(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
03 October 2019 (03.10.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/191784 A1

(51) International Patent Classification:
G09G 5/02 (2006.01) *G06T 11/20* (2006.01)

(21) International Application Number:
PCT/US2019/025260

(22) International Filing Date:
01 April 2019 (01.04.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/US2018/761596
30 March 2018 (30.03.2018) US

(72) Inventor; and
(71) Applicant: **PISNER, Derek** [US/US]; 111 Sandra Muraida Way, Unit 170, Austin, TX 78703 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to the identity of the inventor (Rule 4.17(i))*
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
— *of inventorship (Rule 4.17(iv))*

Published:
— *with international search report (Art. 21(3))*

(54) **Title:** AUTOMATED FEATURE ENGINEERING OF HIERARCHICAL ENSEMBLE CONNECTOMES

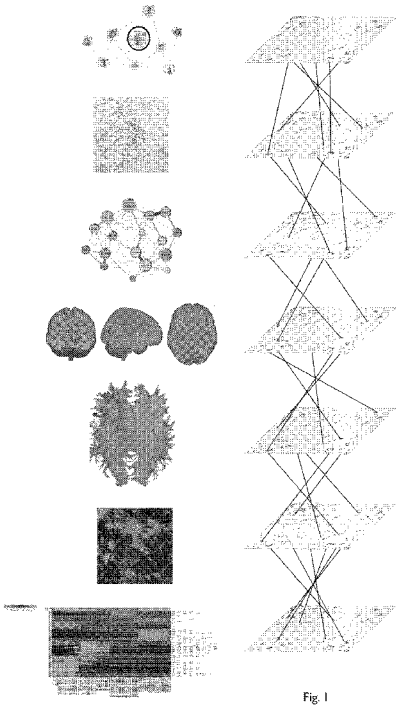


Fig. 1

(57) **Abstract:** Existing methods for analyzing person-specific 'connectomes'are not computationally equipped for scalable, flexible, and integrated processing across multiple network resolutions and drawing from disparate data modalities—a major roadblock to utilizing *ensembles* and *hierarchies* of connectomes to solve person-specific machine-learning problems. The processes implemented in software described herein consists of an end-to-end *pipeline* for deploying ensembles and hierarchies of network-generating workflows that can utilize multimodal, person-specific data to sample networks, extracted from that data, across a grid of network-defining hyperparameters. In essence, this pipeline enables users to perform ensemble sampling of connectomes for given individual(s) based on any input phenotypic datatype, constructed from any data modality or hierarchy of modalities at any scale, and based on any set of network-defining hyperparameters.

WO 2019/191784 A1

AUTOMATED FEATURE ENGINEERING OF HIERARCHICAL ENSEMBLE CONNECTOMES

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application Ser. No. 62/761,596, filed March 30th, 2018, the entire disclosure of which is hereby incorporated by reference in its entirety.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

This invention was made without outside public or private funding, and thus no additional entities may claim rights to the invention.

TECHNICAL FIELD

The included disclosure describes processes implemented in biomedical software intended for usage by experts from a variety of fields including but not limited to neuroimaging, psychology, systems biology, bioinformatics, computational psychiatry, computational linguistics, neurogenetics, ecology, cognitive science, computer science, object-oriented programming, high-powered computing, graph theory, ensemble sampling, machine-learning, and artificial intelligence.

BACKGROUND

What is a 'Connectome'?

In stride with rapid advances in data science, person-specific data has become increasingly available for making statistical predictions about *individuals*—their present behaviors, thoughts, feelings, but also their past dispositions and future trajectories. In particular, rich new sources of person-specific psychophysiological data have emerged from fields like neuroscience and psychology that could amplify these predictions well beyond what has historically been achieved. More specifically, a major paradigm shift in these fields towards the study of the human 'connectome' has revealed immense potential for driving these statistical predictions based on deep-phenotypic information about individuals represented through latent interrelationships across person-specific data. 'Person-specific' hereby refers to elements that are in some way tethered or anchored to individual persons (e.g. person-specific genetic data, person-specific machine-learning model that makes Netflix recommendations, etc.).

A *connectome,* as the term was originally defined, refers to the set of elements and connections comprising an individual's nervous system. Recently, however, this definition has been extended to encompass a broader phenomenological scope beyond the nervous system—what has sometimes been referred to as the '*connectotype*.' By this reformulation, a connectome can also be construed as encompassing the full web of phenotypic connectivity relationships among genetic, molecular, cognitive, behavioral, and even social traits. However we might formulate it, the defining feature of a connectome is that it typically exhibits a set of *network* properties with a complex topology. That is, their specific pattern of pairwise connectivity, considered independently or collectively across scales, is neither completely regular nor random. This complex topology, which can conceivably encapsulate person-specific network traits has the power to characterize deeply multivariate statistical relationships about a given individual. A *network* (or 'graph') is simply a collection of connected objects, referred to as nodes (or vertices), where the connections between these nodes are referred to as edges. A common example of a network would be a social networks where each person is a node, each friendship connecting them an edge. Networks are typically represented in the form of n x n adjacency matrices where nodes and edges can be defined in any number of ways depending on the type of data being modeled. Once a

connectome has been *estimated*, additional thresholding is typically performed (by absolute edge weight, proportion of strongest weights, and other methods) to ensure an adequate level of sparsity in the graph that penalizes spurious (i.e. 'false positive') connections. Graphs (i.e. networks) are then often, but not always, analyzed using an approach called *graph analysis*, which includes implementing some subset or all of several dozen algorithms for characterizing multiple aspects of global and local network organization. Global measures broadly refer to the network as a whole, whereas local measures broadly refer to the local properties of individual network nodes. Global measures include measures of efficiency, clustering, paths/distance, assortativity/core structure, density, degree, and community structure. Local measures include node centrality and local efficiency, among others. In the case of a layered structural-functional multigraph, centrality measures are instead referred to as "versatility." Many, but not all, global and local graph measures can be calculated for any type of graph. Whereas graph analysis may yield derivative measures of network organization that may be highly informative to a given statistical model (i.e. making them useful to obtain as an end in and of themselves), other scenarios may warrant the use of raw graphs in statistical models since the former can add unnecessarily high dimensionality under certain circumstances.

Types of Connectomes

Brain Networks

In the context of neuroscience specifically, connectomes can be used to represent a variety of brain network types. Structural brain networks, for instance, are based on some combination of white-matter, grey matter, and/or molecular properties that capture direct or indirect neural connections among disparate brain regions. These properties are typically captured at a macroscopic level using Magnetic Resonance Imaging (MRI), diffusion Magnetic Resonance Imaging (dMRI), Positron Emission Tomography (PET), Magnetic Resonance Spectroscopy (MRS), Infrared Imaging (IR), Single Photon Emission Computed Tomography (SPECT), and Computed Tomography (CT), but will likely we observable through other neuroimaging modalities as of yet undiscovered. In contrast to temporally static structural brain networks, functional brain networks, can be modeled to reflect co-activation patterns of distinct brain regions over time. These temporal patterns of neural connectivity can be inferred from Blood-Oxygen Level Dependent signal captured using functional Magnetic Resonance Imaging (fMRI) or functional Near-Infrared Spectroscopy (fNIRS), as well as scalp electrical activity measured using Electroencephalography (EEG), Magnetoencephalography (MEG), and other modalities as of yet undiscovered. Functional brain networks derived from these same modalities can also be described dynamically—i.e. as a time-series consisting of multiple functional brain networks across discrete sliding windows. This perspective is a close relative to traditional functional connectivity that is sometimes referred to as 'effective' connectivity in that it can facilitate dynamical causal modeling (DCM) or network information flow. For simplicity sake, we will group dynamic/effective brain networks within the larger umbrella of functional brain networks.

Whereas fMRI data and EEG data are typically stored as a 4-dimensional time-series consisting of 3-dimensional image volume sets, dMRI data not only includes a given diffusion image (typically in nifti, .trk, .cifti, or other similar format) consisting of b0 reference images and as many 'diffusion weighted' volumes as there are directions acquired from the dMRI acquisition, it also includes an accompanying gradient table that characterizes the magnitude and orientation of those weighted volumes, all of which is stored in two accompanying text files – the b-values and b-vectors.

Structural or functional brain networks can be represented at various resolutions of nodes and edges that can be both weighted (i.e. assigned some continuous value) or unweighted (i.e. binarized/boolean). Brain networks can be both directed (i.e. directionality/asymmetry is encoded in edges based on some directionality of the connections) or undirected (directional information is not encoded in the edges).

For both functional and structural graphs, nodes can be defined using any of several methods that include the following: 1) atlas-defined (i.e. based on some a-priori digital brain atlas composed of sulci and gyri cortical surface representations, or subcortical volumes, whereby each relevant brain region is assigned some index as an intensity value); 2) anatomically-defined (i.e. based on an individual's structural MRI image that has been digitally-parcellated into relevant brain regions); or 3) cluster-defined (i.e. based on spatially-distinct clusters of functional activation). Accordingly, nodes can be defined based on labels (i.e. represented as irregular, 3-dimensional geometrical volumes) or as spherical volumes of radius *r* whose centroids correspond to x, y, z coordinates. Importantly, network neuroscientists are often interested in characterizing the properties of sub-networks of the brain (vis-à-vis whole-brain networks), which are sometimes referred to as Resting-State Networks (RSN's) or restricted-networks. Although nodes can be defined in several ways, they can further be 'reduced by affinity' (vis-à-vis selecting a subset of nodes that fall within the spatial constraints of RSN's or some manually-defined restricted network. For example, a core set of both 7 and 17 RSN's, as defined by Yeo et al. 2011 and redefined recently by Shaefer et al. 2018[i] have become key networks of interest for both research and clinical purposes.

Edges of brain graphs, moreover, are defined using an entirely different set of techniques, depending on whether the graphs are derived from functional or structural neuroimaging modalities. In functional graphs, the edges are determined using a connectivity model 'estimator' applied to some individual X's time-series data (i.e. from fMRI, fNIRS, EEG, MEG, or another functional neuroimaging modality). Typically, these connectivity models are based on one of two primary 'families' of statistical relation—correlation and covariance. The correlation family consists of both parametric and non-parametric approaches such as Pearson's and Spearman's rho correlation and partial correlation. Importantly, the use of these correlational approaches require additional normalization schemes such as a Fisher's r-to-z transformation of graph edge weights following connectivity model estimation. The covariance family consists of both traditional covariance estimation as well as a variety of Gaussian Graphical Models (GGM), in which the joint distribution of a set of random variables is assumed to be Gaussian and the pattern of zeros of the covariance matrix is encoded in terms of an undirected graph. The most common GGM is the inverse of the covariance matrix, also called the "precision" matrix, which is inherently sparse and thereby capable of representing only *direct* (as opposed to "indirect") connections between nodes.

In structural graphs, on the other hand, edges are most commonly determined by estimating the number and/or integrity of white-matter streamlines connecting nodes, which often consist of grey matter volumetric information. This can be accomplished using either deterministic or probabilistic tractography—two common methods for using underlying directional information from dMRI (i.e. based on degree of isotropic diffusion of water molecules throughout the brain) to iteratively 'track' white-matter connections between nodes. This tracking process intimately depends both on a the type of diffusion model fit to the data, as well as the method of tractography used once the model is fit. Examples of diffusion models include but are not limited to Constrained Spherical Deconvolution (CSD), tensor, ball-and-stick, and qball. Various methods of deterministic and probabilistic

tractography also exist, the majority of which share a common set of tracking hyperparameters including step size, curvature threshold, tissue classification approach, number of samples, length threshold, and others not stated herein. As a result of this variety, dramatically different connectivity estimations can result that many have argued warrant an ensemble approach to adequately address for purposes of generating reliable connectomes from dMRI data.

Structural and functional brain networks can also be modeled at higher resolutions. At the micrometer resolution – the "microscale" – human neural networks are described neuron-by-neuron. With existing biomedical imaging technology and the daunting task of building computational models capable of accommodating for the billions of neurons found throughout the human cortex, this approach remains largely in its infancy. A "mesoscale" connectome, by contrast, attempts to model anatomically and/or functionally distinct neuronal populations at a spatial resolution of hundreds of micrometers. Even at this broader scale, however, existing neuroimaging technology is remains poorly suited for fine-grained study of brain networks. Nevertheless, as emerging nanotechnology and continual refinements to existing neuroimaging tools permit more fine-grained study of microscopic and mesoscale connectomes, highly detailed subgraphs of brain networks can be modeled and analyzed using similar graph analytic techniques to those employed on the macroscopic level.

Gene Networks

Human twin studies have shown that topological properties of brain networks, including those quantifying specific aspects of network economy, are largely heritable. In the context of genetics, the connectome is often referred to as a 'transcriptome,' which refers to the sum total of all the messenger RNA proteins expressed from the genes of an organism. Subgraphs of the transcriptome are referred to as gene regulatory networks. Both the transcriptome and gene regulatory network subgraphs can be studied using weighted gene co-expression graph analysis.

When inferring co-expression networks from high-throughput gene expression data, one usually takes as primary input the data from a set of n independent measurements of the mRNA gene expression levels and then, by using whatever correlation-based inference method, constructs the corresponding network. The mRNA measurements are carried out by means of microarray techniques, and each measurement, which is able to collect information of a very big number (p) of genes, corresponds to a particular group of cells of a certain individual. Accordingly, multiple methods exist for measuring 'gene expression levels' that can have a varying impact on the resulting gene regulatory network produced. Examples of these include: thermodynamic, Boolean, coupled ordinary different equations, stochastic, and continuous.

The more basic co-expression inference network model that one can find in the literature consists in calculating first the linear pairwise correlation coefficient r of all possible pairs of genes, and then, establishing a link between those gene pairs that show a 'large enough' value of r. The natural assumption behind this construction process is that a large value of the correlation coefficient signifies some functional relationship among the pair of genes involved. In practice, a fixed cutoff (r^2_{cf}) for the squared values of r is often used so that if r^2 is larger than the cutoff, then a link between the pair of genes is established; contrarily, if r^2 is smaller, the gene pair remains unlinked. Fundamentally, however, the value of the fixed r cutoff is a freely varying hyperparameter. Given the small probability that large values of r^2 appear by chance, one expects that this way to proceed generates co-expression networks

containing only trustworthy edges, i.e. edges that represent 'reliable' relationships among genes. Unfortunately, as a side effect, the networks thus inferred tend to contain a small number of edges, which results in a large number of isolated network vertices that, as in the case of brain networks, should be 'pruned' before extracting global or local network metrics from the graph.

Recently, moreover, research has shown that neural phenotypes (both structural and functional) can be linked to gene networks. This approach leverages brain-wide atlases of gene expression, which quantify the transcriptional activity of thousands of genes across many different anatomical locations. The broad anatomical and genomic coverage of brain-wide gene expression atlases makes it possible to comprehensively map the molecular correlates of spatially distributed network properties, thus helping to bridge the gap between the transcriptome and connectome of the brain.

Semantic Networks

A semantic network is used when one has knowledge that is best understood as a set of concepts that are related to one another. Most semantic networks are cognitively based and are often used as a form of knowledge representation. They also consist of arcs and nodes which can be organized into a taxonomic hierarchy. Semantic networks can be directed or undirected graphs consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts, mapping or connecting semantic fields. In practice, semantic networks can be computationally generated using various forms of Natural Language Processing (NLP) which can parse metadata from text information acquired from audio recordings, text messages, emails, and other sources of semantic information. This metadata can then be used as a basis for defining connective relationships between words, phrases, and ideas based on a wide variety of criteria. Some resulting network types include:

1. *Definitional networks* emphasize the subtype or is-a relation between a concept type and a newly defined subtype. The resulting network, also called a generalization or subsumption hierarchy, supports the rule of inheritance for copying properties defined for a supertype to all of its subtypes. Since definitions are true by definition, the information in these networks is often assumed to be necessarily true.
2. *Assertional networks* are designed to assert propositions. Unlike definitional networks, the information in an assertional network is assumed to be contingently true, unless it is explicitly marked with a modal operator. Some assertional networks have been proposed as models of the conceptual structures underlying natural language semantics.
3. *Implicational networks* use implication as the primary relation for connecting nodes. They may be used to represent patterns of beliefs, causality, or inferences.
4. *Executable networks* include some mechanism, such as marker passing or attached procedures, which can perform inferences, pass messages, or search for patterns and associations.
5. *Learning networks* build or extend their representations by acquiring knowledge from examples. The new knowledge may change the old network by adding and deleting nodes and arcs or by modifying numerical values, called weights, associated with the nodes and arcs.
6. *Hybrid networks* combine two or more of the previous techniques, either in a single network or in separate, but closely interacting networks.

Behavioral Networks

When investigating personality, nodes can represent cognitions, motivations, emotions, symptoms in the case of mental illness, and behavioral tendencies including geographical movement and activity patterns, that can vary across individuals or occasions. Nodes can be assessed by single items in questionnaires or interviews, or by aggregates of items, for instance personality facets or Ecological Momentary Assessment (EMA). The choice of an appropriate level of investigation (e.g., items, facets, or even broader traits) depends on which level is most useful for investigating the phenomenon of interest. For personality and psychopathology research, edge weights and directional encoding of edges are fundamental, because they allow distinguishing between intense and weak and between positive and negative associations among variables. Edge direction has been used in psychology particularly for representing temporal dependencies. Examples of sources of data in psychology include participants' rating on an object of interest (e.g., themselves, a peer, or a situation) collected only once (cross-sectional studies) or many times (e.g., as in EMA studies). Whereas networks can be computed both on cross-sectional and longitudinal datasets, disentangling the variation due to individuals' deviations from routine versus situations of novelty is critical. Although correlation networks can be used, the most common method for cross-sectional data has been to elaborate partial correlation networks, which are equivalent to standardized Gaussian Graphical Models (GGM's).

Social Networks

The study of social networks extends the notion of behavioral networks to encompass the full web of social relations of an individual in terms of verbal and non-verbal, active and passive interpersonal interactivity, including face-face, social media, and written/telephonic communications. These relations include friends, family, acquaintances, peer groups, romantic partners, colleagues, sports team members, leaders, and subordinates among other relationship types. This web of social connections can be represented as a 'sociogram' – a graphical representation of both the immediate and distal social links that a person has. Social networks can consist of direct or indirect linkages between an individual and others based upon shared attributes, shared attendance at events, or common affiliations.

Why is the Study of Connectomes Useful?

One key reason that the use of network features might amplify the precision of machine-learning in general is that networks by definition encapsulate a wider scope of information through hierarchical sets of relations. This heightened *complexity*, provided it is balanced by sufficiently large training-sets, confers a favorable *inductive bias* that can be exploited by the learning algorithms to augment the training process. In contrast to traditional 'black-box' machine-learning approaches, such inductive biases specifically serve to scaffold the learning process by feeding it *structured data* based on *a priori* knowledge that makes model training more efficient, reduces ambiguity in the 'solution space,' and allows the learning algorithm to make more informed and accurate predictions. the incorporation of these relational inductive biases have in fact already proven wide utility. They have proven immensely useful, for instance, in the identification of spammers across social media networks, as well as uncovering cybersecurity threats through network flow clustering techniques.

The inclusion of network features in machine-learning is particularly effective when the underlying data exhibits rich relational structure, as is fundamentally the case with respect to the human connectome. Consequently, the inclusion of connectomes as machine-learning features could provide the learner with a capacity for greater *combinatorial generalization*— i.e. constructing new inferences, predictions, and behaviors from known building blocks.

Given the power of networks to model high information complexity (i.e. 'dimensionality') from data, connectomes that represent networks about individual persons may provide an enriched source of person-specific features for machine learning problems concerned with making predictions about individuals. In particular, connectome features may be especially useful for augmenting machine-learning to more accurately forecast individual outcomes. This can be accomplished, for instance, by 'embedding' the relational properties of connectome networks into the feature vectors used to train machine-learning models. Alternatively, the derivative network organizational measures resulting from graph analysis (i.e. many of which are scalar values) can themselves be used as features. Though still nascent, one particularly promising avenue may be the use of connectome-embedding or 'graph-analytic embedding' to dramatically enhance the precision of diagnostic and treatment-matching algorithms in machine-learning of computational medicine.

Enhancing Person-Specific Machine-Learning Through Ensemble Connectomics

To complicate matters, however, emerging research has shown that the networks that make up individual connectomes are largely unstable models. One reason for this is that the process of estimating networks often hinges on a number of freely-varying *hyperparameters*, some of which are unique to each data modality and others of which are common to all modalities (e.g. sparsity threshold). Because there is often no definitive rationale for selecting specific values for these hyperparameters, traditional methods of network analysis often attempt to bypass this limitation by, for example, performing hypothesis-testing with ROC curves or via simulation. Outside of pure mathematical contexts (e.g. deriving graphs from biomedical data where data processing often occurs over many sequential steps), however, more of these freely-varying hyperparameters can be found in the steps preceding network analysis. Accordingly, analyses that do not attempt to comprehensively 'tune' these hyperparameters serve as a threat to the reproducibility of the connectomes that they generate. If machine-learning algorithms are trained using features based only on such unitary network estimations, the algorithms will be less generalizable to new data where optimal set of network hyperparameters may naturally take a different 'shape.'

Consequently, new software is needed for the generation and analysis of connectomes that has the unique capability to perform *ensemble sampling* of connectome networks. Following connectome generation, those connectomes corresponding to a given combination of hyperparameters can then either be selected or discarded based on any of a variety of *feature-selection* approaches. For instance, a hyperparameter 'grid-search' can be conducted with *cross-validation* to determine those networks that maximally contribute to machine-learning model prediction. Networks can even be selected at random from the sampled ensemble, filtered by some minimum variance and collinearity threshold. In more complex cases, Bayesian techniques such as Markov-Chains can be used to guide the selection of ensemble graphs based on known-priors about the impact of unique hyperparameter combinations on network reproducibility. Other consensus methods for analyzing graph ensembles include simple central tendency measures across derivative global and local graph statistics, the encapsulation of these statistics in 'hypergraphs' where the hyperparameters themselves are treated as network nodes, optimization techniques for maximizing the identifiability of a given graph metric or vector of metrics, and other statistical approaches as of yet undiscovered. Common to each of these approaches, however, is the idea that by incorporating multiple networks (i.e. referring the same person, but from different vantage points defined by different combinations of hyperparameters) into *ensemble connectomes*, more stable (and hence *generalizable*) machine-learning algorithms could be developed while accommodating for very high-dimensionality network features.

To accommodate for ensemble connectomes features in machine-learning more directly, a special variety of *ensemble machine learning* algorithms may be particularly appropriate such as Random Forests or Gradient Boosting Machines. In popular machine-learning competitions, ensemble methods such as these often produce more accurate solutions than a single model would. The advantage of these methods is that they provide a collaborative filtering algorithm through an ensemble of 'weak learners'. Examples of ensemble machine-learning methods, in a context where ensemble connectomes are used, might include majority or weighted voting (i.e. for classification problems) or simple or weighted averaging (i.e. for regression problems) of the connectomes. With a machine-learning technique called Bootstrap Aggregating (also known as 'Bagging'), moreover, multiple predictive models can be generated based on multiple random subsamples of the connectome ensemble using bootstrap resampling. Importantly, however, ensemble connectomics does not necessitate that ensemble machine-learning algorithms be subsequently used; rather, in certain cases, deep learning and other alternative methods may be warranted whereby the consensus aggregation of the connectome ensemble occurs as a preliminary feature-selection step instead of during the model training process itself.

A More Robust Ensemble Connectomics Permits the Use of 'Hierarchical' Connectome Features

With precision psychiatry in particular – a field that operates based on a fundamentally wide informational scope in its diagnostic and treatment decision-making – machine-learning efforts have so far proven limited in that they rely primarily on unimodal features that are largely subjective. Future inclusion of connectome-enriched features taken from disparate data modalities relating to individuals (e.g. genetics, brain connectivity, social networks, etc.) might bolster the precision of machine-learning about persons considerably, particularly if the generalizability of such machine-learning is improved through ensemble sampling. In particular, both global and local properties of connectomes can be represented, analyzed, and studied at various levels of complexity – as both unitary and hierarchical networks. Although unitary networks may be informative, hierarchies of networks should theoretically yield even more information. By this notion, the *aggregation* of multimodal networks (i.e. 'multidimensional connectotypes') anchored to the same latent construct—the *person*—might lend to training more information-exhaustive models with an even greater capacity for combinatorial generalization. Hence, if connectome networks unique to a given modality contain overlapping or complementary information with other networks generated in their ensemble, and/or networks across independent modalities contain comparable node-edge definitions, these can be combined hierarchically into so-called 'multigraphs'. These hypergraphs can be used as additional features that capture some further level of emergent complexity beyond that which is available when relying on each modality-specific graph considered independently. By analogy to the field of Astroinformatics, this would be akin to analyzing networks of observable star systems within the same galaxy versus hierarchical relations of galaxies to one another. Hierarchical network analysis, supported by the stabilizing effect of ensemble sampling, provides a potentially invaluable method for embedding connectomes into machine-learning models. To make this possible, however, novel connectome-generating tools are need to produce network-enriched features of multiple variations, resolutions, and scales that can collectively be incorporated into the statistical learning algorithms available through Torch, H2O, Scikit-Learn, TensorFlow, or other emerging machine-learning libraries.

A New Computational Framework is Needed to Generate Ensemble Connectomes

If 'enriched' connectome features are to theoretically draw from multiple layers of individual data and across grids of hyperparameters, the compute resources needed to perform such ensemble sampling will necessarily be greater. Connectome features derived from structural and functional brain data, for instance, have already proven difficult to obtain due to the often unwieldly computational expense of processing such data. Likewise, microscopic gene assays and macroscopic social networks often involve millions of data points whose information is virtually meaningless without computationally rigorous forms of data munging and dimensionality reduction. Consequently, the next wave of connectomic analysis will require immense computing power and a *pipelining* framework for graph generation and analytics that can accommodate. Due to the multiple stages of data interpolation typically involved in the generation of various domain specific graphs, moreover, there is also a higher risk for computational error and noise accumulation that threaten the integrity of features produced through automated data processing. Thus, issues related to computational reproducibility stemming from the use of fully-automated, ensemble connectome estimation methods is of central importance, but are also not yet well-addressed using existing technology. In sum, new software is needed that is capable of providing an unprecedented level of automaticity, parallelism, scalability, flexibility, and statistical reproducibility for connectome generation and pooling across levels of individual data analysis.

The processes implemented in biomedical software as proposed herein constitute a *network feature-engineering* tool, but the disclosed invention specifically outlines a pipeline of processes that can accommodate for the unique feature-engineering needs of ensembles connectomes specific to data acquired about individual persons. Notably, other tools presently exist for automated feature-engineering (e.g. Deep Feature Synthesis (DFS)), but none of these explicitly handle network data. Nilearn, the Brain Connectivity Toolbox, the Connectome Visualization Utility, and other related open-source research software, provide libraries of tools that can be used for network estimation, graph-analysis, and machine-learning, but they do not provide a workflows for performing these operations in the manner described in the claims herein. By contrast, NeuroData MRI Graphs (NDMG), Configurable Pipeline for the Analysis of Connectomes (CPAC), and the Connectome Mapper (CMTK) provide end-end workflows for generating graphs based on structural and/or functional brain networks, but these tools are limited to processing neuroimaging data only (e.g. lack support for social, gene, or cognitive networks); these workflows also do not provide the ability to perform ensemble sampling and consensus analysis of graphs based on grid-searches of hyperparameters. By design, the proposed processes implemented in software as described in the subsequent claims overcomes each of the aforementioned limitations of existing tools through the incorporation of *meta-programming* into the network feature-engineering process. That is, the disclosed pipeline processes actually consist of a 'meta-workflow' that anchors several *nested workflows*, allowing for at least four layers of parallelization (across individuals, across data modalities, across hyperparameters, and across modality-specific vectorized functions) that consist of highly modular 'nodes' of workflow operation. In fact, the disclosed processes implemented in software represent the hyperparameter iterable expansion of all user-specified hyperparameter combinations. Consequently, the proposed series of processes constitute the only existing technology capable of automatically optimizing its own resource scheduling to accommodate for its added computational burden. By design, the proposed invention is also interoperable with any existing or future graph-generating workflow unique to other data modalities not stated herein. Such third-party graph-estimation workflows can optionally be included as 'plugins' through external software dependencies such that those workflows too can be used to samples graph ensembles based

on other person-specific data modalities not stated herein. For the sake of demonstrating this workflow-selection API functionality, the claims described in the present disclosure showcases one such modality-specific sub-workflow (i.e. for generating consensus networks from functional MRI (fMRI) data) that can be executed using the disclosed pipeline. Importantly, we do not wish to make explicit claims with respect to the latter technology in terms of specific content; rather it is our intention to mention these plugins with respect to their novel *application* as a building-block of a pipeline distinctly designed to execute such plugins iteratively and in tandem in the production of ensembles connectomes—the primary focus of our disclosure. Importantly, the present disclosure does not attempt to make claims with respect to the abstract idea of ensemble connectomes; rather, its singular aim is to protect a unique combination of computer-executed processes required to practically implement this idea. Such technology is not presently available nor is it likely to be achievable in the absence of tremendous programmatic innovation. That innovation here consists of a recipe for fully-automated and massively parallel network estimation routines, integrated across any person-specific data modality, and with the flexibility to vary any or all sets of network hyperparameters. Although the technology's name is currently trademarked, and the associated machine-instructed syntax is copyrighted, the processes the constitute the fundamental innovation require protection.

Equipped with the disclosed technology, the proposed invention will allow for the generation of more robust hierarchical and ensemble connectome features, the majority of which have never been created or studied before simply as a result of their computational inaccessibility. Further, the connectome outputs produced by the disclosed processes implemented in software will constitute a rich source of readily-available, person-specific network information that can in turn be used to enrich person-specific machine-learning algorithms.

BRIEF DISCRIPTION

Broadly, the proposed processes implemented in software involves a four-stage process of graph generation and analysis, the combination of which could theoretically be implemented using any object-oriented programming language. For ease of development, proof-of-concept, and usability, these four stages have been so far implemented using the Python programming language, with elements of Javascript (e.g. D3, node.js), and C++, with SQL, and R statistics interfaces as well. The choice of Python syntax allows for maximal interoperability with existing and future workflows that produce person-specific graphs from raw data. This prototype implementation (PyNets™) relies on several key package dependencies, all of which are open-source, but most heavily draws from: 1) Nipype (http://nipy.org/packages/nipype/index.html), which is an open-source Python package that serves to provide a uniform interface to existing neuroimaging software and facilitates interaction between these packages using a number of workflow development utilities that support distributed, object-oriented computing, along with detailed logging and debugging utilities; 2) Networkx (https://networkx.github.io/), which is an open-source Python package for the creation, manipulation, and study of the structure, function, and dynamics of complex networks in an object-oriented framework.

The four-stage workflow broadly consists of the following:
Stage 1 (**23**): User input and parsing to configure and initialize a 'meta-workflow' of workflows.
Stage 2 (**22**): Modality-specific workflow selection (i.e. nested within the 'meta-workflow').
Stage 3 (**3**): Fully-automated graph analysis of all resulting networks produced by Stage 2.

Stage 4 (**98**): Aggregate, summarize, and provide consensus for all networks produced from Stage 2 and all derivative graph metrics produced from stage 3.

By default, data inputs to stage 1 are assumed to be maximally noise-free and *preprocessed.* While there are no strict requirements for what constitute an adequate level preprocessing, the assumption is that this level aligns with standard accepted practices, specific to the modality, for raw data processing in preparation for network analysis.

All or part of the technology disclosed herein may be implemented as a computer program that includes object-oriented instructions stored on one or more non-transitory machine-readable storage media, and that are executable on one or more processing devices. All or part of the systems and techniques described herein may be implemented as an apparatus, method, or process that may include one or more processing devices and memory to store executable instructions to implement the stated functions. The details of one or more implementations are set forth in the accompanying drawings and the descriptions that follow. Other features, objects, and advantages will be apparent from the description and drawings, and from the specific claims. To provide concrete examples of the technology's use in context, we additionally provide a set of two embodiments in the final section of the specification.

DESCRIPTION OF DRAWINGS

FIGURE 1 is a visualization of a hierarchical multigraph spanning seven independent data modalities for which the disclosed invention is equipped to accommodate (*from bottom to top*: genetic transcriptome/ gene regulatory network, molecular neural network, microstructural brain network, functional brain network, cognitive/semantic network, behavioral network, social network). Not all of these modalities currently have a dedicated sub-workflow in the existing software prototype (only functional and microstructural brain networks), but graphs generated from third-party software for these additional modalities can be included as raw graph inputs to the workflow for multiscale analysis. There may also be other person-specific data modalities that are discovered and/or developed in the future but are not here disclosed that could be included in this hierarchy.

FIGURE 2 is a Directed Acyclic execution Graph (DAG) that depicts the sequence of operations and parallelism *anticipated* at the instantiation of the run script. Here, each flow block corresponds to classes for an exemplar ensemble connectome workflow for a single individual for a single modality (fMRI). The included classes of the workflow are both modularized across workflow 'layers' which reflect various degrees of workflow 'nesting' and parallelism in a meta-programming framework. Here, background greyscale intensity reflects underlying nesting dimensions—i.e. *white* corresponds to single-subject parent workflow input data, analysis, output data, and aggregation of output data; *dark grey* correspond to 'meta-workflow' that chains, synchronizes, and aggregates domain-specific workflows corresponding to separate data modalities (e.g. functional, structural, or genetic connectome workflows, etc.); *light grey* corresponds to a nested modality-specific sub-workflow. The multiple lines depicted across this figure reflect a mapping of 'iterable' hyperparameters specified at runtime that propagate as spawning threads of downstream stages in the workflow. When iterated, some parameters trigger further 'exponential' spawning of threads from upstream inputs, and are here depicted as grey boxes rather than ovals.

FIGURE 3 is a flow diagram that all four stages of the workflow, with sub-stages, parameters, inputs, and outputs included at the level of function classes, with process descriptions included where appropriate (listed numbers highlighted rectangular boxes adjacent to flow blocks correspond to numbers as referenced in the detailed description from the section that follows).

DETAILED DESCRIPTION

The four-stage workflow that constitutes the present disclosure proceeds programmatically as follows:

In Stage 1 (**23**), a variety of inputs (boolean, string, and numeric) are specified by the user (**157**)(**153**) as command-line options or through a Graphical User Interface (GUI) (**156**)(**147**) on a cloud or dedicated server. This can occur within a local, remote, or containerized environment (**154**)(**155**)(**158**), and involves running one of several available initialization scripts responsible for building and executing the workflows at runtime (**4**). The types of options (**7**)(**135**)(**9**)(**146**) that can be specified by the user during this pre-runtime configuration stage (**1**), (**133**), (**134**), (**135**), (**136**) include setting modality-unspecific hyperparameter values (**22**)(**24**) as well as modality-specific hyperparameter values (**122**)(**123**). Options also include the specification of file paths (**11**)(**2**)(**60**)(**58**)(**142**)(**59**) whose inclusion are used to determine the logic of the workflow stages that are subsequently instantiated (**61**). These options are then parsed at runtime and assigned to variables, which then further instantiate and configure the appropriate workflow through a set of conditional triggers whose logic consists of hard-coded parameter compatibility rules (**5**). All invalid combinations of runtime parameters (e.g. hyperparameters specific to one data modality are specified when no valid file input from that modality was correspondingly supplied by the user) are either ignored (i.e. in the case that other valid combinations of options are included) or preset to raise anything from a warning (**177**)(**178**) (i.e. in the case of unexpected behavior from given combination of parameters) to errors in the extreme case of total incompatibility of files/parameters specified (**175**)(**176**). Command-line inputs (both hyperparameters or file paths) can be applied to data for single individuals (**6**)(**12**)(**16**), or in 'batch' across multiple individuals (i.e. if file paths to multiple datasets are provided as input) (**6**)(**14**)(**17**)(**18**)(**19**). By default, minimally required file inputs must be specified manually with explicit path strings (i.e. or browse option in the case of GUI usage), but can also be auto-detected using the metadata of structured databases (**146**) whose format follows an established or custom specification protocol (e.g. Brain Imaging Data Structure)(**4**). In this latter case, the user-specified options which are unrelated to file inputs (e.g. hyperparameter selection) will be applied universally to all data present in the specified data structure (**139**). Moreover, single or multiple files available through the standardized data structure specified will be identified based on the user-specified list of unique identifiers and if they exist will be passed as lists of file inputs in a manner that mimics the manual specification of files (**4**)(**6**)(**23**)(**25**).

Regardless of the approach taken to configure the workflow(s) for runtime, resource-handling (**78**) is also addressed initially by way of the initialization script (**10**) and user-specified runtime options (**11**). Although the choice of linear or parallel workflow execution must be stated explicitly by the user (**139**) as a precaution for preventing system overloads, the choice of scheduler plugins (**80**), CPU/GPU/Memory restrictions of the workflow (**79**) can either be auto-set (i.e. by detecting available resources to utilize), manually overridden (based on user specification), or auto-set with accompanying dynamic job-scheduler optimization. With respect to the latter option, ongoing, automated network flow optimization analysis is used to assign 'nice' values of job priority at each opportunity for parallelism that emerges

throughout workflow execution (**86**) and based on regularly sampled resource consumption data from resource profiling (**81**). This option is available because the initialization script instantaneously produces a directed acyclic execution graph (DAG)(**12**)(*See example in* **Figure 2**) that can be exploited for network flow optimization analysis by dynamically referencing incoming resource profiler data against known priors of resource consumption stored in the DAG's attributes. The flow network analysis is useful in this scenario due to what can often be an overwhelming computational load when generating connectome ensembles. To ameliorate this, the network flow optimization aims to balance external supply of compute resources with the greedy compute demands consisting of a set of scheduled 'job' that require some overarching prioritization heuristic to ensure maximal efficiency (**88**). Accordingly, several user-defined objectives can be used for the flow optimization, including but not limited to: minimizing compute dollar spending (**170**), minimizing overall runtime duration (**171**), and minimizing computational-load (**89**).

Aside from a unique identifier (**24**), two core runtimes parameters (**21**) specified by the user (**137**) during Stage 1 (**23**) universally for all types of data modalities are: a) connectivity model type (**27**); and b) graph thresholding approach (**25**). With respect to (a), any one of a variety of connectivity models can be used, including covariance (**29**), correlation (**30**), and/or a modality-specific definition (**43**). The covariance family (**29**) includes empirical covariance (35), Ledoit-Wolfe 'shrinkage' covariance (**37**), and graphical lasso methods (**36**), which includes various implementations of L1 and/or L2 penalized covariance (**38**)(**39**)(**42**)(**39**)(**166**). The correlation family (**30**) includes Pearson's correlation (**32**) and L1 and/or L2 penalized forms of 'partial' correlation (**33**). If a method from the correlation family (**30**) is indicated, then **130** in Stage 3 (**121**) will automatically include a Fischer's r-to-z transformation (**102**) to standardize the resulting graph edge weights. Lastly, modality specific definitions of connectivity (**43**) can also be used beyond correlation/covariance. Examples of these include fiber count (**50**) and fiber integrity (**49**) for the dMRI modality, as well as spectral coherence (**52**) and phase synchrony (**53**) for the EEG/MEG modality. Ultimately, multiple connectivity model types can be specified for the connectivity model (**27**) options, in which case they would be indicated by the user at (**1**)(**138**) as a comma-separated list of length l_{conn}, in which case Stages 2-3 (**22**)(**3**) will be reinitiated for each of the graphs produced through the expanded $l_{conn} * p!$ where $p!$ denotes all combinations of connectivity model options.

In contrast, graph thresholding hyperparameters (b) (**25**) consist of multiple sub-parameters including *types* (**65**) and *schemes* (**66**) that can be specified. *Types* (**65**) include: both global and local thresholding (**69**). Global thresholding consists of proportional thresholding (**67**), absolute thresholding (**68**), density thresholding (**118**). Local-thresholding (**10**) further consists of multiple techniques for network reduction. These include but are not limited to a disparity filter or thresholding via the Minimum Spanning Tree (MST) (**74**). For each of the three forms of global thresholding (**67**)(**68**)(**118**), various thresholding *schemes* can be used. These include singular thresholding (**71**), no thresholding (**75**), iterative thresholding (**70**), and cross-modal adaptive thresholding (**72**) in the case that multiple modalities of data were provided in Stage 1 (**23**). Iterative thresholding (**70**) requires the additional input of minimum, maximum, and step interval values to generate a window of unique combinations of multiple thresholds over which to iterate the selected thresholding type(s). By this approach, the length l_{thresh} of the resulting window of thresholds will be multiplied by the other cumulative iterable parameters specified in Stage 1 (**23**), and separate graphs (i.e. for each unique combination $l_{thresh} * p!$) will be produced by reinitiating Stages 2-3 (**22**)(**3**) accordingly.

Other possible hyperparameters specified by the user at runtime (**138**)(**126**)(**127**) are modality-specific, but impact a variety of fundamental characteristics of how the nodes and edges of a graph can be defined (i.e. the number of nodes or sizes of nodes). Ultimately, each of these hyperparameters specified and parsed in Stage 1 (**126**)(**23**) trigger (and are passed into) Stage 2 (**22**), which involves generating domain-specific graphs by spawning corresponding domain-specific 'nested' workflows (**34**)(**51**)(**60**)(**91**)(**54**)(**31**)(**90**)(**57**) that serve to produce graphs using the various procedures specific to each unique modality.

Even when singular or multiple pre-generated graphs are fed into the workflow directly by the user (a functionality available with the proposed framework that essentially bypasses Stage 2 (**22**) altogether), these are still passed to a nested workflow in Stage 2 (**22**) intended to perform minimal post-processing of the graphs (e.g. thresholding) prior to the graph analysis in Stage 3 (**3**). In the case that data from multiple modalities is included as input to the workflow and the user specified a valid 'multigraph analysis' option at runtime (e.g. multiscale graph correlation, adaptive thresholding for motif-optimization, etc.), those routines are also carried out using this post-processing nested-workflow in Stage 2 (**22**), and made available through child processes as addendum data for Stage 3 (**3**) graph analysis where they will be analyzed as multigraphs. In the case that raw graph *outputs* are all that are of interest to the user, the workflow can also be configured to terminate prematurely following Stage 2 (**22**).

The graphs produced from the nested workflows in Stage 2 (**22**) are then used as inputs for stage 3 (**3**). This Stage first involves loading the ensemble of graphs generated from Stage 2 (22), optionally standardizing graph edge weights (z-scoring, log10, or confound regression) (**34**), and converting the graphs into network 'objects' (**104**) that can in turn be analyzed either as single-layer graphs (**106**), multi-layer (i.e. hierarchical) graphs (**105**), or both. Automated graph analysis is then performed using a variety of conventional global and local graph analysis algorithms (**107**)(**108**)/(**111**)(**112**), some of which will be hard-corded to skip if they are deemed non-applicable *a priori* (e.g. in the case of disconnected graphs with isolated vertices where the 'pruning' option is not specified at runtimes). Some graph metrics are likewise hard-coded to be triggered only in the presence of multimodal data inputs (e.g. versatility (**113**) measures of genetic-brain structural multigraphs). These measures are calculated iteratively during the fully-automated graph analysis in Stage 3 (**3**) for each of the unique graphs produced in Stage 2 (**22**) for each unique combination of hyperparameters specified by the user in Stage 1 (**23**). The graphs and other derivative and temporary files produced in Stage 2 (**22**), along with the graph measures calculated in Stage 3 (**3**), are saved to disc as lightweight, compressed file objects to avoid being held in memory cache while minimizing required disk space (i.e. in the case where large graph ensembles are to be produced (**109**)(**110**)(**114**)(**115**). These files can be saved as SQL-like entries, in simple text-based formats (.csv, .ssv, .tsv, .txt, .pkl, and others not disclosed herein). Consequently, a dictionary of associated Stage 2 (**22**) and Stage 3 (**3**) file output locations is made available to all nodes of the parent and nested workflows so that they can be parsed as needed at subsequent stages of execution.

Ultimately, the dataframes of derivative graph measures (**109**)/(**114**) produced in Stage 3 (**3**) become inputs for the consensus analyses that can be performed in Stage 4 (**98**). In stage 4, the graph measures are aggregated from each unique graph in the current workflow's ensemble that successfully spawned from Stages 2-3 (**22**)(**3**). Any of a variety of user-configurable methods of consensus analysis (**101**)(**100**)(**123**)(**99**)(**102**) can then be used to generate a single annotated summary dataframe that is ultimately saved to disk (**178**).

Depending on the user-specified methods of consensus analysis, this dataframe can include a simple aggregated database (**98**) of graph measures produced from Stage 3 (**3**), a database of basic measures of central tendency (e.g. mean, median, mode) across each unique graph measure calculated for each graph in the ensemble (**115**). The graphs included in these consensus calculations encompass by default all available from the produced ensemble, but could optionally include sub-sample(s) of the ensemble based on Bayesian priors (**119**)(**120**) and/or hierarchical feature selection (**121**), or using an optimization framework (**102**) such as discriminability, identifiability, or some other objective specified by the user (**123**)(**124**). Graph robustness can also be assessed (**99**) to evaluate the ensemble's generalizability and consistency (**117**). Importantly, each dataframe that can be produced includes headers labeled by each hyperparameter combination so as to provide a reference for each corresponding graph measure (**116**), along with a profile of hyperparameter-dependent variance of each graph measure. This information is critical for use in cross-validation procedures (**122**) integral to machine-learning conducted by third-party tools beyond the scope of this disclosure. The final summary dataframe is ultimately written to disk, signifying termination of all layers of workflows (**142**).

EXAMPLES OF EMBODIMENTS

(1) A user has a noise-reduced functional MRI (fMRI) data for a given individual with an arbitrary unique identified of 997. That dataset, called "filtered_func_data_clean_standard.nii.gz" is in 'nifti' format and here consists of 182 volumes of 3D image matrices whose point values represent signal intensity of hemodynamic response as captured from the MRI machine. The user also has the individual's noise-reduced gene expression data in the form of a .txt file called 'expression_data.txt' where rows correspond to a given genetic marker and columns consist of *n* independent measurements of the mRNA gene expression levels for each corresponding genetic marker. The user wishes to generate hierarchical ensemble connectome features based on this multimodal genetic-fMRI data that can be used for subsequent ensemble machine-learning predictions of the given individual's likelihood of Alzheimer's disease. In particular, research has shown that functional brain network disturbances of the so-called 'Default-Mode Network' (DMN) is a biomarker of interest for early detection of Alzheimer's disease, particularly when APOEε4 gene is more central to an underlying gene regulatory network. To perform an ensemble sampling of connectomes for the given individual, both the fMRI and genetic data files are used as inputs to the software.

Given that fMRI measurements of DMN connectivity have been shown to vary considerably across brain atlases, network sparsity, and connectivity model, the user wishes to perform an ensemble sampling across several relevant hyperparameters. These include five different brain atlases (e.g. 'coords_power_2011', 'coords_dosenbach_2010', 'aal', 'desikan', 'CPAC200'), but also threshold the resulting networks iteratively to achieve multiple target densities of (e.g. 0.2, 0.3, 0.4, 0.5), as well as defined connectivity in a number of ways (e.g. correlation, covariance, L1-penalized covariance, L2-penalized partial correlation). Similarly for the gene regulatory network, multiple fixed 'r' minimum thresholds (e.g. 0.4, 0.5, 0.6, 0.7) are used to filter only the most important connections so as to sample across a wide hyperparameter window of that which constitutes 'meaningful' regulatory connections. Likewise, to ensure that the genetic connectome is not biased to any one particular model of connectivity, several gene network models are specified (e.g. boolean, stochastic, and continuous). To automatically engineer a set of hierarchical ensemble connectome features based on these two modalities, the user specifies the aforementioned hyperparameters and file inputs on the command-line of a Linux operating system running on a 500-CPU cloud server

hosted by Amazon Web Services (AWS). The user wants to ensure that the software's execution is optimized for minimal runtime duration, that it uses all 500 cores, restricts overall memory usage to 500 GB, and verbosely logs all processing at runtime. That command-line call appears as follows:

```
pynets_run.py -fmri 'filtered_func_data_clean_standard.nii.gz'
-gene 'expression_data.txt' -id '997' -atlas
'coords_power_2011,coords_dosenbach_2010,aal,desikan,CPAC200'
-model 'corr,cov,L1cov,L2partcorr;bool,stoch,cont' -n 'DMN' -
min_thr '0.20;0.40' -max_thr '0.50;0.70' -step_thr '0.10;0.10'
-pm '500,500' -optimize 'time'
```

Here, pynets_run.py is the pre-runtime configuration script that triggers the main workflow consisting of a series of nested sub-workflows (i.e. one specific to each modality—in this case, a sub-workflow for handling the fMRI data and a sub-workflow for handling the genetic data). Command-line 'flags' are used to indicate various file modality types, and commas are used to separate 'iterable' hyperparameter values (semi-colons enable these hyperparameter windows to be specified per modality in the order of modality-specific data files specified). After each sub-workflow runs, there should be an ensemble of 5 x 4 x 4 = 80 different networks produced for the fMRI modality along with a separate ensemble 3 x 4 = 12 different networks for the genetic modality (based on all specified combinations of hyperparameters). In sum, Stage 2 of the workflow would generate 80 + 12 = 92 unique modality-independent networks that can be further cleaned and analyzed in Stages 3 and 4. Furthermore, Stage 3 will by default aggregate the 80 and 12 networks per modality into an additional ensemble of 'hierarchical' gene-fMRI network multigraphs, yielding 80 x 12 = 960 possible unique combinations of all hyperparameters across modalities. All 1,052 networks (modality-independent and modality-intersecting) with be run through automated network analysis, the results of which along with the networks themselves, will be saved to disk, both with independent (Stage 3) and summary (Stage 4) dataframes. Files with the suffix '_neat.csv' within the fmri subdirectory will contain the network properties for each of the 80 combinations of fMRI hyperparameters. Files with the suffix '_neat.csv' within the gene subdirectory will contain the network properties for each of the 12 combinations of genetic hyperparameters. Files with the suffix '_neat.csv' within the fmri_gene subdirectory will contain the network properties for each of the 960 combinations of fMRI-genetic multigraph hyperparameters. Files with the suffix '_mean.csv' within the base subject directory contain consensus measures along with a dictionary of file paths and dataframes of each graph measure across all hyperparameters across both modalities. The latter can in turn be fed into an ensemble classifier to make highly precise predictions about that individual's likelihood of developing Alzheimer's disease. FIGURE 2 depicts the fMRI components of the workflow, with accompanying description in the subsequent section.

(2) A user has a multi-subject dataset consisting of 10 individuals with noise-reduced functional MRI (fMRI) data in 'nifti' format (here consisting of 182 volumes of 3D image matrices whose point values represent signal intensity of hemodynamic response as captured from the MRI machine), along with diffusion MRI (dMRI) data (here consisting of a b-values text file, and b-vectors text file, and a 'nifti'-formatted file containing 72 'directional' volumes of 3D image matrices whose point values represent signal intensity of water diffusion as captured from the MRI machine). The user wishes to generate hierarchical ensemble connectome features for all individuals in the given dataset based on this multimodal dMRI-fMRI data that can be used for subsequent ensemble machine-learning

predictions to determine their likelihood of responding favorably to a new experimental medication for their Major Depressive Disorder. In particular, research has shown that functional brain network disturbances of the so-called 'Executive Control Network' (ECN) is a biomarker of interest for predicting treatment response to mood disorders like depression, particularly when underlying structural brain network information is also available to characterize the brain's neuroplasticity (i.e. its adaptive potential to reshape neural connections based on different responses to stimuli). To perform an ensemble sampling of connectomes for the given individual, both the fMRI and dMRI data files, for each subject, are used as inputs to the software. Although at first the user was concerned that he would have to process each individual's data in separate software executions, he learns that the given dataset is actually formatted accordingly to the universal standards established by the Brain Imaging Data Structure (BIDS) specification. Accordingly, the entire BIDS-formatted dataset lives in his home directory: /home/users/user1/clinical_MRI_data. Subjects in the dataset have the following uniquye identifiers: 01, 02, 03, 04, 05, 06, 07, 08, 09, 10. Consequently, the user will be able to process all of the individuals' data simultaneously (i.e. in a single execution) to produce the hierarchical ensemble network features that he needs to make his machine-learning predictions.

Given that fMRI measurements of ECN connectivity have been shown to vary considerably across brain atlases, network sparsity, and connectivity model, the user wishes to perform an ensemble sampling across several relevant hyperparameters. These include five different brain atlases (e.g. 'coords_power_2011', 'coords_dosenbach_2010', 'aal', 'desikan', 'CPAC200'), but also threshold the resulting networks iteratively to achieve multiple target densities of (e.g. 0.2, 0.3, 0.4, 0.5), as well as defined connectivity in a number of ways (e.g. correlation, covariance, L1-penalized covariance, L2-penalized partial correlation). Similarly, structural brain networks derived from dMRI data are known to vary as a result of several hyperparameters that impact tractography. Hence, the user will include a list of multiple step sizes (e.g. 0.1, 0.2, 0.3) and curvature thresholds (e.g. 3, 6, 12, 24, 48) to constrain the tractography, in addition to the same multiple definitions of brain atlas and threshold window used for the fMRI data. Unlike with the fMRI data, however, the connectivity type for the dMRI data is measured beyond statistical covariance measures (e.g. fiber count, fractional anisotropy).

To automatically engineer a set of hierarchical ensemble connectome features based on these two modalities, the user specifies the aforementioned hyperparameters and file inputs on the command-line of a Linux operating system running on a 500-CPU cloud server hosted by Amazon Web Services (AWS). The user wants to ensure that the software's execution is optimized for minimizing dollar compute cost when using all 500 cores, restricting overall memory usage to 500 GB, and verbosely logging all processing at runtime. That command-line call appears as follows:

```
pynets_run.py -bids /home/users/user1/clinical_MRI_data -fmri
-dmri -id '01,02,03,04,05,06,07,08,09,10' -atlas
'coords_power_2011,coords_dosenbach_2010,aal,desikan,CPAC200'
-model 'corr,cov,L1cov,L2partcorr;fibercount,fa' -step_sz
'0.1, 0.2,0.3'-curv_thr '3,6,12,24,48' -n 'ECN;ECN' -min_thr
'0.20' -max_thr '0.50' -step_thr '0.10' -pm '500,500' -
optimize 'cost'
```

Here, pynets_run.py is the pre-runtime configuration script that triggers the main workflow consisting of a series of nested sub-workflows (i.e. one specific to each modality—in this case, a sub-workflow for handling the fMRI data and a sub-workflow for handling the dMRI data). Command-line 'flags' are used to indicate various file modality types, and commas are used to separate 'iterable' hyperparameter values (semi-colons enable these hyperparameter windows to be specified per modality in the order of modality-specific data files specified). Additionally, those flags the specify hyperparameters specific to each modality (i.e. non-overlapping) need not include semi-colons. After each sub-workflow runs, there should be 10 separate ensembles (per individual subject) consisting of 5 x 4 x 4 = 80 different networks produced for the fMRI modality along with 10 a separate ensembles (per individual subject) consisting of 5 x 2 x 3 x 5 x 4 = 600 different networks for the dMRI modality (based on all specified combinations of hyperparameters). In sum, Stage 2 of the workflow will generate 80 + 600 = 680 unique modality-independent networks per subject that can be further cleaned and analyzed in Stages 3 and 4. Furthermore, Stage 3 will by default aggregate the 80 and 600 networks per modality into an additional ensemble of 'hierarchical' dMRI-fMRI network multigraphs, yielding 80 x 600 = 48000 possible unique combinations of all hyperparameters across modalities. All 48,680 networks (modality-independent and modality-intersecting) per subject with be run through automated network analysis, the results of which along with the networks themselves, will be saved to disk, both with independent (Stage 3) and summary (Stage 4) dataframes. Files with the suffix '_neat.csv' within the fmri subdirectory will contain the network properties for each of the 80 combinations of fMRI hyperparameters. Files with the suffix '_neat.csv' within the dMRI subdirectory will contain the network properties for each of the 600 combinations of dMRI hyperparameters. Files with the suffix '_neat.csv' within the fmri_dmri subdirectory will contain the network properties for each of the 48000 combinations of fMRI-dMRI multigraph hyperparameters. Files with the suffix '_mean.csv' within each individual's respective base subject directory contain consensus measures along with a dictionary of file paths and dataframes of each graph measure across all hyperparameters across both modalities. The latter can in turn be fed into an ensemble classifier to make highly precise predictions about each of the ten individual's likelihood of responding favorably to the experimental treatment for Major Depressive Disorder.

CLAIMS:

What is claimed is:

1. A connectome generating process, comprising:
    a. An automated method of transforming person-related data into at least one ensemble of connectomes, in the form of networks defined by nodes and edges, recognized among a plurality of possible networks that can be generated based on at least one combination of hyperparameters used in the connectome-generating process;
        i. Wherein connectome ensemble(s) are sampled at one or more hyperparameter resolutions, comprising a set of either: modality-specific networks that collectively capture *one* or *more* target network(s) or optimize some derivative graph metric(s), a hierarchy (or hierarchies) of modality-specific networks that collectively capture *one* or *multiple* target network(s) of interest or optimize some derivative graph metric(s), a hierarchy (or hierarchies) of networks across multiple data modalities that collectively capture *one* or *multiple* target networks of interest or optimize some derivative graph metric(s);
        ii. Wherein connectome-generating hyperparameters comprise of data modality-specific hyperparameters and/or modality-unspecific hyperparameters which impact the estimation of the networks produced by the connectome-generating process;
        iii. Wherein modality-unspecific hyperparameters include but are not limited to connectivity model type used to define network edges, along with network thresholding approach;
            1. With respect to the connectivity model hyperparameter, any combination of a variety of connectivity models can be used, including covariance, correlation, and/or a modality-specific definition (i.e. making the hyperparameter modality-specific);
            2. With respect to the thresholding approach, multiple sub-parameters may be specified including *types* and *schemes*, where *types* include both global and local forms of thresholding, and *schemes* include any method(s) used to achieve the target thresholding type(s);
        iv. Wherein modality-specific hyperparameters are those hyperparameters which are unique to specific data modalities for which sub-workflows (or interfaces to connectome-generating sub-workflows from third-party packages) are available;
        v. Wherein a hard-coded hyperparameter filter, preceding modality-specific sub-workflow execution, that consists of preprogrammed logic for parsing and evaluating the compatibility of user-specified hyperparameters; and
        vi. Wherein input data may be manually specified by the user (i.e. via explicit file path(s) stated as a sequence of alphanumeric strings) or semi-automatically specified (i.e. via explicit path(s), stated as a sequence of alphanumeric strings, to a base directory of dataset or datasets structured according to a supported standardized data specification protocol).

2. The connectome generating process in Claim (1) wherein multiple layers of

computational parallelism facilitate the generation of connectome ensembles using distributed multiprocessing (i.e. as opposed to serial execution), these parallel layers may include but are not limited to:

a. Parallelization across multiple individuals;
b. Parallelization of nested sub-workflows specific to each data modality; and
c. Parallelization of modular functions within each nested sub-workflow specific to each data modality.

3. The connectome generating process in Claim (1) wherein, the degree of user-requested or possible parallelism described in Claim (2) exceeds available compute resources required to ensure efficient scheduling of processes, a network flow-optimized job scheduling utility for assigning priority to parallelized connectome-generating processes may be used. Network flow optimization of for the sake of optimizing runtime execution according to any of several optimization objectives including but not limited to the following:
    a. Minimizing computational load;
    b. Minimizing compute dollar spending; and
    c. Minimizing runtime duration.
4. A four-stage process for converting person-related data into network ensemble(s) comprised of multiple steps:
    a. Initializing at least one meta-workflow configured to trigger one or more nested sub-workflows determined by hard-coded logic to be applicable according to the unique combination of input data and hyperparameters as defined in Claim (1);
        i. Whereby any of a variety of inputs (boolean, string, and numeric) are specified by the user as command-line options or through a Graphical User Interface (GUI) on a cloud or dedicated server, and with a local, remote, or containerized environment; and
        ii. Whereby the initialized individual or multi-individual workflow itself may or may not form a Directed Acyclic execution Graph (DAG) that pre-defines the phases of workflow execution and iterable expansion of all specified files and hyperparameters as defined according to Claim (1).
    b. Executing nested sub-workflows as indicated according to the logic outlined in Claim (1a,v-vi);
        i. Whereby each sub-workflow receives dependent variable inputs from the processes described in Claim (4a) (i.e. both saved in memory cache and in the DAG);
        ii. Whereby modality-specific sub-workflows are triggered if and only if the minimally required set of data and hyperparameter inputs (including defaults) are supplied by the user; and
        iii. Whereby a modality unspecific sub-workflow may be trigger in the case that a raw graph or raw graphs (i.e. pre-constructed connectomes) are provided as inputs by the user during the processes described in Claim (4a), and thus bypassing the connectome-generating process described in Claim (1);
    c. Applying graph analysis to the ensemble(s) of connectomes generated from the processes described in Claim (4a-b) such that:
        i. Graph analysis commences after all valid connectomes from the processes described in Claim (4a-b) have been generated;
        ii. Graph analysis repeats iteratively (or may be parallelized to

accommodate for this iteration in accordance with Claims (2) and (3)) for each of the connectomes generated from the processes described in Claim (4a-b);

iii. Whereby graph analysis may consist of any of a variety of conventional or custom global and/or local graph analysis algorithms, as listed in a provided configuration file that can be modified as needed by the user;

iv. Whereby those graph analysis algorithms that have been specified but which are inapplicable to the network input(s) are hard-coded to be skipped;

v. Whereby resulting graph analysis measures are organized into a 1-dimensional vector of scalar values (per unique network input into the processes described in Claim (4c)), which is subsequently written to disk as an intermediary text file to be used in the processes described in Claim (4d).

d. Aggregating graph-analytic vectors produced from the processes described in Claim (4c) for each of the networks produced from the processes described in Claim (4a-b) are organized into a dataframe (or dataframes in the case that multiple individual workflows are executed or multiple independent target networks are specified) of features for subsequent use in machine-learning or other analytic scenarios wherein the networks produced by the processes described in Claim (4a-b) are:

i. Aggregated into 1-dimensional vector(s) of edge weights for each node-pair;

ii. Used to calculate basic measures of central tendency (e.g. mean, median, mode), select sub-samples of the network ensemble(s) based on some feature-selection criterion specified by the user during the processes described in Claim (4a) including but not limited to:

1. Bayesian Markov-Chain probabilities assigned to each network;
2. Discriminability, identifiability, or other reproducibility/reliability optimization function;
3. Minimum-variance and/or maximum collinearity threshold;

5. The four stage process in claim 4 wherein multiple modality-specific (or modality-unspecific) sub-workflows, hierarchical 'multigraph' analysis will be triggered for the processes described in Claim (4c), unless otherwise specified by the user through an override indicated by the user during the processes described in Claim (4a);

a. Whereby hierarchical multigraph analysis consists of the simultaneous consideration of modality-specific and/or modality-unspecific networks or ensembles of networks as distinct network objects to be analyzed;

b. Whereby additional analytic approaches and hyperparameters (i.e. beyond those available for non-hierarchical graphs) may be implemented including but not limited to:

i. Adaptive thresholding of networks in the hierarchy or hierarchies during the processes described in Claim (4a-c) to optimize similarity of graph properties across the hierarchy or hierarchies; and

ii. The inclusion of additional derivative graph analytic calculations (e.g. versatility) that are only applicable for the analysis of hierarchical multigraphs.

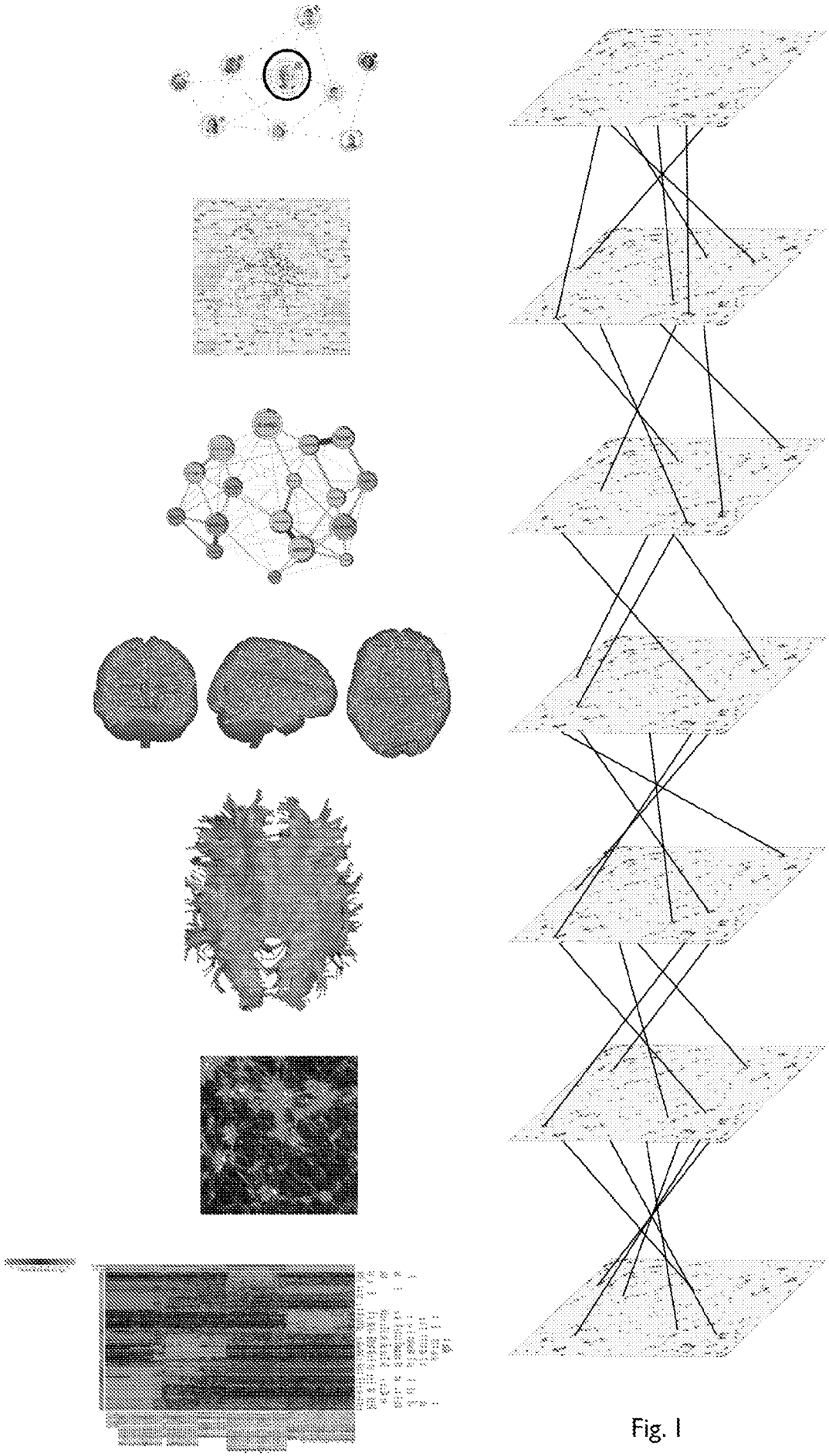


Fig. 1

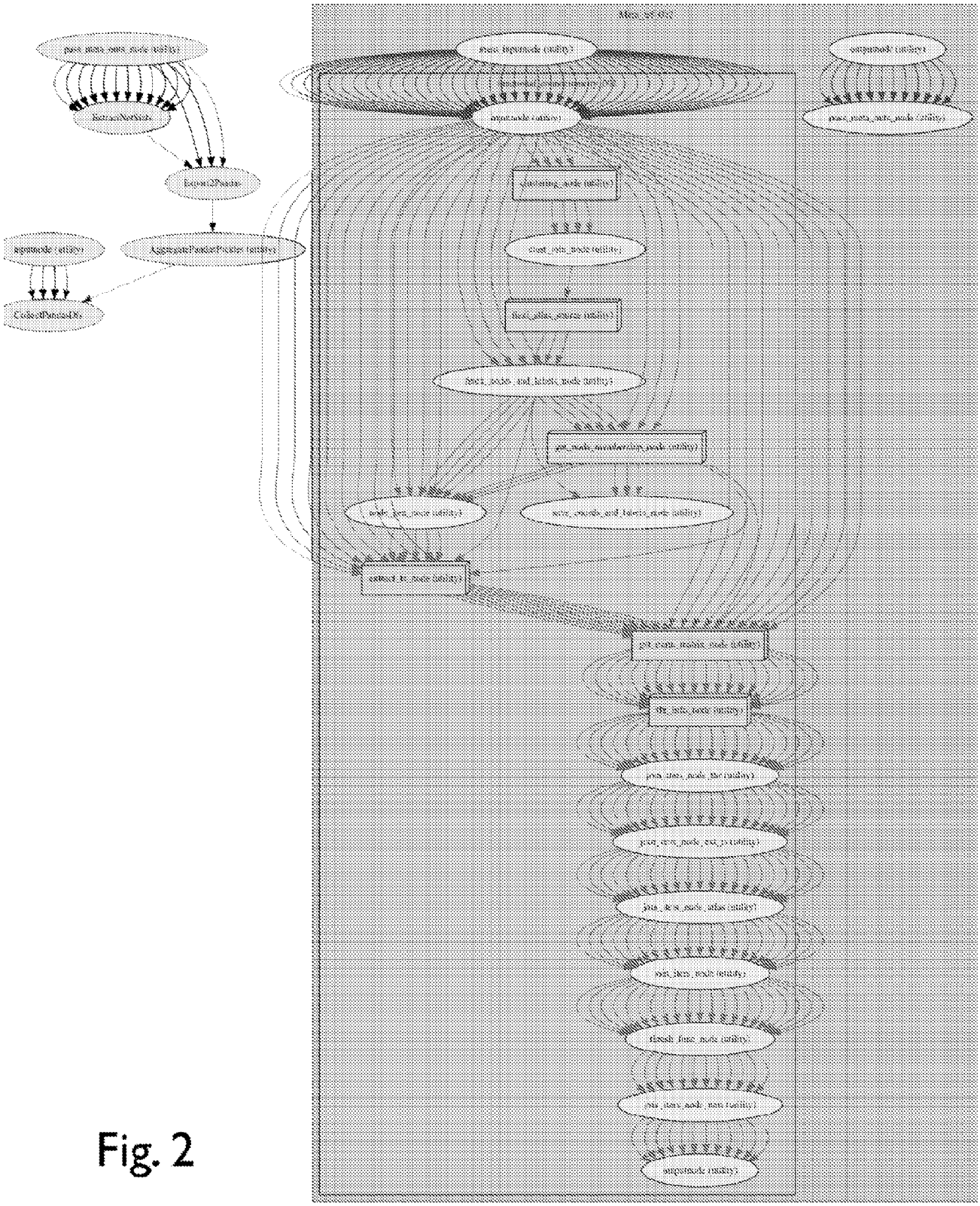


Fig. 2

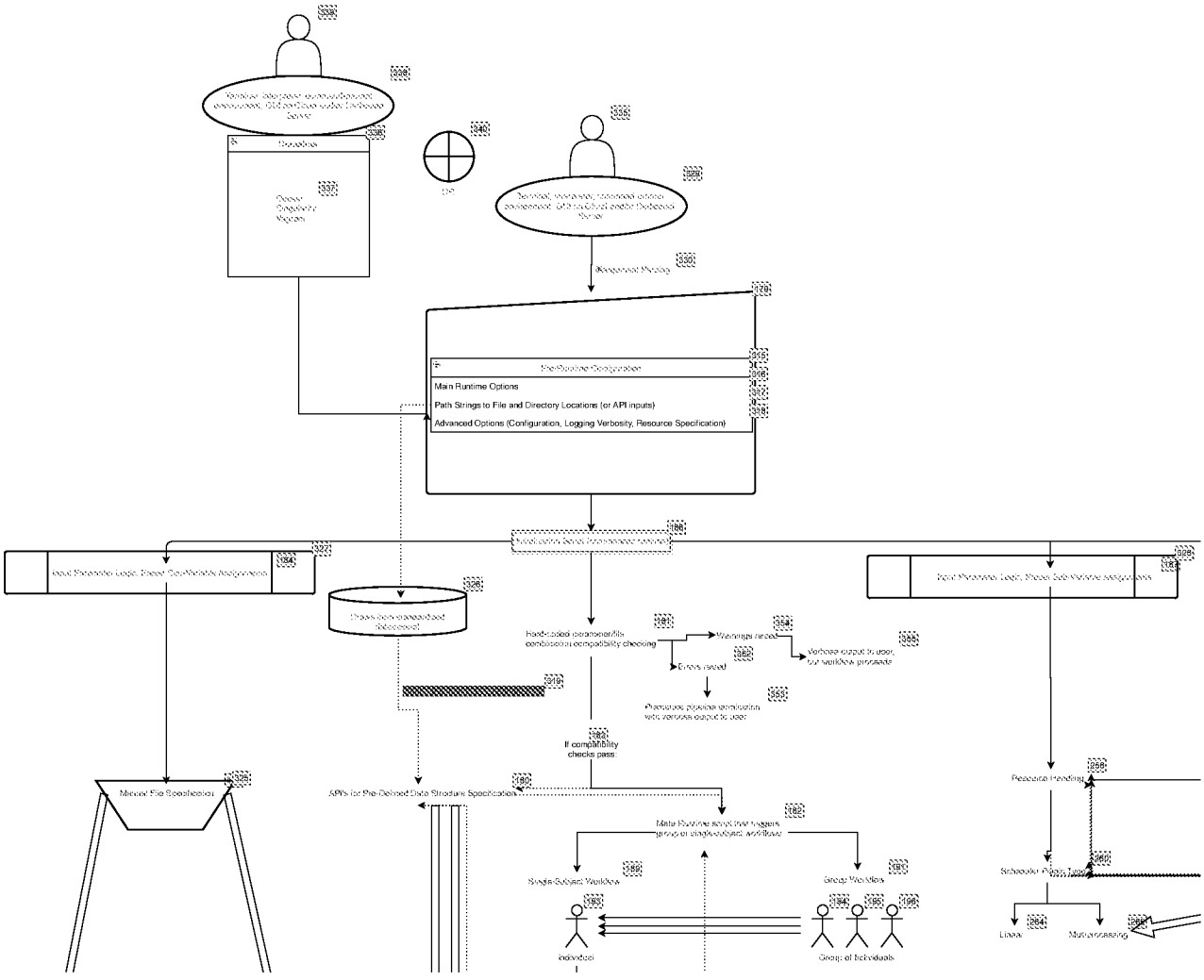

Main Runtime Options
Path Strings to File and Directory Locations (or API inputs)
Advanced Options (Configuration, Logging Verbosity, Resource Specification)
Warnings raised
Errors raised
If compatibility checks pass:
Single-Subject Workflow
Group Workflow
Individual
Group of Individuals
Linear
Multiprocessing

FIG. 3 (Sheet A)

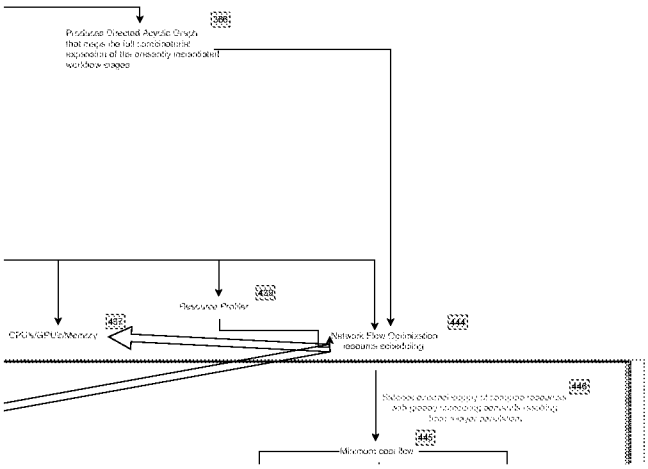

Resource Profiler
CPU/GPU/Memory
Network Flow Optimization
Minimum cost flow

FIG. 3 (Sheet B)

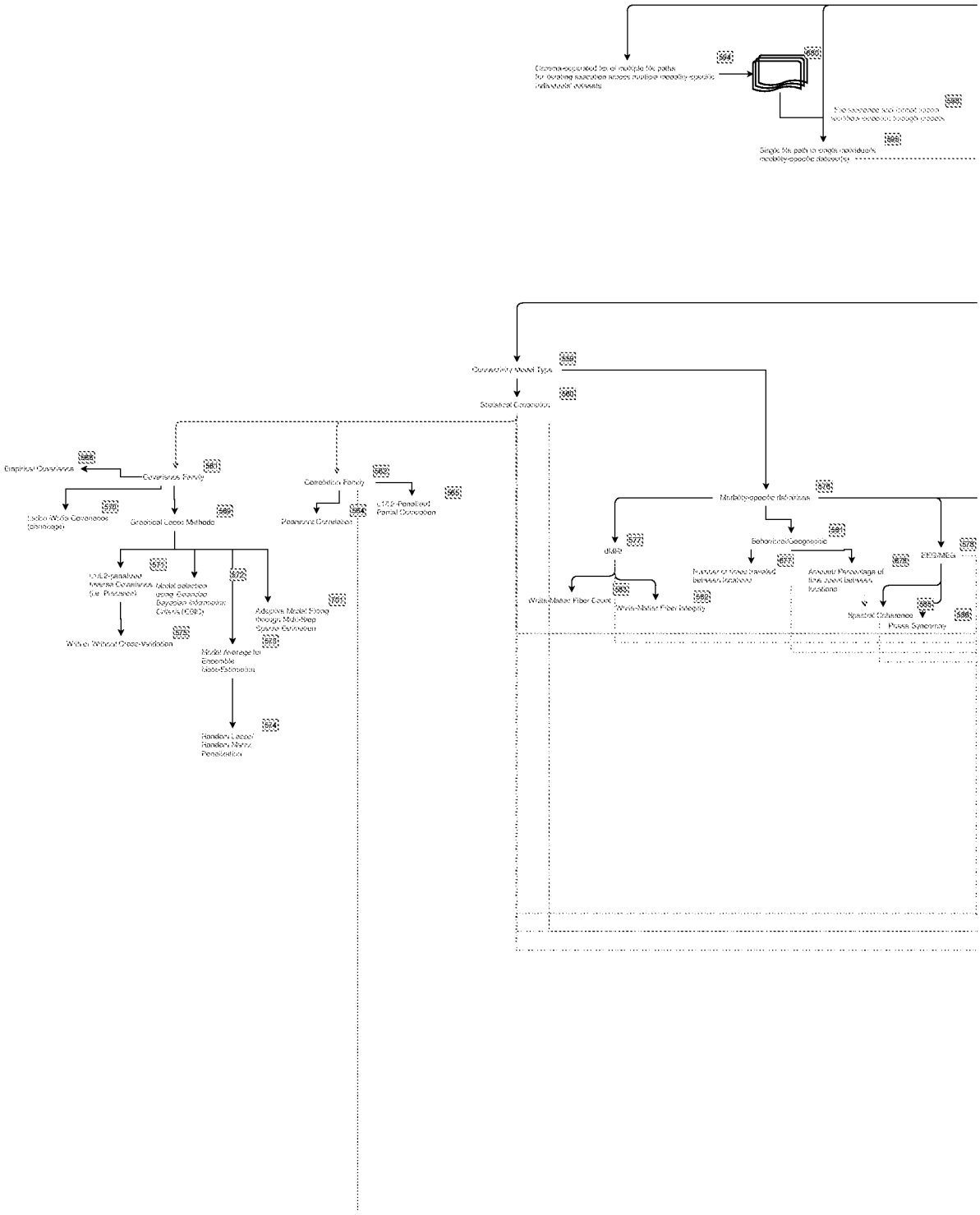

Comma-separated list of multiple file paths for iterating execution across multiple modality-specific individual datasets
Single file path to single individual's modality-specific datasetRdy
Connectivity Model Type
Statistical Covariance
Empirical Covariance
Covariance Family
Correlation Family
Modality-specific definitions
Ledoit-Wolf Covariance (Shrinkage)
Graphical Lasso Methods
Pearsonr2 Correlation
L1/L2-Penalized Partial Correlation
dMRI
Behavioral/Geospatial
EEG/MEG
Number of times traveled between locations
Amount/Percentage of time spent between locations
White-Matter Fiber Count
White-Matter Fiber Integrity
Spatial Coherence
Phase Symmetry
L1/L2-penalized Inverse Covariance (i.e. Precision)
Model selection using Extended Bayesian Information Criteria (EBIC)
Adaptive Model Fitting through Multi-Step Sparse Estimation
With or Without Cross-Validation
Model Averaged Ensemble Lasso-Estimation
Random Lasso/ Random Nano Penalization

FIG. 3 (Sheet C)

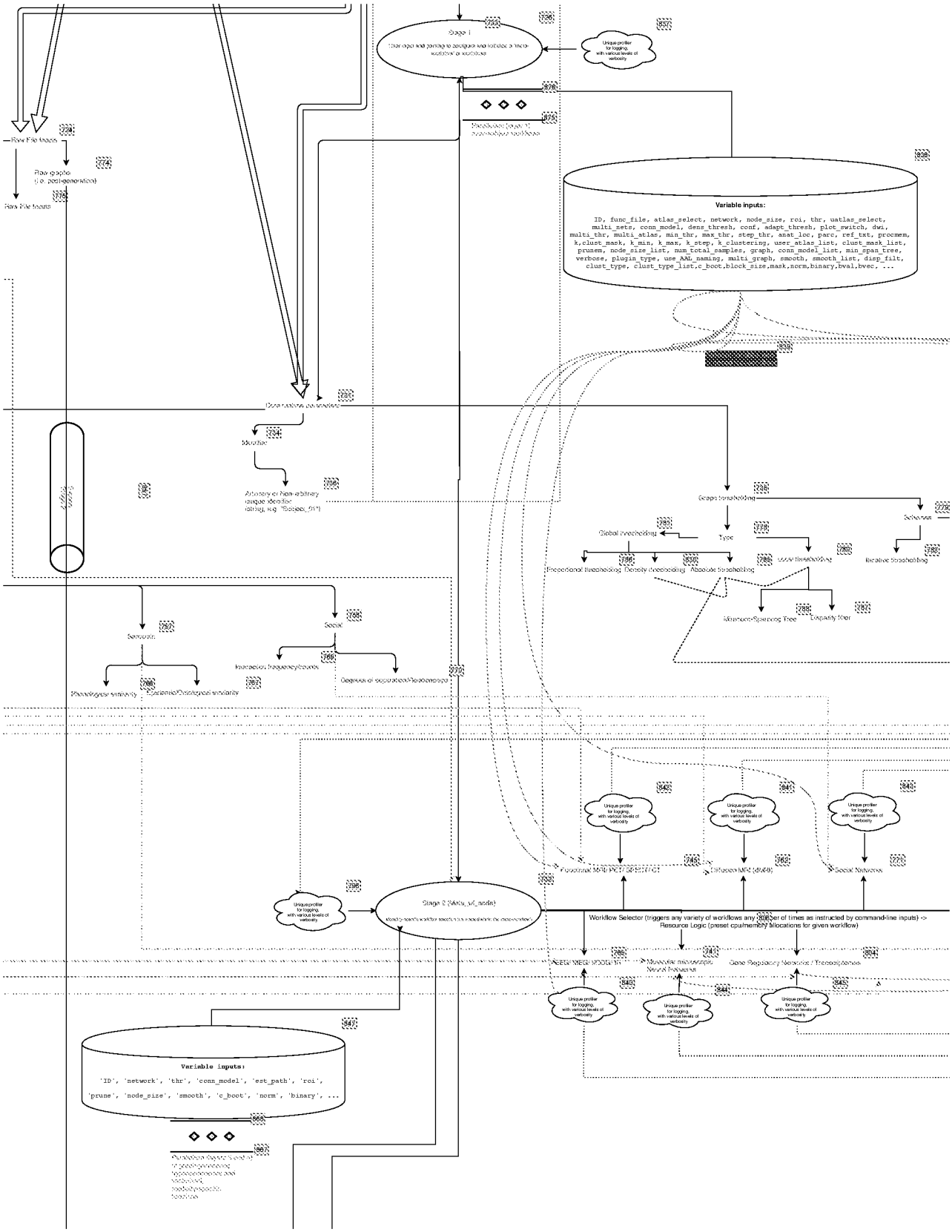

Variable inputs:
ID, func_file, atlas_select, network, node_size, roi, thr, uatlas_select, multi_nets, conn_model, dens_thresh, conf, adapt_thresh, plot_switch, dwi, multi_thr, multi_atlas, min_thr, max_thr, step_thr, anat_loc, parc, ref_txt, procmem, k,clust_mask, k_min, k_max, k_step, k_clustering, user_atlas_list, clust_mask_list, prune, node_size_list, num_total_samples, graph, conn_model_list, min_span_tree, verbose, plugin_type, use_AAL_naming, multi_graph, smooth, smooth_list, disp_filt, clust_type, clust_type_list,c_boot,block_size,mask,norm,binary,bval,bvec, ...
Raw File Inputs
Raw graphs (i.e. post-generation)
Raw File Inputs
Core utilities (all scripts)
Modality agnostic wrapper
Atlas(es) or Non-arbitrary unique identifier (string, e.g. "Default_fc")
Graph Thresholding
Global thresholding
Type
Proportional thresholding
Density thresholding
Absolute thresholding
Local thresholding
Relative thresholding
Minimum Spanning Tree
Disparity filter
Semantic
Social
Phonological similarity
Combined/Ontological similarity
Interaction frequency/counts
Degrees of separation/Relationship
Stage 1
Unique profiler for logging, with various levels of verbosity
Stage 2 (Main_wf_node)
Workflow Selector (triggers any variety of workflows any number of times as instructed by command-line inputs) -> Resource Logic (preset cpu/memory allocations for given workflow)
Functional MRI (fMRI)
Diffusion MRI (dMRI)
Structural MRI (sMRI)
Molecular microscopic Neural Networks
Gene Regulatory Networks / Transcriptomics
Variable inputs:
'ID', 'network', 'thr', 'conn_model', 'est_path', 'roi', 'prune', 'node_size', 'smooth', 'c_boot', 'norm', 'binary', ...

FIG. 3 (Sheet D)

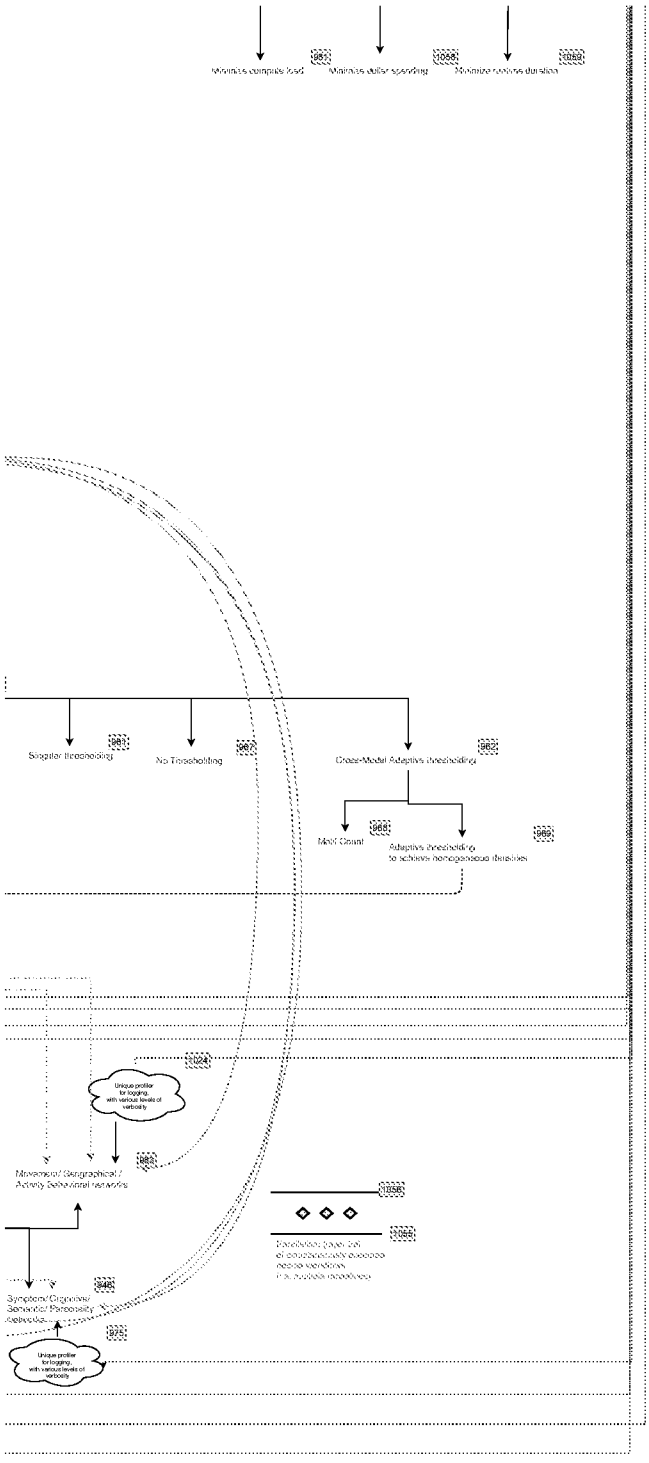

Minimize compute load
Minimize cluster spending
Minimize runtime duration
Stagger threshold
No Threshold
Craze-Model Adaptive threshold
Model Count
Adaptive threshold to achieve homogeneous densities
Unique profiler for logging with various levels of verbosity
Movement/ Geographical / Activity behavioral services
Symptom/ Diagnosis/ Semantic/ Personality services
Unique profiler for logging with various levels of verbosity

FIG. 3 (Sheet E)



FIG. 3 (Sheet F)

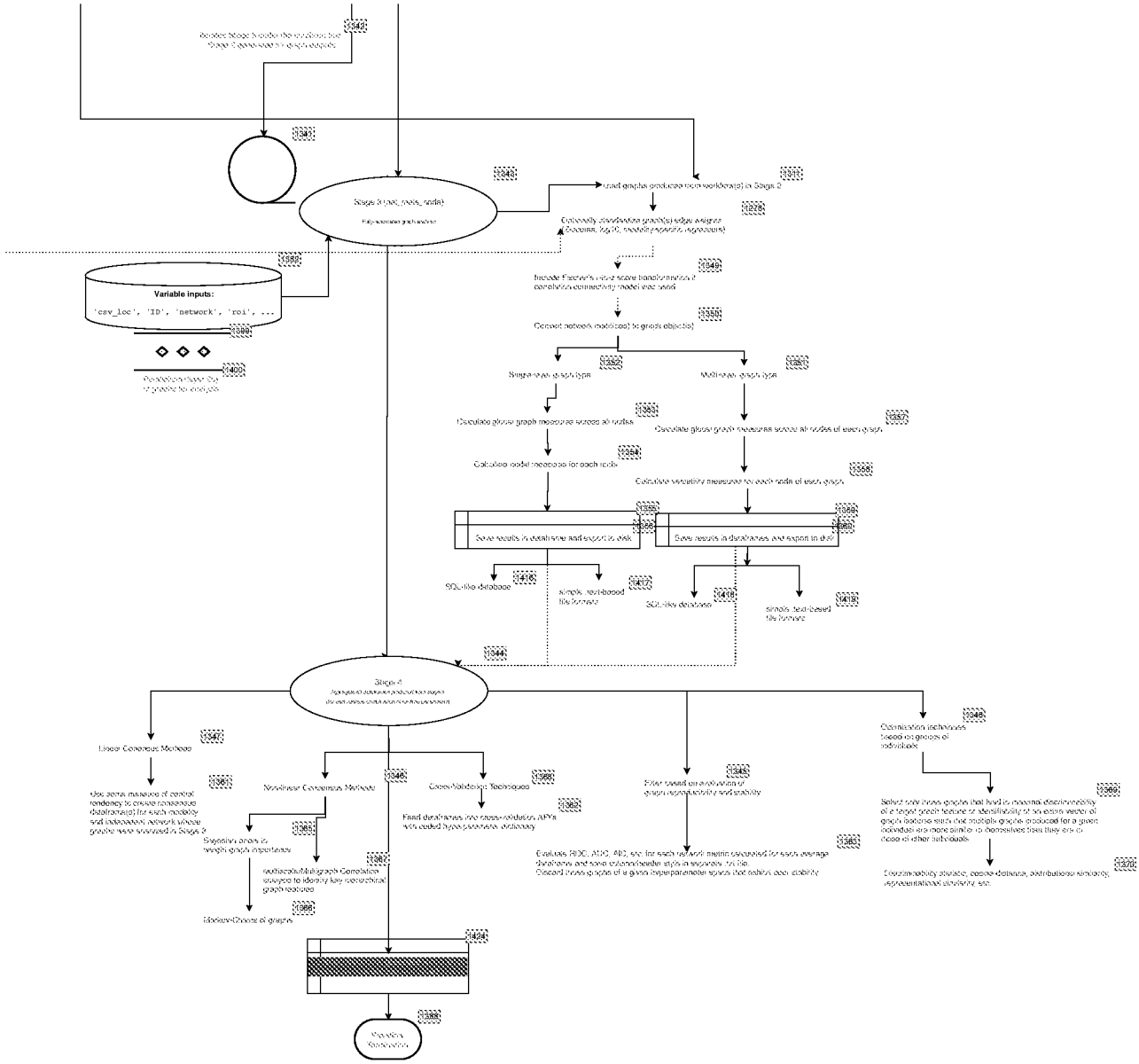

Variable inputs:
'csv_loc', 'ID', 'network', 'roi', ...
Stage 3 (loc, mets, rpnts)
Key-function graph analysis
Load graphs produced from workflow(s) in Stage 2
Include Fisher's r-to-z score transformation of correlation/connectivity model elec loads
Convert network model(s) to graph object(s)
Single-layer graph type
Multi-layer graph type
Calculate global graph measures across all nodes
Calculate global graph measures across all nodes of each graph
Calculate nodal measures for each node
Calculate versatility measures for each node of each graph
Save results in dataframe and export to disk
Save results in dataframes and export to disk
SQL-like database
simple, text-based file format
SQL-like database
simple, text-based file format
Stage 4
Linear Consensus Methods
Non-linear Consensus Methods
Cross-Validation Techniques
Filter based on evaluation of graph reproducibility and stability
Optimization techniques based on grades of individuals
Stage 5
Modularity Optimization

FIG. 3 (Sheet G)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 19/25260

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G09G 5/02, G06T 11/20 (2019.01)
CPC - G06T 11/206, A61B 5/4064, A61B 5/0042

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History Document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History Document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 9,519,981 B2 (Sudarsky et al.), 13 December 2016 (13.12.2016), entire document, especially Abstract; col 3, ln 10-30; col 5, ln 20-60; col 6, ln 25-60; col 7, ln 35 to col 8, ln 15	1-2
Y	Shi et al. "Connectome imaging for mapping human brain pathways. "In: Molecular Psychiatry volume 22, pages 1230-1240 (2017), [online] [retrieved on 25 June 2019 (25.06.2019)] Retrieved from the Internet <URL: https://www.ncbi.nlm.nih.gov/pubmed/28461700>, entire document	1-2
Y	US 2006/0094001 A1 (Torre et al.), 04 May 2006 (04.05.2006), entire document, especially Abstract; para [0079]-[0081]	2
A	Hassan et al. "EEGNET: An Open Source Tool for Analyzing and Visualizing M/EEG Connectome." In: PLoS ONE 10(9): e0138297 [2015], [online] [retrieved on 25 June 2019 (25.06.2019)] Retrieved from the Internet <URL: https://doi.org/10.1371/journal.pone.0138297>, entire document	1-2
A	US 2017/0120043 A1 (John), 04 May 2017 (04.05.2017), entire document	1-2

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 June 2019	15 JUL 2019
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 19/25260

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 3-5
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.