nature machine intelligence

Article

https://doi.org/10.1038/s42256-022-00604-2

Predicting the prevalence of complex genetic diseases from individual genotype profiles using capsule networks

Received: 22 February 2022

Accepted: 22 December 2022

Published online: 13 February 2023

Check for updates

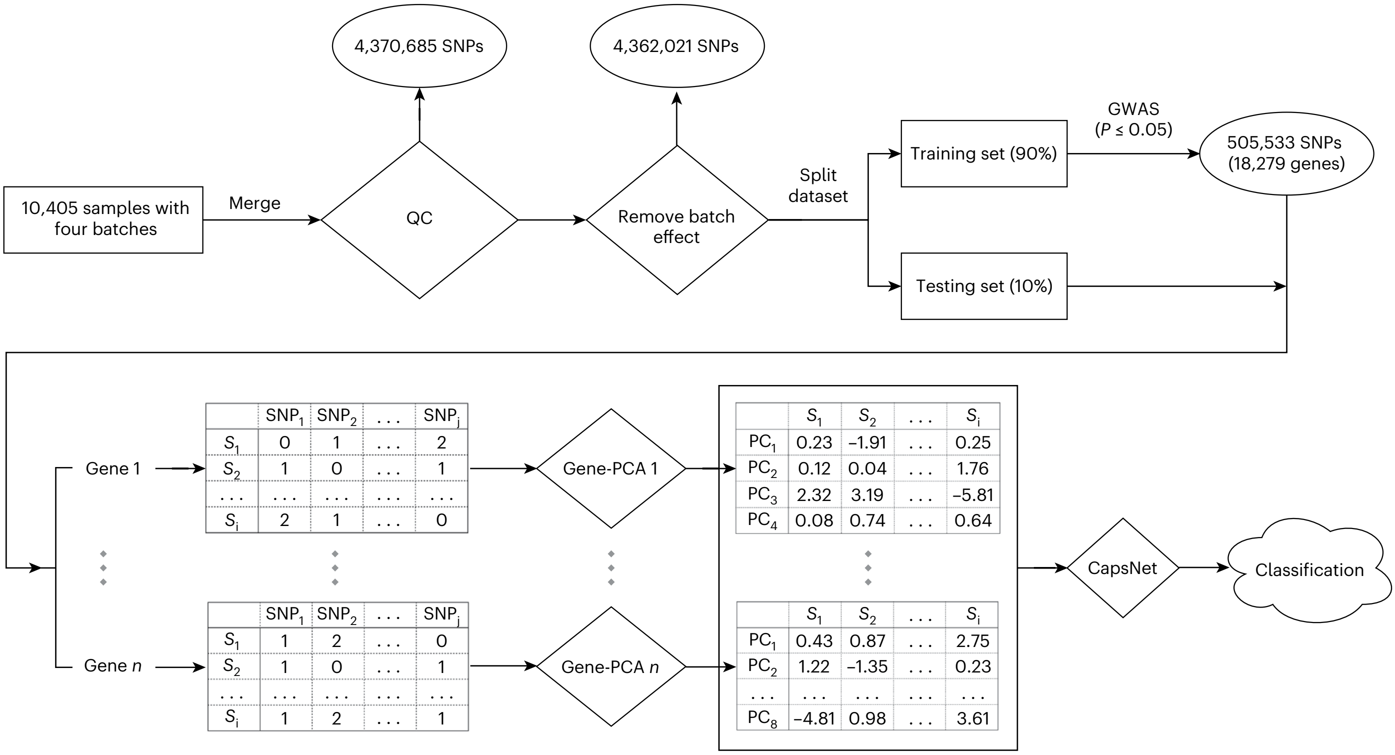
Xiao Luo[1,2,3], Xiongbin Kang[1,2,3] & Alexander Schönhuth[1,2]

Diseases that have a complex genetic architecture tend to suffer from considerable amounts of genetic variants that, although playing a role in the disease, have not yet been revealed as such. Two major causes for this phenomenon are genetic variants that do not stack up effects, but interact in complex ways; in addition, as recently suggested, the omnigenic model postulates that variants interact in a holistic manner to establish disease phenotypes. Here we present DiseaseCapsule, as a capsule-network-based approach that explicitly addresses to capture the hierarchical structure of the underlying genome data, and has the potential to fully capture the non-linear relationships between variants and disease. DiseaseCapsule is the first such approach to operate in a whole-genome manner when predicting disease occurrence from individual genotype profiles. In experiments, we evaluated DiseaseCapsule on amyotrophic lateral sclerosis (ALS) and Parkinson's disease, with a particular emphasis on ALS, which is known to have a complex genetic architecture and is affected by 40% missing heritability. On ALS, DiseaseCapsule achieves 86.9% accuracy on hold-out test data in predicting disease occurrence, thereby outperforming all other approaches by large margins. Also, DiseaseCapsule required sufficiently less training data for reaching optimal performance. Last but not least, the systematic exploitation of the network architecture yielded 922 genes of particular interest, and 644 'non-additive' genes that are crucial factors in DiseaseCapsule, but remain masked within linear schemes.

Amyotrophic lateral sclerosis (ALS) is a rare primary neurodegenerative syndrome characterized by human motor system degeneration. So far, ALS is still not curable, but symptomatic treatment can significantly improve life quality and survival of the affected[1]. As the diagnosis of ALS often comes at a considerable delay[2], most patients miss the advantageous opportunities of early intervention[3]; for example, recent studies show that NAD$^+$ replenishment can improve clinical features of patients with ALS, indicating an encouraging potential novel treatment for ALS[4,5]. These explain why efficient methods and tools for predicting the prevalence and occurrence of ALS have life-saving potential.

Various studies[6–8] have demonstrated that ALS is a complex disorder that has an encompassing genetic background[9] and its heritability amounts to 50% (ref. [10]). However, the genetic variants delivered by genome-wide association studies (GWAS) have been amounting to only 10% of the heritability[11] of ALS. It is therefore reasonable to assume that the missing heritability of ALS amounts to approximately 40%.

[1]Life Science & Health, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands. [2]Genome Data Science, Faculty of Technology, Bielefeld University, Bielefeld, Germany. [3]These authors contributed equally: Xiao Luo and Xiongbin Kang. e-mail: aschoen@cebitec.uni-bielefeld.de



	SNP_1	SNP_2	...	SNP_j
S_1	0	1	...	2
S_2	1	0	...	1
...	...	...	...	...
S_i	2	1	...	0

	SNP_1	SNP_2	...	SNP_j
S_1	1	2	...	0
S_2	1	0	...	1
...	...	...	...	...
S_i	1	2	...	1

	S_1	S_2	...	S_i
PC_1	0.23	−1.91	...	0.25
PC_2	0.12	0.04	...	1.76
PC_3	2.32	3.19	...	−5.81
PC_4	0.08	0.74	...	0.64

	S_1	S_2	...	S_i
PC_1	0.43	0.87	...	2.75
PC_2	1.22	−1.35	...	0.23
...	...	...	...	...
PC_8	−4.81	0.98	...	3.61

Fig. 1 | An overview of the workflow. In the tables for each gene, S_1, S_2, ..., S_i represent sample IDs, and PC_1, PC_2, ..., PC_k represent the 1, 2, ..., *k*-th principal component of each Gene-PCA, respectively. The number of PC is 8 or 4 or 1, which depends on the length of the input, that is, the number of SNPs.

The striking amount of missing heritability, or even the idea that genotype–phenotype relationships are based on the omnigenic[12,13] and not the polygenic model—where only the polygenic model renders the concept of missing heritability a truly reasonable concept—is supposed to be among the major reasons for the poor prognosis of ALS. In summary, the major methodological challenges are as follows: (1) the association signals of complex diseases can spread across most of the genome instead of involving just a few core pathways[12,14], which means that the omnigenic model applies; (2) when following the polygenic model, the corresponding linear models that underlie the GWAS analysis techniques cannot detect non-additive genetic effects such as epistasis[15,16]. Several studies have modelled gene–gene interactions[17–19]. However, only a few of them have been directly applied to predicting the prevalence of complex genetic diseases.

Methods that aim to exploit non-additive relationships have left behind various open questions. When being based on statistical hypothesis testing, they tend to suffer from a lack of power. On the other hand, approaches that are based on potentially omnigenic machine learning models are relatively rare, and so far have left ample room for improvements.

Deep learning, as a predominant machine learning approach, has established the state of the art in many areas. Extensions of the universal approximation theorem[20] provide a theoretical basis for the insight that not only the width of the layers but also the depth of the network is crucial for reaching superior levels of prediction accuracy[21,22]. The intuitive idea is to detect and arrange patterns in a hierarchical way, which leads to elevated levels of resolution when mapping the data[23].

Convolutional neural networks (CNNs) reflect network architectures that are particularly suitable to implement the idea of hierarchies of patterns [24]. While CNNs such as AlexNet[22], VGG[25], ResNet[26] and DenseNet[27] indeed achieved the breakthrough successes in deep learning, major criticisms with regard to interpretability ('deep black boxes')[28] and the enormous demand for training data for reaching superior performance[29] had been remaining. However, in clinical applications, data can be expensive, and the inability to explain lets one remain with ethical concerns[30,31].

Capsule networks (CapsNets)[32,33] were presented as a remedy for addressing such critical issues. The major motivation for CapsNets was to classify distorted or entangled patterns in images correctly. The improved modelling of spatial hierarchies, as empowered by the 'viewpoint invariance property', led to major improvements with respect to the accuracy of the predictions. Beyond that, the 'viewpoint invariance property' is the likely reason for the reduced requirements in terms of training data that one observed in comparison with CNNs. Further, although not primarily intended, CapsNets also enabled a human-mind-friendly interpretation of results.

Therefore, CapsNets have shown to have the potential to resolve two issues of primary concern in biomedical applications. Recent studies indicate that the potential of CapsNets to learn complex hierarchical structures can indeed be leveraged also for biomedical data[34,35]. These applications of CapsNets add to the earlier (innumerous) applications[36–40] of ordinary CNNs or machine learning models in biology or medicine.

There also have been recent applications of deep learning when predicting phenotype from genotype profiles, such as the detection of epistasis[41,42], or the prediction of ALS from genotype profiles of the promoter regions from four chromosomes[43]. Last but not least, an approach was presented that models non-linearities within the range of LD (linkage disequilibrium) blocks, while joining effects of LD blocks in a linear manner[44], which prevents recognition of non-linear interactions across LD blocks.

The omnigenic model[12] requires the modelling of gene–gene interactions across the entire genome, in a way that allows genes to interact in non-additive ways for establishing their joint effects. From this point of view, none of the approaches presented so far in the literature builds on the omnigenic model as its foundation.

Table 1 | Classification results for ALS test data. The values are represented as percentages. SVM, support vector machine. *a*, *b*, *c* and *d* represent PRS-based models that the SNPs were selected by GWAS with the threshold $P<5\times10^{-2}$, $P<5\times10^{-4}$, $P<5\times10^{-6}$ and $P<5\times10^{-8}$, respectively. Note that the best score is marked in bold

Dimension reduction	Classifier	Accuracy	Precision	Recall	F1 score
Gene-PCA	DiseaseCapsule	**86.9**	85.2	89.4	**87.2**
Gene-PCA	MLP	84.2	92.2	74.8	82.6
Gene-PCA	Logistic regression	78.2	71.1	94.8	81.3
Gene-PCA	SVM	76.3	**94.8**	55.8	70.3
Gene-PCA	CNN	74.5	86.1	58.5	69.7
Gene-PCA	Random forest	63.3	73.0	42.1	53.4
Gene-PCA	AdaBoost	62.7	86.3	30.2	44.7
All-PCA	DiseaseCapsule	81.9	80.7	83.8	82.2
All-PCA	Logistic regression	78.1	70.7	**96.0**	81.4
All-PCA	SVM	76.3	**94.8**	55.8	70.3
All-PCA	MLP	72.5	85.2	54.4	66.4
All-PCA	AdaBoost	67.6	84.8	42.9	57.0
All-PCA	Random forest	64.1	73.3	44.4	55.3
All-PCA	CNN	53.8	54.8	42.5	47.9
–	PRS^{a}	81.8	91.5	70.2	79.4
–	PRS^{b}	78.5	84.4	69.8	76.4
–	PRS^{c}	74.2	76.8	69.4	72.9
–	PRS^{d}	63.5	63.5	63.3	63.4

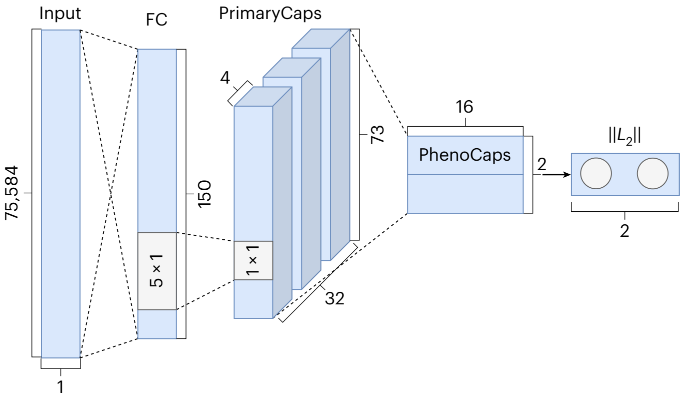


Fig. 2 | The architecture of DiseaseCapsule. The input is the concatenation of the compressed features from all Gene-PCA models, where each feature corresponds to one Gene-PCA. The number of Gene-PCAs is 75,584, so the dimensionality of the input is 75,584 × 1. DiseaseCapsule consists of three layers: a fully connected layer (FC), a primary capsule layer (PrimaryCaps) and a phenotype capsule layer (PhenoCaps). The FC layer consists of 150 neurons followed by ReLU as activation function. The PrimaryCaps is composed of 32 primary capsules. Each of them involves four different convolutional filters (kernel size 5 × 1, stride 2, no padding). PhenoCaps consists of two 16-dimensional vectors. Each phenotype capsule receives input from all 32 primary capsules. The output is a binary classification label (Healthy or ALS).

Our approach will be the first approach to model whole-genome wide, non-additive interactions between genes. For that, it takes a route that one can consider the opposite of the protocol presented in ref. [44]: we summarize local (gene-range) effects of variants linearly, and then combine the local effects in non-additive ways globally (across the entire genome). In this Article, we present a deep neural network-based approach that caters to the omnigenic model as a conceptual basis. More specifically, from a methodical point of view, we present the first approach that employs capsule networks, as an advanced deep neural network class of functions, to map genotypes onto phenotypes.

We refer the interested reader to Supplementary Note 1 for full details on arguments referring to deep learning and capsule networks raised in the introduction.

Results

We present DiseaseCapsule as a framework that can handle genome-scale variant input and reveal non-additive interactions across genes. Please see Supplementary Note 2 for an overview of the approach and a summarizing account of the results; here, for the sake of reproducibility, we will report results in sufficient detail. For all methodical details, see Methods.

Workflow

DiseaseCapsule implements a two-step approach to successfully deal with genome-scale variant screens. The first step consists of a novel protocol to perform dimensionality reduction. This protocol enables to capture millions of polymorphic loci in a way that enables subsequent application of non-additive methods. The second step then is the application of capsule networks, as a fundamentally non-additive model, to the reduced data. These two steps are preceded by basic quality control (QC) and batch effect removal, as well as a basic step to filter out variants that do not matter (regardless of the underlying approach). For an illustration, see Fig. 1. In the following, we describe the essence of the two steps, and refer the reader to Methods for full methodical details. In the following, the data on which DiseaseCapsule and competing methods are validated refers to 10,405 DNA-array-based, whole-genome genotype samples from the Dutch cohort of Project MinE[8,11,45].

For the first step, the challenge is the fact that the application of principal component analysis (PCA) across the polymorphic loci of the entire genome—which is the standard protocol to reduce the dimensionality of the data—annihilates the effects of subsequent application of genome-wide, non-linear models in so far as non-linear interactions between global, linear combinations of variants remain meaning less for the analyst. Our solution is to apply linear techniques, such as PCA, only for small, biologically well-defined functional units of the genome (that is, genes). As linearization happens only within small regions of the genome, non-linear interactions across such small regions can still be detected. Beyond this theoretical reasoning, our experiments confirm these ideas by revealing the idea of only local PCA as the considerably stronger approach (Table 1 and Supplementary Tables 1 and 2). For more details about descriptions of the challenge and the solution, see Supplementary Note 3.

For the second step, see Fig. 2 for an illustration and a brief description of the architecture of DiseaseCapsule; for full details, see Methods.

Predicting ALS: Gene-PCA + DiseaseCapsule yields optimal performance

For the following, see Table 1. We evaluated the performance of DiseaseCapsule with various state-of-the-art approaches, including predominant machine learning approaches and polygenic risk scores (PRS). To further evaluate the contribution of the novel gene-scale dimensionality reduction protocol, we combined each method with standard genome-scale PCA ('All-PCA') on the one hand, and the novel, local PCA-based protocol ('Gene-PCA') on the other hand. We discuss results briefly in the following; for full details, see Supplementary Note 4.

The first observation is that all neural network-based (so fully non-additive) models profit from using 'Gene-PCA' instead of 'All-PCA':

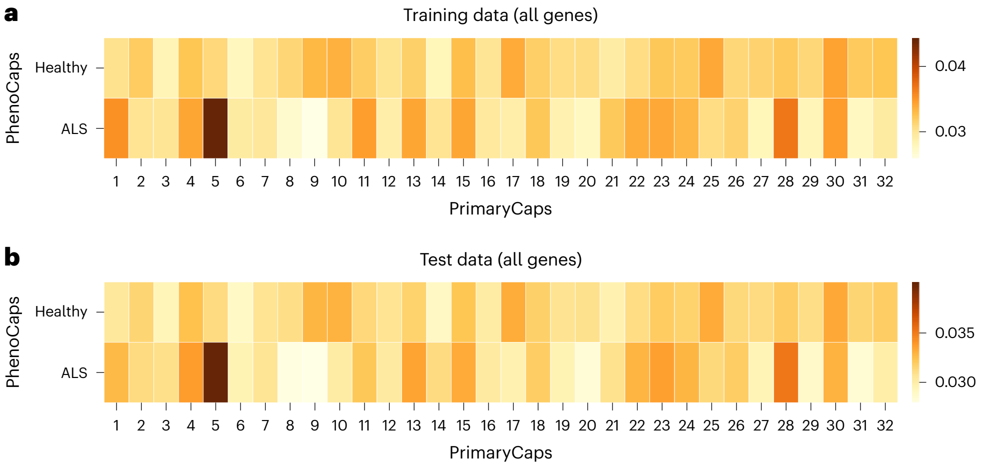


Fig. 3 | Heat map plots for average coupling coefficient matrices of DiseaseCapsule. a, Only plot for training individuals. **b**, Only plot for test individuals. The *x* axis and *y* axis represent the 32 primary capsule groups and 2 phenotype capsules, respectively. All 18,279 genes involved in DiseaseCapsule model are retained when predicting on test samples.

the accuracy of DiseaseCapsule, multilayer perceptron (MLP) and CNN increases from 81.9 to 86.9, from 72.5 to 84.2 and from 53.8 to 74.5, respectively. All other approaches are run on 'Gene-PCA' or 'All-PCA' at near-identical performance. This points out that Gene-PCA predominantly caters to neural network models. From a conceptual point of view, this was to be anticipated, because Gene-PCA preserves the potential to detect non-linearities across genes, whereas ordinary PCA does not. Since applying local linearization before a linear model-based analysis still results in an approach that is linear overall, as the concatenation of two linear functions, linear methods cannot profit from Gene-PCA; still they fail to pick up non-linearities.

In an overall account, DiseaseCapsule achieves a prediction accuracy of 86.9, which establishes the top performance, rivalled only by MLP, as an alternative non-additive approach (84.2). The third-best performance is achieved by DiseaseCapsule on 'All-PCA' (81.9), closely followed by PRS at a GWAS threshold of 5×10^{-2} (81.8). All other performance rates drop below 79. This means in particular that, in a relative comparison with PRS, as a standard prediction technique, DiseaseCapsule leaves 28% fewer individuals misclassified, which establishes considerable, relevant progress, both from the point of view of clinical applications and from the point of view of predictive power in general. It also establishes a first quantification of the contribution of non-additive constellations of variants/genes to identifying ALS (for further analyses, see Supplementary Note 4).

DiseaseCapsule also clearly reveals itself as the most balanced and strongest approach overall: DiseaseCapsule achieves precision and recall of 85.2 and 89.4, respectively, which combines into an F1 score of 87.2, which is unrivalled by the other approaches.

In addition, results show that, compared with other classifiers, DiseaseCapsule is less sensitive to batch-induced or cohort-specific confounding effects (Supplementary Table 3) and needs less data for training (Supplementary Table 4). For a fully detailed discussion of the corresponding experiments, see Supplementary Notes 5 and 6.

Validating DiseaseCapsule on PD

We also validated the predictive performance of DiseaseCapsule in Parkinson's disease (PD) data[46–49], following the exact same protocol as for ALS. In a summary of results, in analogy to ALS, Gene-PCA + DiseaseCapsule outperforms all other approaches in terms of accuracy (62.0%), recall (68.1%) and F1 score (64.2%) in PD (Supplementary Table 5). For more details, see Supplementary Note 7. The loss of accuracy in comparison with ALS, observed for all methods, can be attributed to the relatively small amounts of polymorphic loci inspected for the PD cohorts, which has a clear impact on the expressiveness of all models.

Increasing number of genes improves classification

For the classification performance of DiseaseCapsule when varying the number of genes, see Supplementary Fig. 1, and for full details, see Supplementary Note 8. It is immediately evident that increasing the number of genes improves results in all aspects. This arguably supports the hypothesis that the omnigenic model[12] is in effect. It remains to design a strategy through which to select the genes that are most relevant for classification; most likely, such genes have key roles in establishing or preventing the disease.

Determining genes decisive for classification

For explanations in the following, see Extended Data Fig. 1 and see 'The architecture and parameters of DiseaseCapsule' and 'Model interpretation' subsections in Methods. While *i* indexes primary capsules, *j* indexes higher-level ('phenotype') capsules. In the DiseaseCapsule network, the vectors of the two phenotype capsules 'ALS' and 'Healthy' ($\mathbf{s}_\mathbf{j}$ in Extended Data Fig. 1) consist of linear combinations of output vectors provided by the primary capsules ($\mathbf{u}_{\mathbf{j}|\mathbf{i}}$ in Extended Data Fig. 1). The linear weights c_{ij} that connect the $\mathbf{s}_\mathbf{j}$ with the $\mathbf{u}_{\mathbf{j}|\mathbf{i}}$ are referred to as coupling coefficients. Unlike ordinary parameters of the network (for example, $\mathbf{W}_\mathbf{ij}$ in Extended Data Fig. 1), the c_{ij} are not learned (using backpropagation), but determined through the dynamic routing procedure, as a novel and characteristic component of capsule networks. The dynamic routing algorithm induces situations that favour only few—often even only one—large c_{ij} over several equally large c_{ij} for each *i*. This means that each primary capsule predominantly 'routes' its output to only few or even only one higher-level capsule.

Primary capsules that have great coupling coefficients in connection with the 'ALS' capsule are likely to code for constellations of genes that drive the disease. Vice versa, primary capsules sharing links with the 'Healthy' output capsule that are equipped with large coupling coefficients code for genes whose activation (or de-activation) distinguishes healthy individuals from the ones affected with ALS.

Primary capsules that predominantly route their output to one of the phenotype capsules, 'ALS' or 'Healthy', are crucial factors for the classification process. We therefore investigated how primary capsules related to the 'ALS' output capsule (which predominantly fires when ALS is to be predicted) on the one hand, and the 'Healthy' output capsule (which predominantly fires when the individual is to be classified as not being affected by ALS) on the other hand.

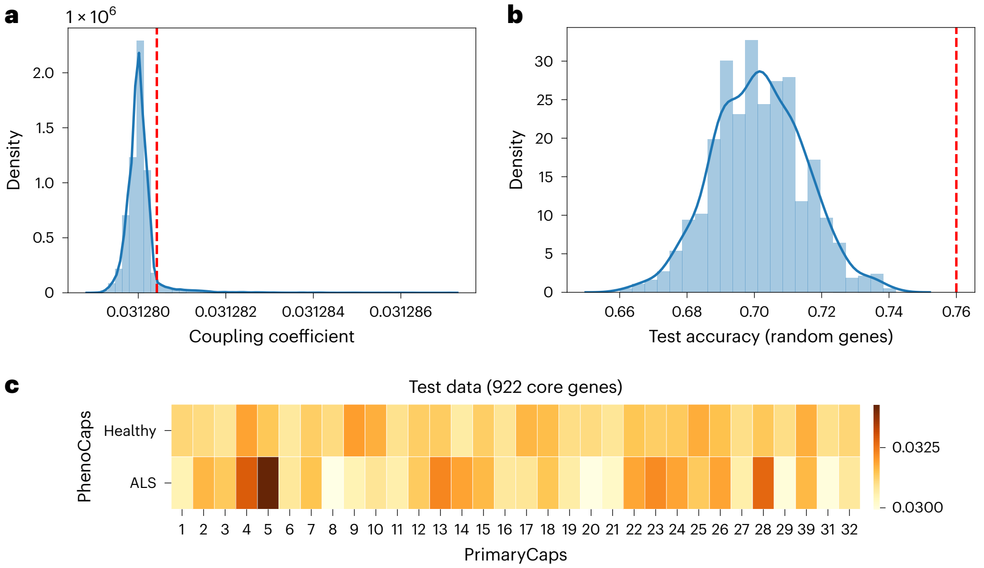


Fig. 4 | Determining and validating core genes decisive for classification. **a**, The distribution of coupling coefficients between primary capsule 5 and phenotype capsule ALS for all genes. The red dashed line indicates the 95th percentile. A total of 922 genes whose coupling coefficients are above the 95th percentile are selected as core genes decisive for classification. The vertical coordinates adopt scientific notation (×10^6). **b**, Test accuracy distribution of using 922 randomly chosen genes as input for DiseaseCapsule model (repeat 1,000 times), while the other genes are masked (set as zero). The red dashed line indicates the test accuracy of using 922 core genes as input. **c**, Heat map for average coupling coefficient matrices (test data) using 922 core genes as input.

To highlight primary capsules that share exceptionally large coupling coefficients with one of the two phenotype capsules, we ran all training and all test samples through the network, amounting to two separate runs, one for the training and one for the test data. The intuition is to demonstrate that, despite not having been part of the training, effects reproduce on data that had not been seen before. We collected the resulting coupling coefficients; we recall that coupling coefficients are computed individually for each sample by way of the 'dynamic routing' protocol during the forward pass[32]. If coupling coefficients were determined as part of backpropagation during training, coupling coefficients would be equal for all individuals. We averaged the resulting coupling coefficients across all samples, for each of the 2 × 32 possible combinations of PrimaryCaps and PhenoCaps. As above-mentioned, we did this for both training and test data. The corresponding averaged coupling coefficients are displayed in the two heat maps in Fig. 3, where Fig. 3a is for the training data and Fig. 3b is for the test data. For further details on the experiments and the corresponding visualization process, see also Methods.

The most striking effect is that primary capsule 5 establishes the strongest link to the 'ALS' output capsule, both for training and test data. Far lesser so, but still apparent, primary capsule 28 activates the 'ALS' capsule. The agreement between training and test data demonstrates that effects do not only get manifested on data that were used to establish the parameters of the network (namely, the training data). In summary, activation of primary capsule 5 is the by far predominant effect from which to determine whether an individual is affected with ALS.

Correspondingly, we developed an algorithmic protocol according to which to determine core genes that markedly contribute to the activation of primary capsule 5; for details, see Methods. Using this protocol, we obtained 922 core genes that contribute to the activation of primary capsule 5 (Fig. 4a). This means that these 922 genes are important for DiseaseCapsule to identify the occurrence of ALS. To validate the predictive power of these 922 genes, we masked all other genes in the test data (that is, we set entries in the input vector to zero if referring to genes not among the 922 selected ones), and ran the modified test data through the (trained) DiseaseCapsule model. As a result, using these 922 genes alone—which as we recall are crucial for activating primary capsule 5—a test accuracy of 76% was achieved. Note that random selections of 922 genes yielded 70% accuracy on average, with a standard deviation of 1%; for corresponding results, see Fig. 4b. So, the 76% achieved by the genes selected through our protocol is significantly greater ($P < 2.2 \times 10^{-16}$).

To further corroborate the predictive power of the 922 selected genes, we computed the average of the coupling coefficients across all test individuals when using only the 922 preferentially selected core genes, see Fig. 4c. The level of activation of primary capsule 5 (0.0342) nearly matches the one when using all genes (0.0399).

It therefore made sense to examine and classify these 922 genes further; for the resulting list, see Supplementary Table 6. In summary, when examining these 922 genes, we found some overlapping genes with the Amyotrophic Lateral Sclerosis online Database (ALSoD) and other studies [44]. Additionally, the collection of enriched Gene Ontology terms and pathways have been shown to significantly relate with human nervous system related diseases, which do include ALS in most cases[50–55]. For full details, see Supplementary Note 9.

644 'non-additive' genes

We designed a simple target function that, for a selection of genes, measures the difference between the genes supporting DiseaseCapsule and the genes supporting a common logistic regression scheme, in terms of accurate classification. This difference is quantified by the difference in accuracy that one achieves when running DiseaseCapsule using these genes alone, on the one hand, and when running the logistic regression scheme using these genes alone, on the other hand. Employing a genetic algorithm, we determined a subset of 644 genes (Supplementary Table 7) that yields a maximum of that target function. So, running methods on these 644 genes maximizes the difference between the accuracy achieved in the non-additive scheme (DiseaseCapsule) and the accuracy achieved in the additive scheme (regression). For full details, see Methods. In other words, these 644 genes remain useless when being used in a linear scheme, but decisively contribute to classification in DiseaseCapsule, as a non-additive scheme; more than that, these 644 genes reflect a selection that is optimal in that respect.

Table 2 | Test accuracy, precision and recall using non-additive genes for prediction. Models are retrained using only 644 non-additive genes

Model	Accuracy	Precision	Recall
Gene-PCA + DiseaseCapsule	0.712	0.715	0.706
Gene-PCA + logistic regression	0.512	0.522	0.292
Difference	0.200	0.193	0.414

To reinforce that these 644 'non-additive' genes are responsible for predominantly non-additive effects, we ran both DiseaseCapsule (Gene-PCA + DiseaseCapsule in Table 1) and logistic regression (Gene-PCA + logistic regression, which proved the strongest additive protocol; Table 1), on both training (which led to establishing parameters for both DiseaseCapsule and logistic regression) and test data (hitherto unseen by either approach). The differences in accuracy are striking, on both training (difference 0.227) and test (difference 0.162) data.

The difference between training and test data may be due to hidden biases. To prevent such biases, and ensure that these 644 genes nearly exclusively yield non-additive effects—which can only be picked up by DiseaseCapsule—we retrained both DiseaseCapsule and logistic regression on these 644 genes alone. This gives both approaches the same fair chance: try to get the most out of the 644 genes they were provided with.

For the corresponding results, see Table 2. Retraining both models yields accuracies of 0.712 for DiseaseCapsule, but only 0.512 for logistic regression (amounting to a difference of exactly 0.2, matching the range of the earlier results). Note that because test data were balanced (equal cases and controls), 0.5 means random performance. So, on these 644 genes, logistic regression matches random classifier rates, while DiseaseCapsule achieves excellent performance rates.

Breaking down results in terms of precision and recall confirms that DiseaseCapsule establishes decent performance rates (precision 0.715; recall 0.706), whereas logistic regression's performance does not even match minimum standards (precision 0.522; recall 0.292(!)). For functional annotations of the 644 non-additive genes, see Supplementary Note 10.

Discussion

In this study, we have presented DiseaseCapsule, as a novel deep-learning-based approach to infer disease phenotype from individual genotype. As the major novelty, DiseaseCapsule captures non-linear, potentially arbitrarily complex functional relationships between genotype and phenotype across the whole genome. All prior approaches presented so far considered to evaluate variants in additive schemes (which reflects the common standard), presented approaches that consider non-additivity only within small local regions of the genome, or resorted to considering only a few genes, or some selected chromosomes when operating in more global ways.

DiseaseCapsule has come with two immediate theoretical advantages. First, because it operates across the whole genome, DiseaseCapsule does not need to focus exclusively on a few core disease-related genes, so it does not miss the weak effects of abundant peripheral genes. Second, DiseaseCapsule improves on capturing the hierarchical structures of the underlying genetic interactions thanks to the high degree of complexity that capsule networks can capture. This plays a particularly relevant role for ALS, because ALS is commonly hypothesized to have a complex genetic architecture.

In practical terms, DiseaseCapsule has outperformed all state-of-the-art approaches when predicting ALS from individual genotype: it has achieved 87% accuracy on hold-out test data. This translates into a relative increase of 28% over PRS, so it remains with 28% fewer misclassified people in comparison with the current clinical standard. This establishes substantial (arguably even drastic) advantages in comparison with what was possible earlier. Analysing results further has revealed that DiseaseCapsule achieves 89.4% recall, which reflects that DiseaseCapsule identifies 64% of the patients with ALS who remain undiscovered according to clinical standards (PRS 70.2%), so it may miss the advantages of early intervention according to current practice.

DiseaseCapsule has also redeemed its two major theoretical promises for application in clinical practice: sustainable use of training input, which reduces costs and efforts when raising clinical data, as well as advances in terms of interpretability of predictions. The latter point has become obvious through experiments based on inspecting the individual capsules of the network. The experiments have revealed 922 candidate genes for being associated with ALS, many of which had not been pointed out before following standard GWAS protocols; note that all of them appear to be plausible according to their functional annotations.

Realizing these advantages has required overcoming various non-negligible technical hurdles. First, integrating whole-genome data means dealing with feature spaces whose dimensions are in the millions, which corresponds to the amount of polymorphic sites in the human genome. Here we have developed a protocol that yields gene-specific principal components. These gene-specific principal components can then be combined in non-linear ways to reflect non-linear interactions across genes, where non-linearities can span the entire genome.

Secondly, capsule networks had never been applied to whole-genome genotype data before. We have enabled this by means of an architecture that uses fully connected, instead of convolutional layers as early layers in the capsule networks. This preserves to capture interactions between genes across the whole genome to a maximal degree, and appropriately accommodates the sequential nature of the input.

While determining the exact reasons for the superiority of DiseaseCapsule in predicting ALS from genotype still requires further investigation, some plausible hypotheses can be raised already.

First, as already alluded to above, capsule networks have the potential to learn the intrinsic hierarchical structures that underlie the data. The enhanced capability to analyse complex biological relationships that underlie diseases (see also ref. [34]) explains the improved generalization over other models.

Second, DiseaseCapsule is able to pick up non-linear genetic interactions between variants (for example, epistasis) that have remained overlooked by the standard approaches.

To further investigate this hypothesis, we considered an objective function that addressed to find genes that supported classification in (the non-linear) DiseaseCapsule, but not in linear regression type schemes. Optimizing according to this objective function (as per a genetic algorithm) yielded a subset of 644 genes that significantly contributed to predicting ALS in DiseaseCapsule (accuracy 71.2%), while not working within the frame of logistic regression (accuracy 51.2%). Evidently, linear approaches remain blind to these 644 genes. Although these 644 genes still require further inspection, it is reasonable to assume that various genes among them have the potential to be of future use in exploring the heritability of ALS further.

Third, one can hypothesize that ALS follows an omnigenic model[12] to a non-negligible degree. Results for PRS corroborate this idea. Gradually relaxing the threshold for inclusion of variants from 5×10^{-8} to 5×10^{-2} increases the accuracy by 20%. This means that numerous variants of small effect contribute to establishing ALS, in addition to the core disease-causing genes. DiseaseCapsule follows a whole-genome approach that does not put significance thresholds on individual variants (or genes) to appropriately take this into account.

As already alluded to above, DiseaseCapsule requires less training data than other approaches to establish excellent performance. While the effects are obvious, the translation of the 'viewpoint invariance property' into the realm of genes and diseases still needs to be provided.

It is reasonable to hypothesize that capsule networks capture the core effects regardless of their distribution across the ancestors of the individuals, and their possible interference. Instead, other approaches may become confused by interfering pathways, so they need to be presented with all possible combinations of interacting pathways before reasonable conclusions can be drawn.

Of course, several open questions have been remaining, some of which point to further promising avenues of research. Such immediate ideas are to also integrate epigenetic information, for example, or adapt models to haplotype data, so as to use phasing information whenever available.

In addition, it appears sensible to develop a formal protocol for machine-learning-based approaches by which to identify (combinations of) variants that are associated with the diseases/phenotypes one examines. While such formal association schemes do not yet exist, we have suggested clear steps towards that goal. For example, DiseaseCapsule has already been able to deliver 922 genes that may be associated with ALS, which deserve to be investigated further.

In addition, fortunately, the field of explainable deep neural networks is moving fast. So, it is reasonable to expect that one will understand the molecular mechanisms that drive diseases from examining the non-linear deep-learning-based approaches further in the nearer future. This then will aid practitioners in their assessment of strategies for preventing and treating the diseases.

Methods

Data: Project MinE

The data we used are from Project MinE (https://www.projectmine.com), a large-scale study that aims to reveal the epi-/genetic mechanisms that underlie ALS, in the frame of globally concerted collaboration[45]. The data we use in this project are from the Dutch cohort of the project. We have complied with all relevant ethical regulations, and informed consent was obtained from participants. The data contain 7,213 healthy (also known as control) individuals and 3,192 individuals affected with ALS. The cohort counts 5,208 females and 5,197 males. All participants of the study were genotyped using Illumina 2.5M single nucleotide polymorphism (SNP) array.

QC. First, we annotated SNPs according to dbSNP137 and mapped them to hg19 as the reference genome. We first performed QC so as to remove low-quality SNPs and individuals of low quality overall, by using PLINK 1.9 (refs. [56,57]) (-geno 0.1 and -mind 0.1). We stratified individuals according to the genotyping platform, and subsequently performed a more stringent SNP QC (-geno 0.0, -maf 0.01, -hwe 1e-5 midp include-nonctrl). We kept only SNPs in autosomal regions, and filtered on the basis of differential missingness (-test-missing midp), excluding SNPs of a *P* value over 1×10^{-4}. As a result, we obtained 4,370,685 SNPs, all of which are contained in the intersection the four batches (see below) the data consist of (Supplementary Fig. 2).

Batch structure and batch effects. The dataset consists of four batches, pertaining to technical identifiers C1, C3, C5 and C44. The number of samples and the ratio of cases versus controls can vary substantially across batches: C1 contains 225 cases and 380 controls, C3 is 130 cases and 49 controls, and C5 no cases but 5,155 controls, whereas C44 finally contains 2,387 cases and 1,629 controls (Supplementary Table 8). It is important to realize that C5 and C44, both of which are highly imbalanced—C5 contains no cases, while the majority of C44 are cases—cover approximately 92.5% of the samples, and thus dominate the dataset.

This points to the importance of removing batch effects. Otherwise, predicting cases and controls can be confounded with predicting C44 from C5, at still decent performance rates.

To remove artefacts by which to distinguish C44 from C5, we considered only the 5,155 and 1,629 healthy individuals from C5 and C44, respectively. Any SNP that supports discrimination between C5 and C44 healthy individuals can be identified as signalling batch-induced effects, and thus should be removed from further analysis.

To filter for batch-effect-transporting SNPs, we computed a 2 × 2 contingency table for each SNP, where rows represent alleles (major or minor allele) and columns represent batches (C5 or C44). Entries in this table reflect allele counts per batch. Subsequently, we performed a Pearson chi-squared test on each table, and thereby obtained a *P* value for each SNP. Small *P* values indicated that the particular SNP transports batch effects. To correct for multiple testing, we used the Benjamini–Hochberg procedure [58] and filtered all SNPs according to their adjusted *P* values, removing SNPs with an adjusted *P* value of less than 0.05. Filtering out 8,664 potentially batch-effect-signal-carrying SNPs this way, we remained with 4,362,021 SNPs from 22 autosomes.

Dimensionality reduction

GWAS for SNP selection. The dimension of the feature spaces amounts to 4,362,021, agreeing with the number of SNPs that passed quality and batch effect control. On the one hand, this means that the number of SNPs one can work with is sufficiently high to transport relevant meaning. On the other hand, it means that the number of features is too large for machine learning approaches to not overfit. Therefore, as usual, the dimensionality of the data needs to be reduced for machine learning approaches to generalize, while preserving the expressiveness of the original set of 4,362,021 features.

To this end, we performed a GWAS using PLINK v1.9 (ref. [56]), and discarded SNPs that are very unlikely to carry disease-status-related signals. This reasoning is based on the fact that every SNP must carry an—albeit potentially rather weak—signal in its own right. Using only the training data (see below for descriptions of training, validation and test data), in agreement with the fact that regular GWAS makes use of disease status labels and thus classifies as part of the training process, we discarded all SNPs whose *P* values were greater than 0.05. Note that a threshold of 0.05 is considerably relaxed relative to the stringent threshold of 5×10^{-8} (ref. [59]) that is used in regular GWAS. Unlike in regular GWAS, however, we would like to keep as many potentially associated SNPs. So, we do not discard all SNPs whose individual signals are too weak in their own right, knowing that weak individual signals can accumulate to strong signals where SNPs interact in possibly non-additive constellations in the deep learning models that we use. As a result, 505,333 SNPs were retained in this step (Supplementary Fig. 3); signals of all SNPs discarded were found to be too weak to potentially play a role.

We further annotated all 505,333 SNPs using ANNOVAR (24 October 2019; latest version)[60], assigning them to genes ('gene-based annotation') based on the human reference genome (hg19). Genes were defined using the NCBI Reference Sequence (RefSeq) database. Each SNP could be assigned to at least one gene. When annotating SNPs with more than one gene, we kept track of the corresponding mapping relationships: if annotated as 'intergenic', SNPs were assigned to only the nearest gene. If annotated as non-intergenic with variant functions in different genes, we assigned the SNP to all related genes. As a result, SNPs were annotated with 18,279 genes overall, where the vast majority of genes have less than 200 SNPs annotated (Supplementary Fig. 4).

As usual, we also transformed genotype data according to minor allele information. Each SNP corresponds to a value $i \in \{0, 1, 2\}$ in each individual, where *i* is the minor allele count at the particular polymorphic site in the individual.

PCA. PCA[61] has been widely applied to exploit SNP data and demonstrated great effectiveness[62]. We recall that whole-genome PCA is not applicable for non-additive approaches to properly work, while gene-based PCA preserves to detect non-additive variant patterns across different genes. In agreement with that reasoning, we performed PCA for each collection of SNPs that became assigned to identical genes. In the following, we refer to that procedure as Gene-PCA.

Correspondingly, for each gene g, we constructed a matrix $\mathbf{A}_g \in \{0, 1, 2\}^{m\times n}$ where $m = 10{,}405$ is the total number of individual samples, and $n \in 1, ..., 1{,}383$ corresponds with the number of SNPs that became assigned to gene g; note that the maximum amount of SNPs assigned to a gene is 1,383, where however the number of genes with more than 200 SNPs assigned was small (see above). For an illustration, see Fig. 1.

PCA can remove noise and generate a robust compressed representation from input features. PCA is efficiently implemented on the basis of singular value decomposition: $\mathbf{A}_g$ is factorized into three matrices

$$\mathbf{A}_{m\times n} \approx \mathbf{U}_{m\times k}\mathbf{\Sigma}_{k\times k}\mathbf{V}^T_{k\times n} \tag{1}$$

Here $\mathbf{\Sigma}$ is a $k \times k$ (usually $k \ll n$) rectangular diagonal matrix of singular values σ_k; importantly, k is the number of principal components one will use. $\mathbf{U}$ and $\mathbf{V}$ are matrices whose columns are orthogonal unit vectors called the left and the right singular vectors of $\mathbf{A}$, respectively. To take into account that the input dimension n varies, we varied k, relative to n, accordingly: $k = 8$ for $n > 20$, $k = 4$ for $4 < n \leq 20$ and $k = 1$ for $1 \leq n \leq 4$.

The reduction in dimension we achieve through this procedure is from 505,333 down to 75,584, where each of the 75,584 dimensions corresponds to a principal component computed for one of the 18,279 genes that had become annotated with potentially relevant SNPs, as discussed above.

The architecture and parameters of DiseaseCapsule

DiseaseCapsule is a neural network model that takes a real-valued vector of length 75,584 as input, and generates binary-valued output, 0 for control/healthy and 1 for ALS/disease. In general, DiseaseCapsule can be flexibly adapted to input vectors of varying sizes, as long as the length of the input vectors does not exceed a certain upper limit; the procedure based on GWAS filtering and Gene-PCA described above warrants this for whole-genome input.

The architecture of DiseaseCapsule follows the architecture of the capsule network that was described in the seminal paper, with modifications to accommodate that the input does not reflect rectangular, pixel-structured image data, and the output is binary. In detail, DiseaseCapsule consists of three layers: a fully connected layer (FC), a primary capsule layer (PrimaryCaps) and a phenotype capsule layer (PhenoCaps); for an illustration and details, see Fig. 2). The initial fully connected layer replaces the convolutional layers used in the seminal application, reflecting that convolution addresses to process rectangularly arranged image data. However, here we expect interactions between all genes across the whole genome, which the fully connected layer reflects.

The FC layer consists of 150 (regular) neurons followed by ReLU as activation function[63]. During training, a dropout rate of 0.5 was used for FC to prevent overfitting. Correspondingly, the output of FC is a 150 × 1 tensor, so virtually a 150-dimensional vector. This, in turn, is the input for the PrimaryCaps layer.

The PrimaryCaps is the first ('low-level') capsule layer. As such, it incorporates convolutional operations. It consists of 32 primary capsules. Each of these involves four different convolutional filters, implementing a 5 × 1 kernel, operating at a stride of 2, with no zero padding. This means that each convolutional filter computes a 73 × 1 tensor (that is a 73-dimensional vector) from the 150-dimensional input (FC output) vector. Using four convolutional filters per capsule results in 73 vectors of length 4 per capsule. This yields 32 × 73 = 2,336 vectors of length 4 as the output of PrimaryCaps. We refer to each of these vectors as $\mathbf{u}_i$, $i = 1, ..., 2{,}336$. Further, as is common, one refers to each array of 73 such four-dimensional vectors as one primary capsule. If needed, we index primary capsules using $k \in \{1, ..., 32\}$. Importantly, all 73 $\mathbf{u}_i$ making part of one primary capsule k share their parameters when transiting to PhenoCaps, the last layer.

Finally, the output of PhenoCaps is used to derive predictions from. PhenoCaps consists of two 16-dimensional vectors $\mathbf{v}_j$, $j = 1, 2$, one referring to 'ALS' and one to 'Healthy'. Each phenotype capsule receives input from all 32 primary capsules (that is, virtually from all 2,336 $\mathbf{u}_i$) making part of PrimaryCaps.

To transform PrimaryCaps output into PhenoCaps input, so-called pose matrices $\mathbf{W}_{ij}$ are applied to the $\mathbf{w}_i$, which yields

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i \tag{2}$$

In our case, pose matrices are 4 × 16-dimensional, so the $\mathbf{u}_{j|i}$ are 16-dimensional. This corresponds with the dimensionality of PhenoCaps. Pose matrices are learnt during training. As mentioned above, pose matrices are shared for all (73) i that refer to an identical primary capsule k. This means that there are 32 pose matrices to be learnt for each $j = 1, 2$.

Subsequently, we performed dynamic routing, as a key feature of capsule networks, intended to improve on the pooling operations. Dynamic routing is an iterative procedure that converges quickly. Here we used three iterations. For the corresponding details, see Extended Data Fig. 1. According to the routing procedure, the input $\mathbf{s}_j$ to one of the PhenoCaps capsules evaluates as

$$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i} \tag{3}$$

where c_{ij} are the coupling coefficients, as determined through the dynamic routing procedure. In a rough description, first b_{ij} are computed by the iterative update $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$ (and initialized as zero), which rewards if $\mathbf{u}_{j|i}$ and $\mathbf{v}_j$ point in the same direction, which means that the output of primary capsule i agrees with phenotype capsule j. To turn b_{ij} into c_{ij} and ensure that $\sum_i c_{ij} = 1$, one eventually performs

$$c_{ij} = \frac{\exp\left(b_{ij}\right)}{\sum_i \exp\left(b_{ij}\right)} \tag{4}$$

to obtain the coupling coefficients.

Reflecting another key principle of capsule networks, one uses the squashing operation

$$\mathbf{v}_j = \mathrm{squash}(\mathbf{s}_j) = \frac{\left\|\mathbf{s}_j\right\|^2}{1 + \left\|\mathbf{s}_j\right\|^2}\frac{\mathbf{s}_j}{\left\|\mathbf{s}_j\right\|} \tag{5}$$

as a non-linear activation function that can process vectors. This ensures that the length of the input vectors $\mathbf{v}_j$ is between 0 and 1. Importantly, the length of capsule output vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ indicates the probability that the capsule 'is activated'.

To derive predictions from the $\mathbf{v}_j$, categorical cross entropy was employed as the loss function used during training.

Model training and testing

We randomly split the dataset into a set of samples used for training and validation set, which comprised 90% of individuals, and a test set, comprising the remaining 10% of individuals. Importantly, the test set is balanced, that is, the ratio of cases and controls is 1 (here: 520 cases and 520 controls), to ensure that models can be evaluated without misleading biases. For details, see Supplementary Fig. 5.

The entire dataset was used for Gene-PCA. As dimensionality reduction works in an unsupervised way, and thus does not require labels, this agrees with a generally applicable protocol. Subsequently, using the training-validation part of the data, fivefold cross-validation was performed to optimize the architecture and determine all other hyperparameters of the capsule network (DiseaseCapsule). To ensure a balanced evaluation during validation, validation splits were first randomly selected under the constraint of preserving a ratio of 1:1 between cases and controls. Subsequently, the cases in the remaining training data were upsampled to the same ratio, so as to avoid poor performance in prediction in particular for the minority class (Supplementary Fig. 5). This reflects a standard procedure in supervised machine learning. Upon having obtained all hyperparameters through cross-validation, the entire training-validation split, upsampled to ensure unbiased training, was used for training. This determines all parameters of the DiseaseCapsule network.

Specifically, we used the Adam algorithm[64] to optimize all parameters in the frame of the usual backpropagation algorithm. We used an initial learning rate of 0.0001, and decayed it by $\gamma = 0.8$ in each epoch using an exponential scheduler. Optimization ran for 30 epochs, operating at a batch size of 128.

As for the simple three-layer perceptron (MLP) and the basic CNN (consisting of four convolution layers and two dense layers) that we used for benchmarking. For details, see Supplementary Fig. 6. Models were trained for 30 epochs with a batch size of 128 using the Adam optimizer, matching the procedure used for DiseaseCapsule.

Model interpretation

Relating coupling coefficients with phenotype recognition. When running DiseaseCapsule on the 1,040 test samples, coupling coefficients c_{ij}, $i = 1, \ldots, 2{,}336$, $j = 1, 2$ are determined individually for each of the test samples. This is due to the fact that coupling coefficients are determined during the forward step, and thus do not correspond to parameters to be learned during training. According to the general principles of capsule networks, one can interpret coupling coefficients c_{ij} as the degree of activation by which $\mathbf{u}_i$ contributes to phenotype capsule j; large c_{ij} means that $\mathbf{u}_i$ makes a crucial contribution to activating j.

As is common, we also determine coupling coefficients c_{kj} that virtually measure the degree by which primary capsule k 'activates' phenotype capsule j. To obtain c_{kj} from the c_{ij}, one sums all coupling coefficients c_{ij} across all $\mathbf{u}_i$ that make part of k:

$$c_{kj} = \sum_{i\,\text{belongs to}\,k} c_{ij} \tag{6}$$

where j is either 'Healthy' or 'ALS', referring to one of the two Phenotype capsules. Recalling that c_{kj} differ for each individual sample, we eventually average the c_{kj} across the individuals to obtain a summarizing c_{kj} one can work with.

Selecting and annotating genes decisive for classification. Evaluating combinations of primary and phenotype capsules according to c_{kj} from equation (6) determines primary capsule 5 as the dominant driver to indicate that the phenotype capsule 'ALS' gets activated (Fig. 3a). In other words, genes that significantly contribute to activation of primary capsule 5 are potentially responsible for the development of ALS; in that, these genes are likely to be the predominant factor ('Determining genes decisive for classification' in Results). Thanks to capsule networks reflecting non-additive relationships, such genes can interact in arbitrary ways.

Computation of c_{kj} involves running DiseaseCapsule on all 18,279 genes selected. Here we would like to determine the genes that play an important role in activating primary capsule 5 in their own right. To do so, we consider each gene g, $g \in (1, 18{,}279)$ as exclusive input to the trained model. To implement this, we mask all other genes, that is, we set the values of all principal components that do not refer to the particularly picked gene g to zero. We do this for each individual sample. Subsequently, we run DiseaseCapsule on all the resulting 'one-gene-only' individual samples and note down the resulting coupling coefficients c^g_{kj}, for $k = 5$, $j =$ 'ALS', for each of the individuals. Computation of c^g_{kj} proceeds analogously to equation (6), when replacing the full individual sample with the 'one-gene-only' individual sample. Eventually, also here all individual c^g_{kj} are averaged across the individual samples to obtain a summarizing c^g_{kj} one can work with. We then kept all genes g whose c^g_{kj} was above the 95-percentile (Fig. 4a), amounting to 922 genes, as core genes decisive for classification.

To annotate the biological functions of these 922 genes, we employed g:Profiler[65] to perform common Gene Ontology and pathway enrichment analyses.

Selecting non-additive genes

Let $S \subset G$ be a subset of genes selected from the set G of 18,279 genes overall. Let further $\mathrm{ACC}_{\mathrm{DC}}(S)$ be the training accuracy achieved by Gene-PCA + DiseaseCapsule (DC) and $\mathrm{ACC}_{\mathrm{LR}}(S)$ be the training accuracy of Gene-PCA + logistic regression (LR), as the best-performing linear approach, when running on only genes S. Running DC and LR on S is done by setting values of principal components referring to genes not from S to zero, in full analogy as for single genes g, as described above.

To determine a good subset of genes that predominantly interact in non-linear constellations, we make use of a genetic algorithm that seeks to determine

$$\max_{S \subset G} \mathrm{ACC}_{\mathrm{DC}}(S) - \mathrm{ACC}_{\mathrm{LR}}(S) \tag{7}$$

that is, the set of genes S that delivers the greatest gains in terms of classification performance in the non-linear DC over the linear LR. To implement the genetic algorithm, we consider subsets of genes as 18,279-dimensional binary-valued vectors $(x_1, x_2, \ldots, x_{18{,}279}) \in \{0, 1\}^{18{,}279}$ where $x_g = 1$ if $g \in S$, that is, gene g belongs to S, and $x_g = 0$ otherwise. Representing sets of genes S this way, one can implement the common evolutionary operations of genetic algorithms, like 'selection', 'crossover', 'mutation' or 'fitness evaluation'. For solving the optimization problem equation (7), we employ 'segregative genetic algorithms', as available through the high-performance genetic algorithm toolbox Geatpy v2.6.0 (ref. [66]).

Subsets S were initialized randomly, and the genetic algorithm was run at a population size of 30 and a maximum of generations of 200 as a stopping criterion. The subset of genes S that were determined to maximize equation (7) can be considered optimal in terms of interacting in exclusively non-additive ways to establish the 'ALS' signal in the frame of the DiseaseCapsule network.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data used in this study are publicly available. The ALS data used in this study are from the Dutch cohort of Project MinE[8,11,45], and have been deposited at dbGaP Study Accession: phs003146.v1.p1. Amyotrophic Lateral Sclerosis online Database (ALSoD) is available at https://alsod.ac.uk/. The data of PD were downloaded from dbGaP Study Accession: phs000918.v1.p1 (refs. [46–49]). Source data for Fig. 4 and Supplementary Fig. 1 are provided with this paper.

Code availability

The source code of DiseaseCapsule is publicly available on GitHub (https://github.com/HaploKit/DiseaseCapsule). Another related code for reproducing results and generating figures in this study is publicly available at Zenodo (https://doi.org/10.5281/zenodo.7118988)[67].

References

1. Miller, R. G. et al. Practice parameter update: the care of the patient with amyotrophic lateral sclerosis: drug, nutritional, and respiratory therapies (an evidence-based review): report of the quality standards subcommittee of the American Academy of Neurology. *Neurology* **73**, 1218–1226 (2009).

2. Brown, R. H. & Al-Chalabi, A. Amyotrophic lateral sclerosis. *N. Engl. J. Med.* **377**, 162–172 (2017).
3. Kiernan, M. C. et al. Amyotrophic lateral sclerosis. *Lancet* **377**, 942–955 (2011).
4. Lautrup, S., Sinclair, D. A., Mattson, M. P. & Fang, E. F. Nad$^+$ in brain aging and neurodegenerative disorders. *Cell Metab.* **30**, 630–655 (2019).
5. de la Rubia, J. E. et al. Efficacy and tolerability of eh301 for amyotrophic lateral sclerosis: a randomized, double-blind, placebo-controlled human pilot study. *Amyotroph. Lateral Scler. Frontotemporal Degen.* **20**, 115–122 (2019).
6. Al-Chalabi, A. et al. An estimate of amyotrophic lateral sclerosis heritability using twin data. *J. Neurol. Neurosurg. Psychiatry* **81**, 1324–1326 (2010).
7. Parone, P. A. et al. Enhancing mitochondrial calcium buffering capacity reduces aggregation of misfolded sod1 and motor neuron cell death without extending survival in mouse models of inherited amyotrophic lateral sclerosis. *J. Neurosci.* **33**, 4657–4671 (2013).
8. Van Rheenen, W. et al. Common and rare variant association analyses in amyotrophic lateral sclerosis identify 15 risk loci with distinct genetic architectures and neuron-specific biology. *Nat. Genet.* **53**, 1636–1648 (2021).
9. Nguyen, H. P., Van Broeckhoven, C. & van der Zee, J. Als genes in the genomic era and their implications for ftd. *Trends Genet.* **34**, 404–423 (2018).
10. Ryan, M., Heverin, M., McLaughlin, R. L. & Hardiman, O. Lifetime risk and heritability of amyotrophic lateral sclerosis. *JAMA Neurol.* **76**, 1367–1374 (2019).
11. Van Rheenen, W. et al. Genome-wide association analyses identify new risk variants and the genetic architecture of amyotrophic lateral sclerosis. *Nat. Genet.* **48**, 1043–1048 (2016).
12. Boyle, E. A., Li, Y. I. & Pritchard, J. K. An expanded view of complex traits: from polygenic to omnigenic. *Cell* **169**, 1177–1186 (2017).
13. Génin, E. Missing heritability of complex diseases: case solved? *Hum. Genet.* **139**, 103–113 (2020).
14. Shi, H., Kichaev, G. & Pasaniuc, B. Contrasting the genetic architecture of 30 complex traits from summary association data. *Am. J. Hum. Genet.* **99**, 139–153 (2016).
15. Tam, V. et al. Benefits and limitations of genome-wide association studies. *Nat. Rev. Genet.* **20**, 467–484 (2019).
16. Moore, J. H. The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Hum. Hered.* **56**, 73–82 (2003).
17. Jiao, S. et al. Genome-wide search for gene–gene interactions in colorectal cancer. *PLoS ONE* **7**, e52535 (2012).
18. Hung, H. et al. Detection of gene–gene interactions using multistage sparse and low-rank regression. *Biometrics* **72**, 85–94 (2016).
19. Ferrario, P. G. & König, I. R. Transferring entropy to the realm of gxg interactions. *Brief. Bioinformatics* **19**, 136–147 (2018).
20. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
21. Montufar, G. F., Pascanu, R., Cho, K. & Bengio, Y. On the number of linear regions of deep neural networks. *Adv. Neural Inf. Process. Syst.* **27**, 2924–2932 (2014).
22. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **25**, 1097–1105 (2012).
23. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
24. Alzubaidi, L. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **8**, 1–74 (2021).
25. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations* 1–14 (Computational and Biological Learning Society, 2015); https://arxiv.org/pdf/1409.1556.pdf
26. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016); https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90
27. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 4700–4708 (IEEE, 2017); https://ieeexplore.ieee.org/document/8099726
28. Chakraborty, S. et al. Interpretability of deep learning models: a survey of results. In *2017 IEEE Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)* 1–6 (IEEE, 2017); https://ieeexplore.ieee.org/document/8397411
29. Hestness, J. et al. Deep learning scaling is predictable, empirically. *CoRR* **abs/1712.00409** (2017).
30. Ching, T. et al. Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, 20170387 (2018).
31. Wainberg, M., Merico, D., Delong, A. & Frey, B. J. Deep learning in biomedicine. *Nat. Biotechnol.* **36**, 829–838 (2018).
32. Sabour, S., Frosst, N. & Hinton, G. E. Dynamic routing between capsules. *Adv. Neural Inf. Process. Syst.* **30**, 3856–3866 (2017).
33. Sabour, S., Frosst, N. & Hinton, G. Matrix capsules with em routing. In *6th International Conference on Learning Representations, ICLR 2018* (OpenReview.net, 2018); https://openreview.net/pdf?id=HJWLfGWRb
34. Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C. & Collins, J. J. Next-generation machine learning for biological networks. *Cell* **173**, 1581–1592 (2018).
35. Wang, L. et al. An interpretable deep-learning architecture of capsule networks for identifying cell-type gene expression programs from single-cell rna-sequencing data. *Nat. Mach. Intell.* **2**, 693–703 (2020).
36. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V. & Fotiadis, D. I. Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.* **13**, 8–17 (2015).
37. Curbelo Montañez, C. A., Fergus, P., Chalmers, C. & Hind, J. Analysis of extremely obese individuals using deep learning stacked autoencoders and genome-wide genetic data. In *Computational Intelligence Methods for Bioinformatics and Biostatistics: 15th International Meeting, CIBB 2018, Caparica, Portugal, September 6–8, 2018, Revised Selected Papers 15* (eds Raposo, M. et al.) 262–276 (Springer, 2020).
38. He, B. et al. Ai-enabled in silico immunohistochemical characterization for Alzheimer's disease. *Cell Rep. Methods* **2**, 100191 (2022).
39. Chen, D. et al. A stacking framework for multi-classification of alzheimer's disease using neuroimaging and clinical features. *J. Alzheimer's Dis.* **87**, 1627–1636 (2022).
40. Xie, C. et al. Amelioration of Alzheimer's disease pathology by mitophagy inducers identified via machine learning and a cross-species workflow. *Nat. Biomed. Eng.* **6**, 76–93 (2022).
41. Li, X., Liu, L., Zhou, J. & Wang, C. Heterogeneity analysis and diagnosis of complex diseases based on deep learning method. *Sci. Rep.* **8**, 1–8 (2018).
42. Greenside, P., Shimko, T., Fordyce, P. & Kundaje, A. Discovering epistatic feature interactions from neural network models of regulatory dna sequences. *Bioinformatics* **34**, i629–i637 (2018).

43. Yin, B. et al. Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype. *Bioinformatics* **35**, i538–i547 (2019).
44. Zhang, S. et al. Genome-wide identification of the genetic basis of amyotrophic lateral sclerosis. *Neuron* **110**, 992–1008 (2022).
45. Consortium, P. M. A. S. et al. Project mine: study design and pilot analyses of a large-scale whole-genome sequencing study in amyotrophic lateral sclerosis. *Eur. J. Hum. Genet.* **26**, 1537 (2018).
46. Auer, P. L. et al. Imputation of exome sequence variants into population-based samples and blood-cell-trait-associated loci in African Americans: NHLBI GO Exome Sequencing Project. *Am. J. Hum. Genet.* **91**, 794–808 (2012).
47. International Parkinson's Disease Genomics Consortium (IPDGC) & Wellcome Trust Case Control Consortium 2 (WTCCC2). A two-stage meta-analysis identifies several new loci for Parkinson's disease. *PLoS Genet.* **7**, e1002142 (2011).
48. Nalls, M. A. et al. Large-scale meta-analysis of genome-wide association data identifies six new risk loci for Parkinson's disease. *Nat. Genet.* **46**, 989–993 (2014).
49. Nalls, M. A. et al. Neurox, a fast and efficient genotyping platform for investigation of neurodegenerative diseases. *Neurobiol. Aging* **36**, 1605.e7–1605.e12 (2015).
50. Leal, S. S. & Gomes, C. M. Calcium dysregulation links als defective proteins and motor neuron selective vulnerability. *Front. Cell. Neurosci.* **9**, 225 (2015).
51. Van Spronsen, M. & Hoogenraad, C. C. Synapse pathology in psychiatric and neurologic disease. *Curr. Neurol. Neurosci. Rep.* **10**, 207–214 (2010).
52. Lepeta, K. et al. Synaptopathies: synaptic dysfunction in neurological disorders—a review from students to students. *J. Neurochem.* **138**, 785–805 (2016).
53. Ikemoto, A., Nakamura, S., Akiguchi, I. & Hirano, A. Differential expression between synaptic vesicle proteins and presynaptic plasma membrane proteins in the anterior horn of amyotrophic lateral sclerosis. *Acta Neuropathol.* **103**, 179–187 (2002).
54. Burk, K. & Pasterkamp, R. J. Disrupted neuronal trafficking in amyotrophic lateral sclerosis. *Acta Neuropathol.* **137**, 859–877 (2019).
55. Südhof, T. C. Neuroligins and neurexins link synaptic function to cognitive disease. *Nature* **455**, 903–911 (2008).
56. Chang, C. C. et al. Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience* **4**, s13742-015-0047-8 (2015).
57. Purcell, S. & Chang, C. Plink 1.9 beta. *PLINK 1.9* http://www.cog-genomics.org/plink/1.9/ (2015).
58. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B* **57**, 289–300 (1995).
59. Consortium, I. H. et al. A haplotype map of the human genome. *Nature* **437**, 1299 (2005).
60. Wang, K., Li, M. & Hakonarson, H. Annovar: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res.* **38**, e164 (2010).
61. Pearson, K. LIII. on lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901).
62. Price, A. L., Zaitlen, N. A., Reich, D. & Patterson, N. New approaches to population stratification in genome-wide association studies. *Nat. Rev. Genet.* **11**, 459–463 (2010).
63. Nair, V. & Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning* (eds Fürnkranz, J. et al.) 807–814 (Omnipress, 2010).
64. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations* (Ithaca, NY: arXiv.org, 2015).
65. Raudvere, U. et al. g: Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic Acids Res.* **47**, W191–W198 (2019).
66. Jazzbin et al. geatpy: the genetic and evolutionary algorithm toolbox with high performance in Python. *Geatpy* http://www.geatpy.com/ (2020).
67. Luo, X., Kang, X. & Schönhuth, A. Diseasecapsule: v1.0.0. *Zenodo* https://doi.org/10.5281/zenodo.7118988 (2022).

Acknowledgements

X.L. and X.K. were supported by the Chinese Scholarship Council. A.S. was supported by the Dutch ALS Foundation via grant AV20190010. A.S. was also supported by the Dutch Scientific Organization through Vidi grant 639.072.309 during the early stages of the project. A.S. further received funding from the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreements no. 956229 (ALPACA) and no. 872539 (PANGAIA). In addition, we thank B. Yin and S. Ray for their helpful discussions in the early stage of this project. We thank M. Balvert for helpful discussions in the early stages and insightful suggestions when revising the manuscript. We also thank K. van der Eijck and J. Veldink for providing data and helpful information about the mechanisms of ALS. Finally, we thank the International Parkinson's Disease Genomics Consortium (IPDGC), NINDS, NIA and DoD USAMRAA proposal, 10064005/11348001, for providing the PD data.

Author contributions

X.L., X.K. and A.S. developed the method. X.L. and X.K. implemented the code and conducted the data analysis. X.L., X.K. and A.S. wrote the manuscript. All authors read and approved the final version of the manuscript.

Funding

Open access funding provided by Universität Bielefeld.

Competing interests

The authors declare that they have no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-022-00604-2.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00604-2.

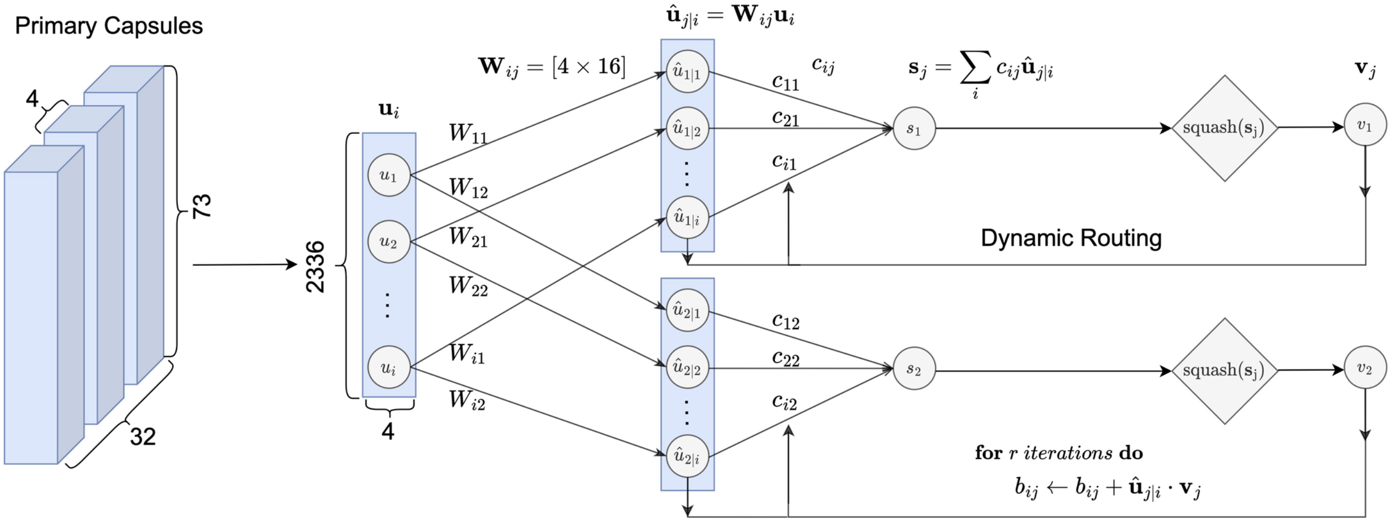
Correspondence and requests for materials should be addressed to Alexander Schönhuth.

Peer review information *Nature Machine Intelligence* thanks Evandro F. Fang and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

https://doi.org/10.1038/s42256-022-00604-2



Extended Data Fig. 1 | The schematic diagram of the dynamic routing algorithm. The PrimaryCaps layer outputs 32 × 73 = 2336 vectors $\mathbf{u}_i (i = 1, \ldots, 2336)$, which are then transformed into $\hat{\mathbf{u}}_{j|i} (j = 1, 2)$ by premultiplying pose matrices $\mathbf{W}_{ij}$. Note that pose matrices are learned during training. The right half of this figure shows the iterative dynamic routing procedures. The coupling coefficients c_{ij} indicate the probability that primary capsule i agrees with phenotype capsule j. The input $\mathbf{s}_j$ to one of the phenotype capsules is subsequently processed by using the squashing operation. The output $\mathbf{v}_j$ is used to update the b_{ij} and c_{ij} until the model converges.

nature research

Corresponding author(s): Alexander Schönhuth

Last updated by author(s): Dec 9, 2022

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a | Confirmed

□ ☒ The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement

□ ☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly

□ ☒ The statistical test(s) used AND whether they are one- or two-sided
Only common tests should be described solely by name; describe more complex techniques in the Methods section.

□ ☒ A description of all covariates tested

□ ☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons

□ ☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)

□ ☒ For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted
Give P values as exact values whenever suitable.

□ ☒ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings

□ ☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes

□ ☒ Estimates of effect sizes (e.g. Cohen's d, Pearson's r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection

PLINK v1.9;
ANNOVAR (Oct 24, 2019; latest version);
Geatpy v2.6.0;
Python v3.7;
PyTorch v1.5.0 (GPU);
TensorFlow v2.1.0 (GPU);
sklearn v0.22.2;

Data analysis

The source code of DiseaseCapsule is publicly available on GitHub: https://github.com/HaploKit/DiseaseCapsule. Other related code for reproducing results and source data for generating figures in this study is publicly available at Zenodo: https://doi.org/10.5281/zenodo.7118988

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The ALS data used in this study has been deposited at dbGaP database under accession number: phs003146.v1.p1
Amyotrophic Lateral Sclerosis online Database (ALSoD) is available at https://alsod.ac.uk/.
The data of Parkinson's disease was downloaded from dbGaP Study Accession: phs000918.v1.p1

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences □ Behavioural & social sciences □ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size
All 10405 ALS samples are from Dutch cohort of Projet MinE https://www.projectmine.com/

Data exclusions
No data were excluded.

Replication
Validated on 11402 Parkinson's disease samples, which are publicly available. Results show that the replication was successful (like in ALS data, our method DiseaseCapsule achieved the best prediction performance).

Randomization
Samples were allocated randomly.

Blinding
Blinding is not relevant because we just want to predict the phenotypes(disease or healthy) from genotype profiles.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study	
☒	□	Antibodies
☒	□	Eukaryotic cell lines
☒	□	Palaeontology and archaeology
☒	□	Animals and other organisms
□	☒	Human research participants
☒	□	Clinical data
☒	□	Dual use research of concern

Methods

n/a	Involved in the study	
☒	□	ChIP-seq
☒	□	Flow cytometry
☒	□	MRI-based neuroimaging

Human research participants

Policy information about studies involving human research participants

Population characteristics
The ALS samples are from Dutch population in Project MinE. This dataset contains 3192 ALS patients and 7213 health individuals. The cohort counts 5208 females and 5197 males. Phenotypes (e.g. disease status, gender) of each individual are displayed in the .fam file under dbGap accession number: phs003146.v1.p1

Recruitment
ALS samples were collected in four batches. Informed consent was obtained by participants.

Ethics oversight
UMC Utrecht

Note that full information on the approval of the study protocol must also be provided in the manuscript.