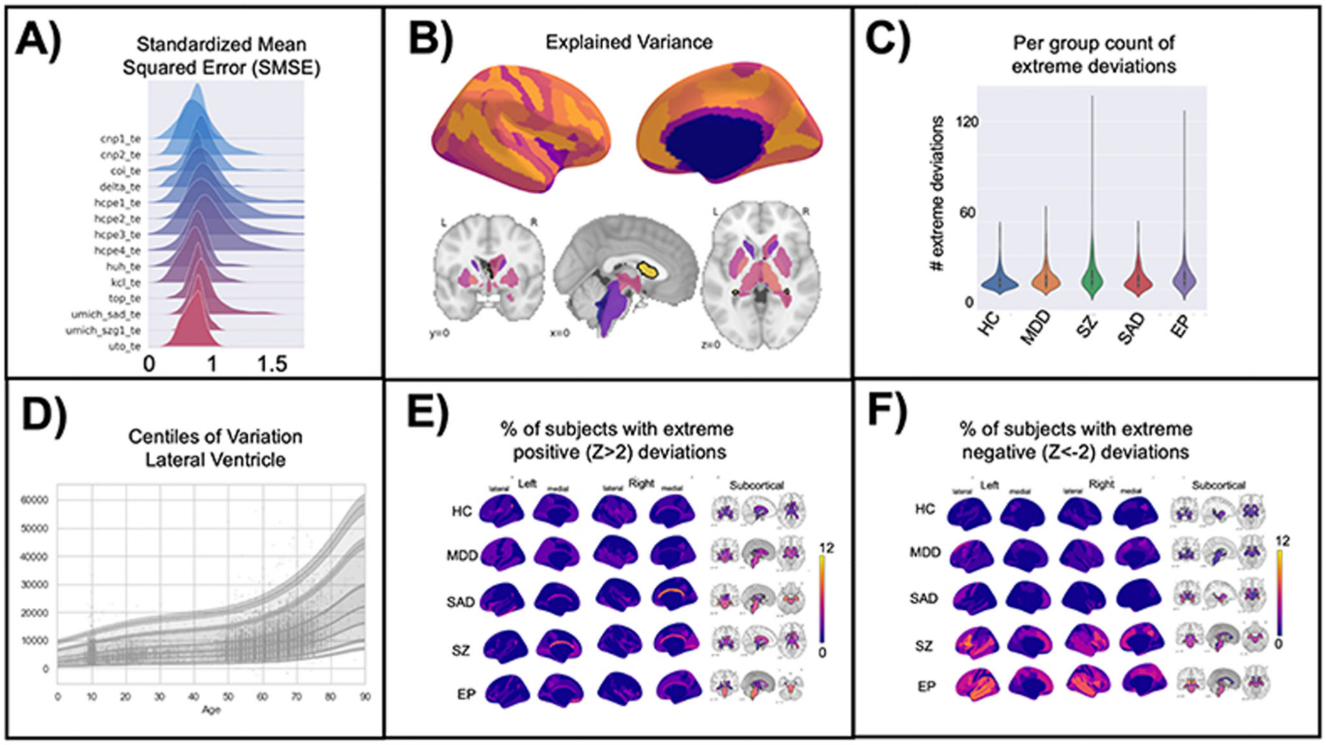


Figure 4. Visualization of Normative Model Evaluation Metrics.

A) A ridge plot showing the distribution across all brain regions of the standardized mean squared error (SMSE), an evaluation metric that represents accuracy, visualized for each site in the test set. Visualizing for each test site can help identify if there are sites where the model is performing poorly. Ideally, the distribution will be Gaussian and should look similar across all sites. Small shifts in the mean across sites is to be expected and is acceptable. **B)** Explained variance is shown for cortical thickness of every brain region in the Destrieux parcellation) and volume of subcortical regions. Visualizing the evaluation metrics in brain space helps to identify patterns and see the big picture. **C)** The number of extreme deviations (both positive and negative) are counted for each individual in the test set, group ID is used to plot the distribution of the extreme deviation count for each group. A statistical test can be done on the count to determine if there is a significant difference between groups. Testing group differences in the count of deviations does not require there to be spatial overlap of the deviations within the group (i.e., this test can account for within-group heterogeneity of deviations). **D)** The normative trajectory for an example brain region (lateral ventricle) showing age (x-axis) versus the predicted volume (y-axis). The centiles of variation are shown by the lines and shaded confidence intervals. Each subject in the test set is plotted as a single point. **E-F)** Extreme deviations, separated into positive (**E**) and negative (**F**), are summarized for each group. For each brain region, the number of subjects with an extreme deviation in that region is counted, then divided by the group sample size, to show the percent of subjects with an extreme deviation. These visualizations demonstrate the benefit of normative modeling as there is within group heterogeneity that other methods (i.e., case-control group difference testing) are not equipped to handle. Abbreviations: HC

= Controls, MDD=Major Depressive Disorder, SZ=Schizophrenia, SAD=Social Anxiety Disorder, EP=Early Psychosis.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Table 1

PCNtoolkit Normative Modeling Algorithm Overview*.

Algorithm	**Implemented in PCNtoolkit?**	**Transfer to new sites?**	**Fast compute time with large sample sizes?**	**Model non-Gaussianity?**	**Federated learning framework?**
Gaussian Process Regression (GPR) **	yes	no	no	no	no
Hierarchical Bayesian Regression (HBR)	yes	yes	yes	yes	yes
Bayesian Linear Regression (BLR)	yes	yes	yes	yes	no
Generalized additive models of location, scale, and shape (GAMLSS) ***	no	yes	yes	yes	no

* Random feature approximation and neural processes algorithms are not well documented in the PCNtoolkit and do not have tutorials available, thus these algorithms are not included in the table and are only recommended for advanced users who can implement the code on their own.

** The vanilla GPR algorithm implemented in the PCNtoolkit cannot model non-Gaussianity and does not scale well to large datasets. However, this is a question of implementation, and there are versions of GPs algorithms that satisfy these criteria[82,83].

*** Implemented in R, see this GitHub repository.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Table 2

Normative Model Metrics.

The 'Individual or summary?' column refers to whether there is a value for every subject or if the metric is summarized across all subjects. For summary metrics, there is one value per brain region (model), and for individual metrics there are `n_subjects x n_brain_regions` values.

Variable name	Full name	Definition	Interpretation	Individual or summary?
y_d	True data			individual
$\hat{y}_d$	Predictive mean			individual
σ_d^2	Predictive noise variance	Represents uncertainty in the data.		individual
$(\sigma_*^2)_d$	Predictive modeling variance	Represents uncertainty in model estimation.		individual
Z	Deviation score	A statistical estimate (Z-score) of how much each subject deviates from the normative range.	Z > 2 'extreme' positive deviation Z < -2 'extreme' negative deviation	individual
Rho	Pearson correlation between true and predicted responses	A measure of linear correlation between true and predicted responses. It is the ratio between the covariance of true and predicted values and the product of their standard deviations.	Ranges between -1 and 1. Closer to 1 = better model performance.	summary
pRho	Parametric p-value for the Pearson correlation	The probability of obtaining test results at least as extreme as the results actually observed, under the assumption that the null hypothesis is true.	Ranges between 0 and 1. Closer to 0 = more statistically significant.	summary
SMSE	Standardized mean squared error	The square root of the squared residual between the mean prediction and the target at each test point, averaged over samples in the test set, normalized by the variance of the targets in the test set.	Closer to 0 = better (more accurate) model performance.	summary
EV	Explained variance	The proportion to which the predicted value accounts for the variance of the true value. Sensitive to the mean fit, dependent on flexibility of the model.	Closer to 1 = better model performance.	summary
MSLL	Mean standardized log-loss	The log loss minus the loss that would be obtained under the trivial model which predicts using a Gaussian with the mean and variance of the training data, averaged over the test set. Sensitive to the variance, penalizes the flexibility of the model.	More negative = better model performance.	summary