Supplementary Methods:

OMIM file types: OMIM uses 5 symbols to prefix the MIM records (+, *, #, % and *none*). % describes “mendelian phenotype or locus, molecular basis unknown”, * describes “gene with known sequence”, + describes “gene with known sequence and phenotype”, # describes “phenotype description, molecular basis known”, % describes “mendelian phenotype or locus, molecular basis unknown” and *none* describes “other, mainly phenotypes with suspected mendelian basis”. (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=Limits&DB=omim).

Performance and problems of MMTx. MetaMap[1] is a program that maps text to the Unified Medical Language System (UMLS)[2] metathesaurus (MTH) concepts. MetaMap Transfer (MMTx) is a Java implementation of MetaMap. The program segments text into phrases, and links the phrases to the best matching concept by using synonymy (this is an integral part of the MTH). MMTx also exploits lexical information from other sources, hereby generating lexical variants to find candidate mappings to the MTH. This procedure adds computable semantics to the text [3]. MMTx has a number of well known problems including co-reference problems, difficulty with word sense ambiguity, over calling abbreviations, coordinating conjunction problems, implicit meaning difficulties etc. (see Divita et al. 2004[3] for details). Most if not all of these problems will lead to deterioration in recall and precision of the program when parsing the OMIM records. To investigate what we could expect in terms of recall and precision of MMTx on text such as the OMIM records, we searched the literature for performance documentation on comparable types of text, where a similar setup as ours had been compared to manual curation. These studies report recall in the range 0.55 to 0.72, and precision in the range

0.50 to 0.56[4, 5], and we expect the performance of MMTx to be similar to this in our study. It has been shown that the precision and recall of MMTx can be improved if the UMLS vocabulary is pruned for irrelevant categories[5]. MMTx can also be combined with a number of different negation algorithms (e.g. NegEx[6]). We believed that a negation algorithm would not improve the performance in this study, because in OMIM negations are often used to describe words that are relevant in a record. E.g., in "MIM 117550 Sotos syndrome" the negation "although Sotos syndrome in its classically described form was not present, Robertson and Bankier (1999) concluded that this entity might reflect a related, perhaps allelic, condition."

Here the entity "Sotos syndrome" is subject to the negation, but it is also clearly relevant in the phenotypic description.

Comprising a high-confidence phenotype association benchmarking data set.

Identifying high-confidence references of annotated medical text is a well known problem when evaluating the performance of different methods for text-mining medical text[3]. For our purpose we needed a very big reference of different text records that were phenotypically overlapping. In other words we need different natural language text records describing phenotypes, and a knowledge of which records describe overlapping or identical phenotypes.

OMIM has an intricate way of cross-linking records based on pair wise relevance in the discussion of the main entry. If a curator finds that OMIM record B is of relevance to the discussion in OMIM record A, the curator will reference record B in record A, thus establishing a link between the two records. See **Supplementary Table 1** online for examples of cross-linked records. In many instances the link is established based on a cited reference in the literature, and many of the links between OMIM records will thus be based on bidirectional phenotypic overlap between the linked records.

In order to benchmark our phenotype association score, we investigated if a subset of these reference links could be exploited to create a high-quality phenotype association reference standard. If so, an association in the reference network would be based on an expert curator's opinion on what is phenotypically overlapping, additionally the link will often be based on cited literature references by experts in the particular diseases discussed in the records. As described above, not all OMIM records are phenotypic descriptions. In order to extract a sub-network that only cross-links records with phenotype information, we restricted ourselves to links between files with # or % prefixes (see **OMIM file types** for more information).

We started by parsing the OMIM text files linking all files in %, and # prefixed files if one file referenced the other. If one file (A) referenced the other (B) we used this link in the reference network. Thus, we did not consider it necessary that there was both a link from A to B and B to A. This network consisted of roughly 7,000 links, and only includes links between the OMIM files that actually represent phenotypes.

To systematically verify that this network could in fact be considered a set of high-confidence record pairs with a high degree of phenotypic overlap, we extracted a random set of 100 links and manually investigated the nature of the links between the files and the context leading to the link between the files. The manual analysis of this set of randomly chosen links from the full network can bee seen in (**Supplementary Table 1,** online). All examples mentioned below are from this table. The investigation revealed that the network consisted of links of different types. Roughly, there were five different types of links. *Type 1)* Links between two distinct records describing the same disease e.g., Ichthyosis Lamellar I and ichthyosis lamellar II, *type 2)* links between different subtypes of the same main disease e.g. Walker-Warburg syndrome and Fukuyama congenital muscular dystrophy are linked, because they are both congenital muscular dystrophies with a clear clinical overlap, *type 3)* links between diseases where one is part of the clinical spectrum of the other e.g. Pilomatrixoma and Rubinstein-Taybi syndrome are linked because of a cited reference stating that pilomatrixomas occour in Rubinstein-Taybi syndrome, *type 4)* links based on overlap in clinical traits between two distinct diseases e.g., Bardet-Biedl syndrome is linked to Meckel syndrome because there is a cited reference stating that the combination of cystic kidney dysplasia and polydactyly occurs both in Meckel syndrome and Bardet-Biedl syndrome, and *type 5)* links not due to phenotypic associations between the linked files. Categories 1-4 are called *True Positive* (TP) links, as they reflect high-confidence phenotypic overlap between the linked records, and category 5) are call*ed False Positive* (FP) links. There are various types of false positive links e.g. referencing between files based on similar modes of inheritance, due to contiguous gene syndromes etc.

The manual curation of the 100 randomly extracted associations showed that the links were highly enriched in TP links, and there were only 6 FP links. Based on this investigation we predict that over 90% of the extracted 7,000 file pairs in the benchmarking reference set have a high degree of phenotypic overlap. We used this data set to benchmark our computational phenotype association score against.

MIM vector density . The total size of the phenotype vectors is 18,429 dimensions (different terms) and on average an OMIM record is matched to 55 terms with a standard deviation of 61. This means that the vectors are very sparse and in order to avoid artificially high similarity between vectors with few but overlapping terms, we discarded all vectors with less than five dimensions with a value different from zero. This way, there should be enough data to pull them away from each other in phenotype space even if the same arbitraray term is mapped to two different phenotypes. This issue should also be compensated for by the term weighing as it will reduce the impact of common terms. The number of dimensions in the vector can be reduced with the use of a concept hierarchy to collaps dimensions for similar concepts. However, this would also reduce the resolution of the vectors and as our tests show a good performace of the scores based on leaf concepts, we decided against using a hierarchy. This should not be confused with the use of synonyms, which is already taken care of as synonyms describing the same condition are mapped to a single concept.

MIM vector term weighing. As some clinical traits are very prevalent in records (e.g. mental retardation is listed in 1,330 records), we applied a weighting scheme to the found terms, in order for the importance of the term as a clinical descriptor to be an inherent characteristic of the vector. This weight is called “term frequency inverse document frequency” (*tf-idf*) [7]:

$$tf\text{-}idf = tf \cdot \log(D/d)$$

Where *tf* is the importance of the term in the particular record (i.e. the fraction of all terms in that document being the weighted term), *D* is the total amount of records and *d* is the number of records in which the term occurs.

The effect of this procedure is that terms in each record (and thus vector) are awarded a weight in relation to the number of occurrences of that term in the record, relative to the occurrences of the term in all of OMIM. Furthermore the score is normalized for the length of the particular record and the total length of all records in OMIM. This weight removes negative bias towards short records, and positive bias towards longer records, as the length of the record and the total length of all OMIM records are taken into account when scoring terms in single records (and vectors).

Impact of the tf-idf weighing scheme on the predictive quality of the phenotype data: To investigate the impact of the weighing scheme on the predictive quality of the data, we have made a benchmark of the phenotype similarity scores where we try three different weight scenarios. We made three sets of phenotype association scores between all OMIM records. One set was made without the weight, one set with only the *idf* part of the weight, and the last set with the whole *tf-idf* weight. We benchmarked these three sets, and can clearly see that the weight improves the performance of the phenotype association score. If we consider the performance over the whole interval of scores from zero to one, the set without the weight performs the worst, the set with only the idf weight performs better than without the weight, but worse than the set with the tf-idf weight. Herby we show that we get optimal performance over the whole score interval when we use the tf-idf weight (**Supplementary Figure 7** online**)**. The general effect of removing the weight shifts the calibration curve downwards, showing that the weight highly increases the accuracy of the phenotype association score. We note outliers in the 0.7-0.8 bin for both the idf and the unweighted calibration curve. This is because there is a very small overlap between the record pairs scoring in this interval, and the benchmarking reference.

Robustness of the phenotype vector and cosine similarity measure for identifying phenotypic overlap.

To investigate the robustness of the method of constructing phenotype vectors and using the cosine score as a measure of phenotypic overlap, we constructed word vectors based on a different text body, weighing scheme, and vocabulary. We downloaded GeneCards files for each gene in the database, and extracted the Bioalma disease relationships for each gene. Bioalma is a proprietary data set, but is used in various academic settings, such as the GeneCards and Babelomics[8] databases. Bioalma links genes to disease terms by data-mining Medline, and scores the gene-disease relationship based on co-occurrence of the gene and the disease term. Furthermore Bioalma, uses their own vocabulary for mapping the terms extracted from Medline. A weight is applied for each term in relation to the gene by comparing the observed number of documents where the elements co-appear, and the number of documents where both appear independently. This result is then compared to an expected value based on a hypergeometric distribution (see urls: (http://www.genecards.org/info.shtml#AKS) (http://www.bioalma.com/aks2/Whitepapers/Identifying_biomedical_concepts.pdf)).

Bioalma disease terms and weights were directly used to construct phenotype vectors in the same way as previously described. This was done for each gene in GeneCards. We then calculated the cosine distance between the constructed vectors and benchmarked the performance of these vectors against the high-quality phenotype association reference standard previously generated (**Supplementary Figure 8)**. Because Bioalma vectors represented single genes, we had to link the genes to OMIM records to be able to benchmark the Bioalma vectors. This was done using Ensembl mart (http://dec2005.archive.ensembl.org/Multi/martview). The performance of the Bioalma vectors is worse, but clearly comparable to the performance of the OMIM based phenotype vectors where MMTx was used for parsing and tf-idf was used as a weighing scheme. As the Bioalma vectors are constructed very differently than our phenotype vectors, the comparable performance shows that the phenotype-vector cosine score is a robust measure for calculating phenotypic overlap between entities based on medical natural language text. Furthermore this analysis shows that our method for constructing phenotype vectors is optimal compared to the vectors we could construct in this other way.

Performance of phenotype similarity scheme on phenotypes with same molecular basis.

Although the above tests convincingly show that our phenotype similarity score is able to identify phenotypic overlap between OMIM entries, we also wanted verify that it can be used to identify phenotypes with a shared molecular basis. To this end, we compared the distribution of pairwise similarity scores between OMIM records that are associated with the same gene to the score distribution of random pairs of OMIM records. Some records in OMIM are meta records that describe phenotypes that are also described in more specific records, and both the meta record and the specific record can be associated with the genes relevant for the phenotype. These meta records are usually prefixed with a '#', and in order to avoid any bias caused by the relation between specific and meta records, associated with the same gene, we only used those records that were prefixed with a '#'. This way, we could be reasonably sure to only use descriptions of different phenotype. As can be seen from **Supplementary Figure 10** the two distributions are clearly different. A Kolmogorov-Smirnov two-sample test also finds the distributions to be significantly different with a p-value < 2.2e-16 (D=0.6781). In other words the phenotype similarity scores between phenotypes associated with the same gene are significantly higher than the random scores. Similar results were obtained also if we only used OMIM records not prefixed with a '#', or included all OMIM records.

Bias in the phenotype association scheme. To investigate other potential differences between phenotype associations in the training and validation set, we examined the connectivity among the phenotypes (i.e. the number and strengths of associations to other phenotypes). We noticed that there is a difference because records with no identified disease genes/proteins are in some instances more or less draft versions of a final record. In contrast, records reporting data on genes known to be involved in disease are often more meticulous in describing the phenotype and more well annotated, than files reporting a critical interval and a phenotype, but no identified gene. In this way there is a bias, against poorly described disease records. We believe this is the main reason for the lower rate of good hits when we predict compared to when we validate (**Supplementary Table 4**, online). However, we can not do anything about this as we can only work with the data at hand. Another reason for the lowered rate of good hits in the predictions compared to the validations is most likely that many linkage intervals are only confirmed by one study, and we have also noted curation mistakes in OMIM where the linkage interval specified in morbid map did not match the interval specified in the article reporting the linkage analysis. Furthermore old linkage studies rely on versions of the human genome that are outdated, and positional data from older studies can in such cases be incorrect. This is also noted by [9].

Comprising a high confidence protein interaction set. This set was comprised of high confidence small scale data (< 5 human interactions per study) (downloaded from MINT, BIND and IntAct). Furthermore, KEGG Enzymes involved in neighboring steps (ECrel) and KEGG annotated protein protein interactions (PPrel) [10] were included. Finally peer-reviewed data from Reactome [11] on proteins involved in the same complex, indirect complex reactions, or neighboring reactions were also included. This lead to a total set of ~35,000 non-redundant high-confidence interactions.

Making an interaction score. To get a probabilistic confidence score for all interactions in the network, we implemented a scoring scheme based on a topological scoring method reported in the work by de Lichtenberg et al. [12]. Every interaction was assigned a raw score (*RS*) from zero to - ∞, based on the topology of the network surrounding the interaction (i.e. number of non-shared interaction partners):

$$RS = -\log((NS_1 + 1) \cdot (NS_2 + 1))$$

NS_1 and NS_2 are the amount of non-shared interaction partners of proteins 1 and 2 respectively. The closer the raw score is to zero the fewer non-shared interaction partners. Besides network topology other issues are relevant when considering interaction confidence: 1) Interactions from large-scale experiments are generally of lower quality and contain more false positives, than interactions from small-scale experiments [13]. 2) An interaction is more reliable, if it has been shown in more than one independent interaction experiment [13]. To take these two issues into account we devised the following equation for post-processing the raw score of a given interaction:

$$\text{Score} = RS \ / \sum_{i=1}^{N} 1/\log(\text{int}\, i)$$

Where *i* is a publication showing the interaction and (int *i*) is the number of interactions in publication *i*. Thus for every interaction publication *i* showing a given interaction, the raw score (*RS)* is divided by one over the logarithm of the number of interactions in the experiment (to compensate for issue 1). If more than one interaction experiment has shown the interaction, the raw score is divided by the sum of these factors, for the individual interaction publications *i* (to compensate for issue 2). The final score is a multiplication of the raw score and the post processing factor. To convert the scores to probabilistic confidence scores we fitted a calibration curve of the score against overlap in a high confidence interaction set of ~35.000 human interactions. The curve shows that the interaction score is directly correlated with the probability of overlap with interactions in the high confidence set (**Supplementary Fig. 3**) and thus a reliable indication of interaction confidence.

Investigation of bias in the protein interaction data. To investigate the effect of this potential bias we repeated the training and crossvalidation of the Bayesian predictor using only large scale protein interaction data. Such data was defined as interaction data from a publication of more than 100 pairwise interactions. As these data are produced in large automated screens, and much of it comes from proteome wide screens in non-human organisms (~50% is from yeast, data not shown), this data should not be biased in relation to single molecular interactions in specific human diseases. We also investigated the composition of datasets in the large scale data, by downloading the abstracts from PubMed and extracting the titles of the 161 publications in the dataset. The titles were manually investigated for statements that could bias the data in the publication in relation to human disease. Out of the 161 publications 6 revealed potential bias in relation to four human diseases (Huntingtons Disease, Alzheimer's Disease, Treacher Collins syndrome Wiskott-Aldrich syndrome, respectively). However, none of these diseases are represented in the true positive hits in our large scale benchmarking results (data not shown) ruling out that these datasets have biased the training and validation based on the large scale dataset.

Intervals used for benchmarking and predicting. The predictor was implemented on OMIM records using the reported cytogenetic intervals. We are aware that in some cases there is more specific marker information that could be used to refine the intervals, but often the relationships between the markers and the critical intervals are not straight forward. Sometimes the critical interval is between two (or more) markers, sometimes it surrounds single or multiple markers. These relationships are difficult to extract automatically, hence, to avoid mistakes, we decided to use the whole cytogenetic intervals for our predictions.

Comparison of performance to other computational methods. Below we compare a number of other computerized methods for prioritizing (enriching) candidates in linkage intervals associated with disease. Traditionally these methods are tested by measuring average times enrichment of positional probability, even if this measure does not allow for rigorous comparison of the methods. If a method is able to rank the true candidate in the top 10% of all candidates in 50% of the linkage intervals, there is a tentimes enrichment in the successful predictions intervals and fivetimes enrichment on average. The results are summarized in **Supplementary Table 3** online. *Lage et al:* in our method we have tested 1,404 linkage intervals, containing an average of 108.8 genes. The linkage interval sizes were randomized so that they have a distribution similar to the intervals in OMIM morbidmap for which no gene has been identified. With a threshold over 0.1 we report 298 genes correctly ranked number one 371 not correctly ranked number one 1, and 735 cases of no rankings over this threshold. This leads to a 108.8 times enrichment in 21.2 % of the cases and no enrichment in 78.8 % of the cases. For the successful predictions this gives an average of 108.8 times enrichment and an average of 23.1 times enrichment for all tested linkage intervals. *Perez-Iratxeta et al.* test their predictor on 100 linkage intervals of an average size of 30MB and report a 47% chance of the correct candidate being in the top 8, and 62% percent chance that the candidate is in the top 30 of all scored candidates [14]. This implies that there is a 38% chance of not scoring the candidate. In 30 MB areas where it is a reasonable assumption that there is 300 genes [15], this translates to a 47% chance of 38 times enrichment, 15% chance of 10 times enrichment and 38% chance of no enrichment, yielding an average of 31.2 times enrichment for successful predictions and an average of 19.4 times enrichment for all predictions. *van Driel et al.* only test their method on 10 linkage intervals , and report a ten times enrichment on average [16]. It seems fair to expect that their method also has produced false positive or no enrichment predictions, but the extent of this is hard to evaluate. *Turner et al.* use 163 different sets of intervals containing 100, 500 and 1000 genes respectively and observe an average enrichment of 5.04 times, 4.06 times, and 6.62 times for all predictions in these intervals and an average of 12, 29, and 42 times enrichment for successful predictions only, giving an average of 27.7 times enrichment for successful predictions in all sets of intervals [17]. *Franke et al.* use a set of 409 linkage intervals of 150 genes, and report that in 24% of the cases the correct gene is in the top 10 of the rankings giving an 15 times enrichment for successful predictions and a 3.6 times average enrichment for all linkage intervals [18]. *Freudenberg et al.* test their method on 878 disease genes, and at a reasonable threshold report a 33 times enrichment in 1/3 of the cases a seven times enrichment in 1/3 of the cases and in 1/3 of the cases no enrichment [19]. This yields an average of 17 times enrichment for successful predictions and 13.3 times enrichment for all predictions. *Adie et al.* use a sequence based method on 734 known disease genes and report a two times enrichment in 77 % of the cases, five times enrichment in 37% of the cases, and 20 times enrichment in 11% of the cases [20]. This yields an average of 5.6 times enrichment on average and an average of 6.0 times enrichment in successful predictions only. *Oti et al.* test their method on 1,114 loci and report an average of 10 times enrichment [21]. *Aerts et al.* report that the correct gene is on average ranked as nr 13 out of 200 candidates yielding a 15.4 times enrichment. However this method allows for independent training and selection of a training set by experts in different disorders. Therefore the performance is expected to vary depending on the chosen training genes [22].

Bayesian Disease Gene Predictor

Introduction

In this paper we explore the importance of protein interaction networks for genetically based disorders, and construct a disease gene predictor that utilizes protein-protein interaction data. In particular, the following hypothesis forms the basis of our work: for a module of interacting proteins it is hypothesized that the phenotypic effects of disrupting any single protein in the module will be very similar no matter which individual protein that is disrupted. If this is generally true then protein interaction data can be used for discovering new disease-related proteins given a disease description and a set of candidate proteins. Specifically, the most likely candidates are those that interact with one or more proteins, which are involved in disorders that are similar to the one being investigated. For instance, imagine that some protein X is known to be involved in disorder #1 and we are now interested in finding the proteins involved in the phenotypically similar disorder #2. Imagine further that we have a list of four candidate disease-related proteins for disorder #2: A, B, C, and D. If protein B is found to interact with protein X, then B is a very good candidate for being involved in disorder #2.

In the following sections we describe how we have attempted to capture this “guilt-by-association” logic in the form of a probabilistic model that we then use to construct a predictor of disease-related proteins. Specifically, this predictor, which is based on Bayesian inference, takes as input a disease description and a set of candidate proteins, and produces as output a ranked list indicating for each individual candidate the probability that it is the disease-related one. It should be noted that a mathematical model of a biological system can be thought of as a stringently phrased hypothesis about that biological system. Where a hypothesis, such as the one outlined above, is expressed in qualitative terms, a mathematical model instead uses a set of parameters in an attempt to express exactly how the system works. Importantly, it is rarely the goal to construct a model that fully describes the biological reality. Such a model would be overly complicated and would contain so many parameters that it would be difficult to get good parameter estimates from real world data sets. Instead, the objective is typically to construct a model that is capable of accounting for the most important aspects of the system using a relatively limited number of parameters, and that can make reasonably good predictions about system behavior for unseen conditions.

In the present case the data consists of (1) protein-protein interaction data from a variety of sources, (2) information about genetic disorders from the OMIM database, including textual descriptions of phenotypic traits as well as information regarding disease-related proteins, (3) a critical interval, i.e., a chromosomal region containing a set of genes one of which is assumed to be involved in the investigated disorder. In addition to this “raw” information, we have devised methods for computing quantitative measures of how trustworthy any given protein-protein interaction is, and for the degree of overlap between different phenotype descriptions.

Terminology

Before we present the details of the model, it is necessary to introduce some terminology As mentioned above, the starting point is a phenotypic description of some disorder and a set of candidate proteins likely to be involved in that disorder (a “critical interval”). For each candidate protein we find the corresponding set of interacting proteins. For obvious reasons (see **Supplementary Fig.** 9 online) we sometimes refer to candidate proteins as “central proteins” while their putative interaction partners are dubbed “peripheral proteins”. Similarly we refer to the phenotypic description of the investigated disorder as the “central description”, while all other phenotypic descriptions are called “peripheral descriptions”. As described above, the trustworthiness of a putative interaction between a central and a peripheral protein is quantitated and assigned a score in the range zero to one; we will refer to this measure as the “interaction confidence score”. In the same way, the degree of overlap between a central and a peripheral description is also expressed as a number in the range zero to one; this measure will be referred to as the “description overlap”. For both the interaction confidence score and the description overlap we determine a threshold value below which we do not include the putative interaction partners and peripheral descriptions.

If a peripheral protein is mentioned in a peripheral description then this means that there is an indirect link between the central (candidate) protein and the investigated disease (the central phenotype description, see **Supplementary Fig.** 9 online. This is exactly the sort of signal we are looking for, and we take this as evidence for the candidate being the correct one. In this situation we say that there is a “connection”. The particular set of connections (protein-protein and description-description) for a given central protein is referred to as the “connection profile” for that candidate (see **Supplementary Fig.** 9 online). The set of connection profiles for all candidates in a critical region is termed the “full connection profile” or the “full profile” for that disorder.

The Model

The final goal of our modeling approach is to compute, for each possible candidate in a critical interval, the probability that this is the disease-related protein. Computation of these probabilities is based on the full connection profile (which constitutes all the available data) and is performed using Bayes’ theorem. Specifically, if we have N candidates then the probability of candidate number i being the disease protein is computed as follows:

$$P(\text{dis} = i \mid \text{full profile}) = \frac{P(\text{full profile} \mid \text{dis} = i) \times P(\text{dis} = i)}{\sum_{j=1}^{N} P(\text{full profile} \mid \text{dis} = j) \times P(\text{dis} = j)}$$

The term $P(dis = i \mid \text{full profile})$ is called the *posterior* probability, and represents our belief in candidate number i being the disease-related protein after seeing all the data. Conversely, $P(\text{dis} = i)$ is the so-called *prior* probability and represents our belief in candidate number i before seeing any evidence. For the purpose of this work we use a flat prior distribution and simply set this value to $\frac{1}{N}$ for all candidates. However, it should be noted that in principle previously available information regarding the candidates could be used to set the prior probabilities in a more informative manner. The term $P(\text{full profile} \mid \text{dis} = i)$ refers to the probability of obtaining the observed data if candidate number i was in fact the correct one. This is often called the *likelihood* (note that the likelihood is the probability of the data given the model, while the posterior is the probability of the model given the data). Specifying the model also means devising a way of computing the likelihood.

For the purpose of this model, we have chosen to divide candidate proteins into two distinct classes:

(1) Background, meaning the candidate is not involved in the investigated disorder. There might, however, still be spurious connections between the candidate and the phenotype description but we assume that, on average, they are less frequent and involve less significant description overlaps than authentic connections.

(2) Foreground, meaning the candidate is in fact involved in the investigated disorder. In this case there will often be a connection, presumably involving a fairly good description overlap. However, incomplete interaction and phenotype data and the genetic nature of some diseases will occasionally result in a lack of connectivity even in these cases.

The model used here is probabilistic. This means that all parameters are probabilities of various aspects of the biological system (*e.g.*, the probability that a candidate is connected, the probability that a phenotypic overlap has a magnitude in a certain range, *etc*., *etc*.). The essence of our hypothesis is, then, that these probabilities are different for the background and foreground cases, and this is what allows us to differentiate between disease-genes and non-disease genes.

Specifically, our model includes the following parameters:

(1) **Interaction probability:** $P(0 \text{ interactions})$, $P(\geq 1 \text{ interactions})$. These parameters, which sum to one, reflect the probability of a candidate protein having any (reported and trustworthy) interaction partners. Although there is a significant difference between the probabilities of having known interaction partners in the foreground and background (with a higher probability in the foreground case), we decided to use the global probability based on the full interaction database (after cleaning) in both cases. Since the proteins present in the foreground cases are better studied, the use of this parameter for discriminating could give rise to serious bias in the benchmarks since all proteins without interaction partners would be lower ranked than proteins having one or more interactions with proteins totally unrelated to the disease in question. As described below we assess the reliability of protein-protein interaction data and determine a threshold below which we do not believe in the interaction. The probabilities here are estimated after this cleaning step (see below).

(2) **Interaction confidence:** after applying the above-mentioned threshold we used a supervised discretization scheme for dividing the remaining range of interaction confidence ("IAC") values into a number of separate bins. For each bin we have a corresponding parameter reflecting the probability that a given interaction has a confidence value in that interval: $P(val_1 < \mathrm{IAC} \leq val_2)$, $P(val_2 < \mathrm{IAC} \leq val_3)$, …, $P(val_n < \mathrm{IAC} \leq 1)$. Again the probabilities sum to one, and there are background and foreground versions of this probability distribution.

(3) **Connection number:** P(0 connections)**,** P(1 connection)**,** P(2 connections)**, …,** P(≥ 7 connections)**.** These parameters, which sum to one, reflect the probability that a given peripheral protein has 0, 1, 2, …, or 7 or more connections to peripheral descriptions. The upper limit for the number of connections to consider individually (7 in the present example) was optimized using a genetic algorithm, and may be different for different subdivisions of the data in the cross-validation approach. Again, there are background and foreground versions of this probability distribution.

(4) **Description overlap:** in a manner similar to what we did for the interaction confidence values, we use a genetic algorithm to determine a threshold for description overlap ("DO") values that would give the optimal performance of our model and only include peripheral descriptions whose overlap with the central description is above this limit. Again we use a supervised discretization scheme

for dividing the remaining range of description overlap values into a number of bins, and for each of these bins we include a parameter reflecting the probability that a given connection involves a description overlap in a certain range: $P(val_1 < \mathrm{DO} \leq val_2)$, $P(val_2 < \mathrm{DO} \leq val_3)$, …, $P(val_n < \mathrm{DO} \leq 1)$. Again the probabilities sum to one, and again there are background and foreground versions of this probability distribution

Estimation of parameter values

All the parameters described above, except for the interaction probability, were estimated for both the background and foreground case, by determining the corresponding frequency in the training data. For instance, the probability of a peripheral protein not being connected to the central phenotype description, was determined by estimating the frequency of non-connected peripheral proteins in the training data. Due to the size of our data set these frequencies were found by sampling from the data (as opposed to counting all instances in the data). For the background case we used 10,000,000 samples, while 1,000,000 samples were used to estimate foreground probabilities. Specifically, background probabilities were estimated according to the following scheme:

(1) Pick random central protein.

(2) Determine interaction partners (if any), note the interaction confidence values.

(3) Repeat steps 1-2 a total of 10,000,000 times.

(4) Pick random peripheral protein.

(5) Pick random central phenotype description.

(6) Note number of connections between peripheral protein and central description.

(7) For each connection: note description overlap.

(8) Repeat steps 4-7 a total of 10,000,000 times.

Foreground probabilities were estimated according to the following scheme:

(1) Pick random disease causing protein (a central protein), and its corresponding phenotype description (a central description).

(2) Find interaction partners (if any). These are *peripheral* proteins.

(3) Select random peripheral protein among the set, note interaction confidence.

(4) Note number of connections between the selected peripheral protein and the central description.

(5) For each connection: note description overlap.

(6) Repeat steps 1-5 a total of 1,000,000 times.

Once all these sampled values were collected, a genetic algorithm was employed for determining (a) the interaction confidence threshold, (b) the maximal number of connections to consider, and (c) the description overlap threshold, that would give the optimal performance of our model. Subsequently, all interaction partners and peripheral descriptions below the thresholds were discarded, and an automatic scheme was used to find good discretizations of the probability distributions over interaction confidence and description overlap values so as to optimize the difference between the foreground and background. Finally, the probability parameters described above could be estimated by computing the corresponding frequencies. For instance, if we observe that a peripheral protein has no connection in 3,550,000 out of 10,000,000 samples, then we estimate the probability as: $P(0 \text{ connections}) = \frac{3{,}550{,}000}{10{,}000{,}000} = 0.3550$

Computing the probability of a connection profile for one central protein

Once all parameters have been estimated, they can be used to compute the probability of obtaining the actually observed connection profiles. This computation is done for both the background and foreground cases. We will describe the method by going over an example (see **Supplementary Fig. 10** online). For the purpose of this example we will assume that we have already gone through the sampling scheme described above and that we have estimated the following foreground parameter values (we will not go over computation of the background case which would be along the same lines, but using the background parameter values instead):

Interaction probability:

$$P(0 \text{ interactions}) = 0.35$$
$$P(\geq 1 \text{ interaction}) = 0.65$$

Interaction confidence ("IAC"):

$$P(0.1 < \text{IAC} \leq 0.25) = 0.55$$
$$P(0.25 < \text{IAC} \leq 0.73) = 0.43$$
$$P(0.73 < \text{IAC} \leq 1.0) = 0.02$$

Connection number:

$$
\begin{aligned}
&P(0 \text{ connections}) = 0.75\\
&P(1 \text{ connection}) = 0.18\\
&P(2 \text{ connections}) = 0.05\\
&P(\geq 3 \text{ connections}) = 0.02
\end{aligned}
$$

Description overlap ("DO"):

$$
\begin{aligned}
&P(0.1 < \text{DO} \leq 0.30) = 0.85\\
&P(0.30 < \text{DO} \leq 0.75) = 0.10\\
&P(0.75 < \text{DO} \leq 1.0) = 0.05
\end{aligned}
$$

Given these parameter values, the probability of the connection profile depicted in (see **Supplementary Fig.** 9 online) is computed as follows (note that for simplicity we omit multinomial coefficients from this computation – see explanation below):

$$
\begin{aligned}
&P(\text{profile } i \mid \text{foreground}) =\\
&\qquad P(\geq 1 \text{ interactions})\\
&\qquad \times P(0.1 < \text{IAC} \leq 0.25) \times P(0 \text{ connections})\\
&\qquad \times P(0.1 < \text{IAC} \leq 0.25) \times P(2 \text{ connections}) \times P(0.30 < \text{DO} \leq 0.75) \times P(0.1 < \text{DO} \leq 0.30)\\
&\qquad \times P(0.25 < \text{IAC} \leq 0.73) \times P(1 \text{ connection}) \times P(0.75 < \text{DO} \leq 1.0)\\
&\qquad = 0.65 \times 0.55 \times 0.75 \times 0.55 \times 0.05 \times 0.1 \times 0.85 \times 0.43 \times 0.18 \times 0.05\\
&\qquad = 2.4 \times 10^{-6}
\end{aligned}
$$

Here, the term on line one is included because candidate number i has any interaction partners in the first place. The terms on the second line are contributed by peripheral protein A, which has an interaction confidence score of 0.2 and zero connections. The terms on line three are contributed by peripheral protein B, which has an interaction confidence of 0.2 and two connections with description overlaps of 0.6 and 0.2 respectively. The terms on the final line are contributed by peripheral protein C, which has an interaction confidence of 0.5 and one connection with a description overlap of 0.9.

As mentioned, we have omitted multinomial coefficients from this computation, and therefore the above does not give the exact probability. However, this will have no influence on the final posterior probability, since these terms will appear in both the numerator and the denominator and they will therefore factor out of the computation.

If we had wanted to compute the exact probability, we should have included the following terms:

$$P(\text{profile } i \mid \text{foreground}) \;=\; 2.4 \times 10^{-6} \times \begin{pmatrix} 3 \\ 2\,1\,0 \end{pmatrix} \times \begin{pmatrix} 2 \\ 1\,1\,0 \end{pmatrix} = 2.4 \times 10^{-6} \times 3 \times 2 = 1.44 \times 10^{-5}$$

Here, the multinomial coefficient $\begin{pmatrix} 3 \\ 2\,1\,0 \end{pmatrix}$ derives from the computation of the probability that the central protein has three interaction partners. Specifically, it gives the number of ways in which you can have three interaction partners of which two have interaction confidence values in the lowest range, one has an interaction confidence value in the middle range, and zero have an interaction confidence value in the top range. Similarly, the multinomial coefficient $\begin{pmatrix} 2 \\ 1\,1\,0 \end{pmatrix}$ derives from the computation of the probability for peripheral protein B, which has two connections of which one has a description overlap in the lowest range, one has a description overlap in the middle range, and zero have a description overlap in the top range. There should be no multinomial coefficients for peripheral proteins A and C, since they have zero and one connection respectively, and there is only one possible permutation in each case.

Computing the probability of a full set of connection profiles

For each candidate protein we compute the probability of its connection profile, for both the background and foreground cases, along the lines explained in the example above. This gives us, for candidate number i the following probabilities: $P(\text{profile } i \,|\, \text{background})$ and $P(\text{profile } i \,|\, \text{foreground})$.

These values can now be used to compute the probability of the full connection profile given that protein number i is the disease-causing one (it is in the foreground while all other proteins belong to background):

$$\begin{aligned} P(\text{full profile} \,|\, \text{dis} = i) = & \prod_{j=1}^{i-1} P(\text{profile } j \,|\, \text{background}) \\ & \times P(\text{profile } i \,|\, \text{foreground}) \\ & \times \prod_{j=i+1}^{N} P(\text{profile } j \,|\, \text{background}) \end{aligned}$$

Computing the posterior probability of candidate number i being the disease-causing one.

If we have N candidates then the posterior probability of candidate number i being the disease protein can now be computed as follows:

$$P(\text{dis}=i\,|\,\text{full profile}) = \frac{P(\text{full profile}\,|\,\text{dis}=i)\times P(\text{dis}=i)}{\sum_{j=1}^{N} P(\text{full profile}\,|\,\text{dis}=j)\times P(\text{dis}=j)}$$

As mentioned, we use a flat prior distribution and simply set $P(\text{dis}=i)$ to $\frac{1}{N}$ for all candidates.

Genetic algorithm

A genetic algorithm was employed to determine (a) the interaction confidence threshold, (b) the maximal number of connections to consider, and (c) the description overlap threshold. The genetic algorithm operates on a 'gene' consisting of 11 bits where four bits represent the interaction confidence threshold in the range 0.10-0.72 in 0.02 intervals, three bits represent the maximal number of connections to consider in the range 0-7, and four bits represent the description overlap threshold in the range 0.10-0.72 in 0.02 intervals. The fitness of each gene is measured as the Pearson correlation between the posterior probability of each protein in the training set and the binary states of the proteins as correct (1) or false (0), so that when the correct proteins are given high posterior probabilities by the Bayesian predictor, the Pearson correlation will be higher and the genetic algorithm will favor that parameter combination. The genetic algorithm was stopped when no improvement (with a precision of 0.00001) over the best parameter combination was found in five successive generations. The genetic algorithm was run three times with different random seeds to increase the chance of finding the global and not a local maximum.

Supervised Discretization

Both interaction confidence and description overlap are measured using values in the range zero to one. For the purpose of the Bayesian predictor, however, we needed discrete probability distributions over the possible values. To achieve this we developed a method that automatically chooses the optimal discretization based on the distribution of values in a foreground and background set. To get the optimal performance of our predictor, the method aims to optimize a symmetrical version of the Kullback-Leibler (KL) distance between the background and foreground distributions. We computed the symmetric KL-distance using this expression:

$$KL = \sum_{i=1}^{k} p_i \log\left(\frac{p_i}{q_i}\right) + \sum_{i=1}^{k} q_i \log\left(\frac{q_i}{p_i}\right)$$

Here, k is the number of bins in the discrete distribution, p is the foreground probability of being in that bin, while q is the corresponding background probability. Specifically, our algorithm first places a number of equally spaced grid-points over the range of possible values. In the present analysis we used 1000 divisions. These grid-points are the values that we consider as possible bin-borders. The algorithm then progressively adds bin borders by iteratively scanning all possible positions and in each iteration choosing the position that gives the highest increase in KL-distance compared to last iteration. Addition of bin-borders is stopped when the improvement is less than 1%.

Five fold cross-validation and prediction

The benchmark data was partitioned into five parts and the parameters of the Bayesian predictor were optimized on four parts and then validated on the last part. This was repeated five times so that in the end, there were five optimal models, each from a different but overlapping training set, and five validations on five different partitions. The full benchmark figures were derived by summing the five validations. This way, the validation data is independent of the training data, but we can still make use of the entire benchmark set to get reliable statistics.

The predictions presented are the average results from predicting with each of the five optimal models.

References

1. Aronson, A.R. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc AMIA Symp*, 17-21 (2001).
2. Bodenreider, O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res* **32**, D267-270 (2004).
3. Divita, G., Tse, T. & Roth, L. Failure analysis of MetaMap Transfer (MMTx). *Medinfo* **11**, 763-767 (2004).
4. Pratt, W. & Yetisgen-Yildiz, M. A study of biomedical concept identification: MetaMap vs. people. *AMIA Annu Symp Proc*, 529-533 (2003).
5. Chapman, W.W., Fiszman, M., Dowling, J.N., Chapman, B.E. & Rindflesch, T.C. Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap. *Medinfo* **11**, 487-491 (2004).
6. Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F. & Buchanan, B.G. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform* **34**, 301-310 (2001).

7. Polavarapu, N. et al. Investigation into biomedical literature classification using support vector machines. *Proc IEEE Comput Syst Bioinform Conf*, 366-374 (2005).
8. Al-Shahrour, F. et al. BABELOMICS: a systems biology perspective in the functional annotation of genome-scale experiments. *Nucleic Acids Res* **34**, W472-476 (2006).
9. Perez-Iratxeta, C., Bork, P. & Andrade, M.A. Association of genes to genetically inherited diseases using data mining. *Nat Genet* **31**, 316-319 (2002).
10. Kanehisa, M. et al. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res* **34**, D354-357 (2006).
11. Joshi-Tope, G. et al. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res* **33**, D428-432 (2005).
12. de Lichtenberg, U., Jensen, L.J., Brunak, S. & Bork, P. Dynamic complex formation during the yeast cell cycle. *Science* **307**, 724-727 (2005).
13. von Mering, C. et al. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* **417**, 399-403 (2002).
14. Perez-Iratxeta, C., Wjst, M., Bork, P. & Andrade, M.A. G2D: a tool for mining genes associated with disease. *BMC Genet* **6**, 45 (2005).
15. Venter, J.C. et al. The sequence of the human genome. *Science* **291**, 1304-1351 (2001).
16. van Driel, M.A., Cuelenaere, K., Kemmeren, P.P., Leunissen, J.A. & Brunner, H.G. A new web-based data mining tool for the identification of candidate genes for human genetic disorders. *Eur J Hum Genet* **11**, 57-63 (2003).
17. Turner, F.S., Clutterbuck, D.R. & Semple, C.A. POCUS: mining genomic sequence annotation to predict disease genes. *Genome Biol* **4**, R75 (2003).
18. Franke, L. et al. Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am J Hum Genet* **78**, 1011-1025 (2006).
19. Freudenberg, J. & Propping, P. A similarity-based method for genome-wide prediction of disease-relevant human genes. *Bioinformatics* **18 Suppl 2**, S110-115 (2002).
20. Adie, E.A., Adams, R.R., Evans, K.L., Porteous, D.J. & Pickard, B.S. Speeding disease gene discovery by sequence based candidate prioritization. *BMC Bioinformatics* **6**, 55 (2005).
21. Oti, M., Snel, B., Huynen, M.A. & Brunner, H.G. Predicting disease genes using protein-protein interactions. *J Med Genet* (2006).
22. Aerts, S. et al. Gene prioritization through genomic data fusion. *Nat Biotechnol* **24**, 537-544 (2006).