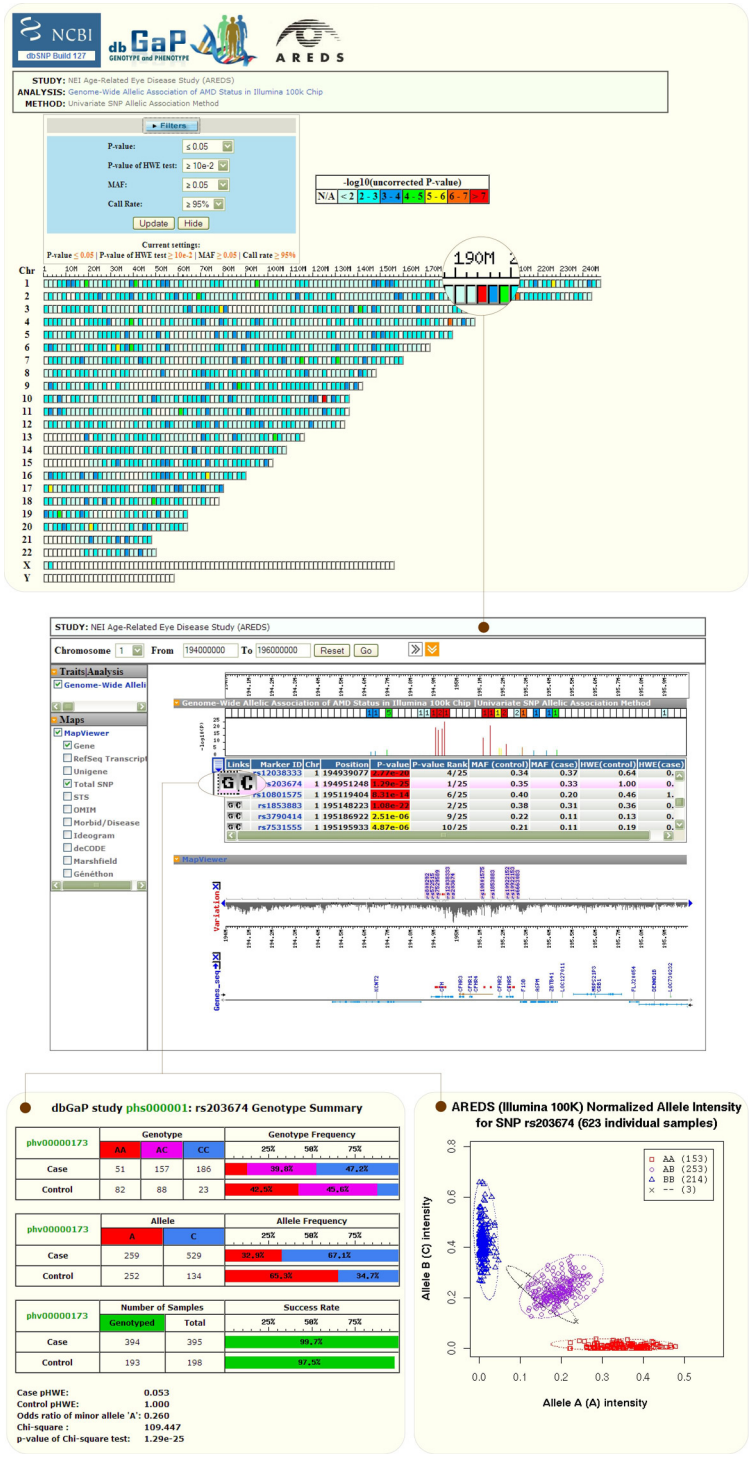

Supplementary Figure 1: The **dbGaP association browser** presents a heat map representation of each chromosome, where 2 megabase bins are color-coded to indicate the most significantly associated markers in the bin. Adjusting parameters such as P-value, LOD score, Hardy-Weinberg equilibrium, minor allele frequency, and call rate allows users to filter results. Clicking on a particular bin zooms into a higher-resolution view where the number of interrogated markers is indicated within each bin. Mousing over any numbered bin scrolls through a table of statistical information for the relevant genotyped markers. Various genomic tracks can be added to the browser, including genes, RefSeq RNAs, OMIM allelic variants, dbSNP variations, and several genetic marker maps that demonstrate the genomic context of a specific marker of interest. Clicking on the "G" and "C" buttons to the left of any row of the table will bring up genotype statistical data and cluster plot images, respectively.

The NCBI dbGaP database of genotypes and phenotypes

Supplementary Note

Extensions to the NLM Archiving and Interchange DTD

The NLM journal DTD was extended in three ways to meet the needs of the dbGaP documents. First, the model for the "front matter" (usually bibliographic information when dealing with journal articles or books) was altered to allow dbGaP-specific metadata, including information about the document's parent study. Second, we added the ability to reference phenotype variables from relevant pieces of text (for example, to link a paragraph describing how a measurement was taken or an item in a list describing the piece of equipment used to make a measurement to the variable that contains the measurement). Finally, we created a model for markup of "questions" within specific data collection forms.

Because the documents are encoded using XML they must go through a rendering step to produce versions that are viewable and interactive (HTML for a web-browser) or printable (PDF). Both versions are produced using components of XSL (the eXtensible Stylesheet Language). The HTML version of a document shows the original text of the document with the links built from objects in the documents to the variable report pages, as described above. The PDF versions of the document are produced for those who want a printable copy of a document (1).

The first two modifications were minor and straightforward, but the third, determining a model for questions, proved to be harder than anticipated. The reasons for this are two-fold. First, questions can have both complex structures and complex relationships to

other questions or other items on the form. Second, questions are often laid out with the purpose of fitting them within a page structure; this can confound the process of determining the underlying intellectual structure of questions that have been combined to save space during the page layout process.

The premise of question modeling is that there is a certain "atomic" unit that is common across all questions: the variable (which can also be thought of as the value input, blank line, or checkbox on a questionnaire). The variable is the appropriate point of focus of the modeling because any input on a form could, potentially, become a value in our database. Therefore we started with the variable and have built the remaining question structure around it.

After looking at a number of different questionnaires, and working through a number of intermediate designs, we have come up with a series of assumptions that summarize our XML structure for questions:

1. There is a basic unit, the "variable," which corresponds to any place on a form that allows input.
2. A variable can contain different types of text (questions, instructions, scripts, description) and may either have an area for free-form input or a list of items from which to select.

3. A variable has a particular type: e.g. input – allows user-entered values; select1 – allows the user to choose one from a list of choices; and select – allows a user to choose more than one from a list of choices.
4. If a variable is of the type select1 or select, then the items that can be chosen must be defined.
5. If a variable is of the type input then the 'input-box' size and location must be defined.
6. Variables can be grouped together into "variable-groups."
7. Variable groups may contain other variables and text, such as questions, instructions, scripts, descriptions.

References

1. Kay, Michael. XSLT Programmer's Reference: 2nd Edition. Wrox Press Ltd., Birmingham. (2001)