Supplementary Note 1: MEM, Gene recommender, meta-data set correlation algorithms and implementations

Gene recommender

Gene recommender[1] is an algorithm that can retrieve relevant experiments based on the query, and use these experiments to retrieve query co-regulated genes. It performs an experiment (or sample)-level weighting rather than a data set-level weighting. First it merges samples from all data sets to form a meta matrix Y_{ij} (i = gene, j = experiment). Given the query genes, the weighting algorithm is based on a number of criteria such as the gene expression of the query genes, and the expression variance of the query in each experiment. The original matrix Y_{ij} of n genes by p experiments (or samples) is transformed to ranks Y_{ij}', where

$$Y_{ij}' = \frac{R_{ij} - (p_i + 1)/2}{p_i/2}$$

R_{ij} is the rank of i among Y_{ij} for $j = 1 \ldots p$, and p_i is the number of experiments containing gene i. The experiment scoring is calculated as:

$$Z_j = \sqrt{k_j}\,\frac{avg(Y_{Qj})}{\sqrt{V_{Qj} + \frac{1}{3p^2}}}$$

where $avg(Y_{Qj})$ is the average expression (Y_{ij}') over query genes Q in j, V_{Qj} is the variance of the query in j, k_j is the number of genes in j. This scoring prefers experiments with a tight clustering of the query genes with high expression, low variance. In order to use the experiment scoring to return query-coregulated genes, a threshold ε defines the number of relevant experiments (with top scores), so the final score of gene i is calculated as: S_i = the mean of ($avg(Y_{Qj}) \times Y_{ij}'$) over all relevant experiments j. The parameter ε is set at 0.05 (or 5% of the total experiments).

MEM

The MEM algorithm[2,3] assumes that the query is a single gene q. For each data set j, it first transforms the correlations containing the query gene into ranks, so that each gene has a rank n that represents the n-th correlated gene to the query. Ranks are normalized to [0, 1] by dividing each rank by the maximal rank in each data set. Then, the ranks are transformed so that for each gene g_i, we generate a rank vector $r(q, g_i) = [\, r_1^{\,i}, \ldots, r_m^{\,i} \,]$, where $r_j^{\,i}$ is the position of g_i in the query on data set j, and m is the number of data sets. MEM assumes a null hypothesis where in a model rank-list the genes are randomly permuted, and $r(q, g_i)$ contains uniformly distributed ranks. It reorders $r(q, g_i)$ in order to obtain a vector of order statistics, $r_{(1)}^{\,i}, \ldots, r_{(m)}^{\,i}$ where $r_{(1)}^{\,i}$ is the smallest, and $r_{(m)}^{\,i}$ is the largest value in $r(q, g_i)$. Assuming null hypothesis, it then calculates the

probability from binomial distribution, that the order statistic $r'^{i}_{(k)}$ is smaller or equal to $r^{i}_{(k)}$, where $r'^{i}_{(k)} < r^{i}_{(k)}$ is generated by null model:

$$b(k) = \sum_{j=k}^{m} \binom{m}{j} (r^{i}_{(k)})^{j} (1 - r^{i}_{(k)})^{m-j}$$

The score of each gene is then the *p*-value:

$p(g_i) = \min b(k)$ for each k in the range $[0, m]$.

Intuitively, if the rank vector of a gene contains a large number of small ranks (which means that the gene is consistently correlated to the query in large number of data sets), the distribution of $r(q, g_i)$ will be heavily biased towards the small values and different from the uniform distribution.

Meta-data set correlation

Meta-data set correlation is a simple approach for combining data sets. First, data sets are concatenated into one matrix, called meta-data set. Then, genes were ranked according to the average of Pearson correlations to the query genes in the meta-data set. As data sets may include different sets of genes, we calculated correlation only for pairs of genes where each gene in the pair is present in at least 50% of the data sets, yielding a reasonable set of 17,689 genes being ranked. Where the data set coverage of two genes differs, we chose the entire set of samples with values present for both genes in the matrix for computing their correlation.

Supplementary Note 2: Hedgehog (Hh) query – detailed analysis of the retrieved genes

Below we describe additional details of the top retrieved genes for the Hh pathway example described in the manuscript. The known Hh pathway members *SMO* (rank 1), *HHIP* (rank 6), *BOC* (rank 7), and *PTCH2* (rank 9) are all among the top 10 SEEK-retrieved genes, and *KIF7* is ranked 22 – all in the first view immediately available to the biologist running SEEK. Other Hh-associated genes are also retrieved with top ranks. Multiple studies show that the TGF-beta pathway genes *RGMA* (rank 2), *LTBP4* (rank 8) are significantly co-induced with *GLI1* and *GLI2* in recurrent tumors [4,5]. The ortholog of protocadherin 18 (*PCDH18*, rank 3) interacts with *DAB1*, which functions in concert with the Hh pathway to control retina development [6]. *FZD7* (rank 4) is an important receptor in the Wnt pathway that extensively cross-talks with the Hh pathway [7]. The Notch signaling protein *HEYL* (rank 15) regulates *HES1*, which directly modulates *Gli1* expression and Hh signaling [8,9]. *HHIP-AS1* (rank 20) encodes the antisense RNA of the Hh interacting protein *HHIP*, which is a vertebrate-specific inhibitor of Hh signaling [10]. Many others genes among the top 25 retrieved – *KIF26A* (rank 10),

CRMP1 (rank 11), *CCDC8* (rank 13), *SLC26A10* (rank 14), *RUNX1T1* (rank 17), *MRAP2* (rank 18), *GPR124* (rank 19), and *PCYT1B* (rank 21) – have literature evidence for either regulatory interactions (direct or indirect), or pathway-level cross-talk with members of the Hh signaling pathway.

Supplementary Note 3: Web interface

SEEK has been implemented as an interactive, easy-to-use website that allows biologists to perform queries, view expression patterns of the retrieved co-expressed genes, and perform visualization-based analyses. The goal of the SEEK web interface is to offer a Google-like engine for expression and co-expression retrieval, enabling biomedical researchers to fully utilize the thousands of expression data sets for accomplishing their analyses with a focused yet flexible and interactive web-based system. The web interface offers three flexible modes of visualizing users' results: **expression view**, **co-expression view**, and **condition-specific view**.

Expression view is the first view that the user sees upon completion of their search. **Fig. 2a** (main text) shows an example. The top 100 co-expressed genes are shown for the query *GLI1*, *GLI2*, and *PTCH1* (the user can easily see other lower-ranked genes of interest). The data sets are displayed in order of relevance, allowing the user to focus on those most related to their area of interest based on query co-expressions. In this view, expression levels for each gene are displayed, and a score is provided for each gene that conveys its level of normalized, hubbiness-corrected, and weighted co-expression to the query. A weight is provided for each data set, which offers a measure of the co-expression between the query genes in that data set as an indication of data set relevance. Each page juxtaposes multiple data sets' expression matrices to allow quick comparison and navigation. Within each data set's expression matrix, SEEK hierarchically clusters the conditions in the data set according to expression of the retrieved genes that are shown to the user. This clustering provides a quick visualization for identifying up- and down-regulation pertinent to the query genes.

Condition-specific view (**Fig 1 of this note**) is activated by clicking on the expression pattern of a gene in a particular data set. This view allows users to associate co-expressed genes with the meta-information (or measured outcome) attached to the data set, such as disease state, cell type, cell line, drug treatment, and patient characteristics. Users can choose among the data set's available attributes, and re-cluster the selected data set based on an attribute of interest. For example, by selecting the attribute "anatomical sites" for a Hedgehog related data set, and viewing the Hedgehog genes in the context of anatomical sites, they can observe that Hedgehog signaling is abundant in testis and pancreas, but not in lymph node tissues (**Fig 1 of this note**). Thus, potential associations to various measured outcomes can be readily uncovered post-search through the condition-specific view.

SEEK's **co-expression view** (**Fig 2 of this note**) provides a "bird's-eye" view of the co-expression landscape across up to 50 data sets at a time. Users can readily identify the data sets that are most relevant for the query, based on the co-expression of each retrieved gene to the query visualized as single columns. Users can readily assess the contribution of each data set. This view also serves to visually analyze the query coherence (**Fig 2 of this note,** top heat-map), helping users in constructing a coherent query gene set, which in turn guides SEEK in producing more relevant results.

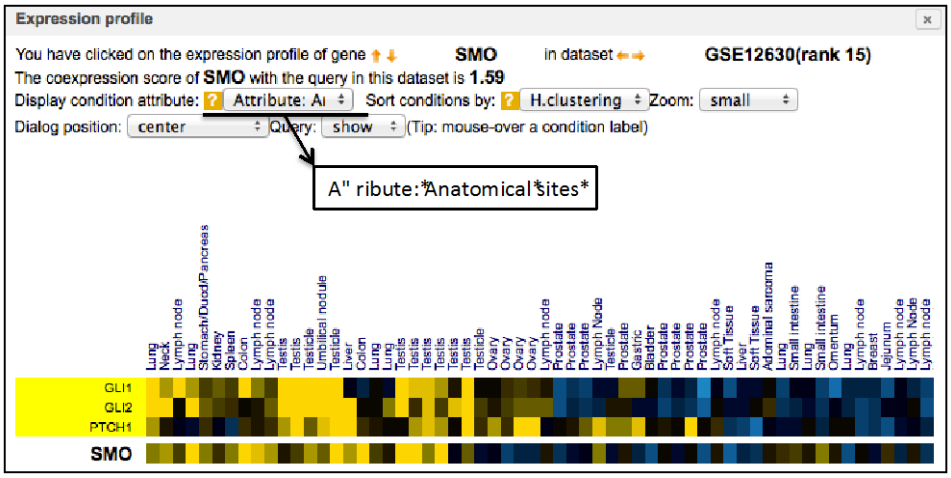

Fig 1 of Supplementary Note 3: Condition-specific view. This zoom-in view is generated by clicking on the row corresponding to *SMO* and GSE12630 data set in the result page of the *GLI1, GLI2, PTCH1* query.

Fig 2 of Supplementary Note 3: Co-expression view. Top heat map: query coherence, measured by the degree to which each query gene correlates with the rest of the query across the top 50 data sets. Each column represents a data set. Any “outlier” genes can thus be identified and subsequently removed from the query.

Downstream analyses – Refine Search

An important feature of SEEK is providing the user with flexible search refinement options (**Fig 3 of this note**). Although the SEEK algorithm enables robust search over the whole expression compendium, there are cases when users intend to restrict the search domain to a subset of data sets, for instance, when they desire a tissue- or disease-oriented co-expression analysis, or when the user encounters a situation when her query is too small or heterogeneous, and the intended context is not readily identifiable from the query alone. The **Refine Search** function provides users with several ways of refining the search analysis. Users may narrow their results down by:

1) Limiting to a tissue or disease of interest. There are currently hundreds of selectable tissues, cell-types, and diseases defined by UMLS and BRENDA keywords.

2) Limiting the search to only cancer or non-cancer data sets. The cancer data compendium includes primary tumors, metastasized tumors, and cancer cell lines. The non-cancer compendium includes diverse non-cancer samples, including stem cells, muscle and adipose cells, neurodegenerative, immune and infectious disease samples, epithelial and endothelial cell types, and blood cell types in non-cancerous diseases.

3) Limiting to multi-tissue profiling data sets only. This group of 13 data sets is useful for checking the expression of gene(s) across normal tissues, cell lines, cell types, and diseased tissues from various organs.

4) Limiting to primary tumor data sets only. Users can select the 224 TCGA RNASeq data sets as well as around 200 data sets from independent research studies that profile single-tissue tumors in each data set.

SEEK provides users with an easy-to-use and easily searchable data set-type selector (**Fig 3 of this note**). After a category has been selected, SEEK will perform the data set prioritization and co-expressed gene search within the chosen category of data sets only.

Fig 3 of Supplementary Note 3: Available options within the Refine Search window. The second column lists the number of data sets in each data set category.

Supplementary Note 4: Batch effect evaluation

SEEK uses the data set weighting algorithm to systematically address the challenge of the possible batch effects that exist in certain data sets in the compendium. To evaluate SEEK's effectiveness, we identified low quality data sets with severe batch effects in the compendium based on the variation in the samples' expression value distribution within each data set. Specifically, for each data set d, with d_n samples, we calculate the standard deviation σ_d

$$\sigma_d = std(IQR_{d_1}, IQR_{d_2}, ..., IQR_{d_n})$$

where d_1, …, d_n are the samples in data set d, and IQR is the interquartile range for the expression values in a sample d_x. A relatively high σ_d signifies a technical bias or a shift in the median and IQR of the gene expression values in that array,

which is generally caused by batch effects. We then examined the 100 datasets with the highest σ_d (and thus most severe batch effects) in the compendium to see where they are ranked in full dataset prioritization (~4,500 data sets) for 121 diverse GO-slim queries (GO-slim[11] provides a curated set of diverse, high-level GO terms that are nonetheless specific enough for experimental evaluation and span the full set of GO biological processes). There was indeed a negative enrichment of the 100 data sets in full data prioritizations across 121 GO slim queries (**Supplementary Fig. 7**), indicating that data sets with substantial batch effects are systematically ranked lower than randomly selected data sets, and thus effectively down-weighed in the SEEK search process. In fact, a high proportion (84%) of these 100 low quality data sets have a non-significant data set weight assigned by SEEK (at *P* more than the 0.001 significance cut off) (**Supplementary Fig. 7**, source data), thus demonstrating the effectiveness of the SEEK data set weighting algorithm in automatically handling low-quality data sets.

Supplementary Note 5: Raw data processing and normalization

Each microarray platform had a relatively accepted pipeline for processing its data sets. Briefly, for Affymetrix platforms, we normalized each array using Robust Multi-array Average (RMA) [12], which ensures that the distribution of expression values per array is the same within each data set. We note that SEEK also performs similarly well for data sets processed with other techniques, such as MAS5. For Agilent, there are two types of arrays: single-channel and dual-channel arrays. Dual channel arrays are designed for measuring fold-change between case and control conditions. In dual-channel arrays, individual arrays are normalized by loess normalization (Zahurak et al[13]). Next, we calculated the log-2 Cy3 over Cy5 fold change and applied between-array normalization, which is essential in two color array analysis, as it normalizes channel intensities and log-ratios to be comparable across arrays. Single-channel arrays were normalized by within-data set quantile normalizations. The above analysis was done using the Bioconductor R and limma package[14] following the guide in Chapter 6 in the limma manual[15]. For Illumina BeadChip platforms, we limit to the set of data sets that have no missing probe measurements, termed "unnormalized" raw data obtained from the Gene Expression Omnibus. We normalized the arrays using quantile normalization [16] as this is the recommended approach in the study Ritchie et al[17]. This use of consistent processing pipeline across all data sets within a given platform helps remove systematic differences between data sets[18].

For data sets from the RNA sequencing platforms, we obtained 5,085 RNASeq samples that were pre-processed level-3 data from TCGA [19]. Discussion of the processing is found in [20,21]. On a high level, for these TCGA samples, we use normalized counts, which are the raw counts divided by the 75th percentile of

each column multiplied by 1000 (known as the upper-quartile normalization[22]). TCGA samples have been split into 224 data sets according to unique 'disease type, sample source' pairs. We also extracted 54 RNASeq data sets from GEO that have been processed by submitters of the data sets. These data sets have been published in their associated studies (**Supplementary Data 4**), where the processing of each data set is discussed. We use results summarized in raw counts format, and we further performed upper-quartile normalization on counts data to be consistent with the TCGA samples. Final measurements are normalized by $\log_2$(1+normalized_counts).

The gene expression data sets normalized using the abovementioned procedure are publicly available for download on the SEEK website.

Supplementary Note 6: Search algorithm pseudocode

The SEEK search algorithm is a general search algorithm that works to integrate co-expressions from diverse data sets across platforms, and tackles the problem of incomplete gene ranking that arises from the diverse gene coverage across data sets. The algorithm is described in the following pseudocode:

Input: query genes (Q), genes in genome (G), data sets (D), correlation z-scores for pairs containing Q across data sets ($z_d(g,q), g \in G, q \in Q, d \in D$).

Variables: $M_{g,d}$, matrix of gene scores across D; $count_g$, vector of coverage of genes; w_d, vector of data set weight; F_g, vector of final gene scores.

Constants: α, β, θ.

Begin:

Compute $\tilde{z}_d(g,q)$ for each g, q, and d, as described in **Eq. 1** (**Methods**).
//Hubbiness control

Initialize $M_{g,d} = 0$ for all g,d; $count_g = 0$ for all g; $w_d = 0$ for all d.

For each data set d: //Data set weighting

Let V = set of genes that d contains

If $|V| < \boldsymbol{\alpha}$ or $|V \cap Q| < \boldsymbol{\beta}$: //not enough genes, or query genes present

continue

Compute w_d as described in **Eq. 2** (**Methods**).

$M_{g,d} = \sum_{q \in V \cap Q} \tilde{z}_d(g,q)/|V \cap Q|$, for each gene $g \in V$

$count_g = count_g + 1$, for each gene $g \in V$

End for

For each g in G: //Gene scoring

If $count_g > \boldsymbol{\theta}$: //sufficient data set coverage for g

Let U = set of data sets that contain g

$$F_g = \frac{1}{\sum_{d \in U} w_d} \sum_{d \in U} w_d \times M_{g,d}$$

Else:

$$F_g = -inf$$

End if

End for

Sort F based on decreasing score, generate gene ranking (R_G)

Sort w based on decreasing weight, generate data set ranking (R_D)

Return R_G and R_D

End

The three thresholds α, β, θ are designed to maximize data utilization while keeping in check the biases introduced by incomplete data. α is the minimum number of genes required to be present in a data set. β is the minimum number of query genes that have to be measured in a data set, and θ is the minimum number of data sets required to contain a gene to include the gene in search ranking. Based on our experience, the following thresholds provide robust performance for a variety of queries and for large compendia with diverse data sets: $\alpha = 10{,}000$, $\beta = 2$, $\theta = 0.5|D_w|$ where $D_w \subseteq D$ is the set of weighted data sets for the given query.

Supplementary-only references

1. Owen, A. B., Stuart, J., Mach, K., Villeneuve, A. M. & Kim, S. A Gene Recommender Algorithm to Identify Coexpressed Genes in C. elegans. *Genome Res.* **13,** 1828–1837 (2003).

2. Adler, P. *et al.* Mining for coexpression across hundreds of datasets using novel rank aggregation and visualization methods. *Genome Biol.* **10,** R139 (2009).

3. Kolde, R., Laur, S., Adler, P. & Vilo, J. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics* **28,** 573–80 (2012).

4. Kwon, M. & Shin, Y. Regulation of ovarian cancer stem cells or tumor-initiating cells. *Int. J. Mol. Sci.* **14,** 6624–48 (2013).

5. Steg, A., Bevis, K. & Katre, A. Stem cell pathways contribute to clinical chemoresistance in ovarian cancer. *Clin. Cancer Res.* **18,** 869–81 (2012).

6. Homayouni, R., Rice, D. & Curran, T. Disabled-1 interacts with a novel developmentally regulated protocadherin. *Biochem. Biophys. Res. Commun.* **289,** 539–47 (2001).

7. Katoh, Y. & Katoh, M. WNT antagonist, SFRP1, is Hedgehog signaling target. *Int. J. Mol. Med.* **17,** 171–5 (2006).

8. Schreck, K. C. *et al.* The Notch target Hes1 directly modulates Gli1 expression and Hedgehog signaling: a potential mechanism of therapeutic resistance. *Clin. cancer Res.* **16,** 6060–70 (2010).

9. Jalali, A., Bassuk, A. & Kan, L. HeyL promotes neuronal differentiation of neural progenitor cells. *J. Neurosci. Res.* **89,** 299–309 (2011).

10. Chuang, P. & McMahon, A. Vertebrate Hedgehog signalling modulated by induction of a Hedgehog-binding protein. *Nature* **397,** 617–21 (1999).

11. Myers, C., Barrett, D. & Hibbs, M. Finding function: evaluation methods for functional genomic data. *BMC Genomics* **7,** 187 (2006).

12. Irizarry, R. a *et al.* Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4,** 249–64 (2003).

13. Zahurak, M. *et al.* Pre-processing Agilent microarray data. *BMC Bioinformatics* **8,** 142 (2007).

14. Smyth, G. Limma: linear models for microarray data. in *Bioinforma. Comput. Biol. Solut. using R Bioconductor* (Gentleman, R., Carey, V., Dudoit, S., Irizarry, R. & Huber, W.) 397–420 (Springer, 2005).

15. Smyth, G. K. *et al.* limma: Linear Model for Micrarray Data User's Guide. at <http://www.bioconductor.org/packages/release/bioc/vignettes/limma/inst/doc/usersguide.pdf>

16. Bolstad, B. M., Irizarry, R. a, Astrand, M. & Speed, T. P. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19,** 185–93 (2003).

17. Ritchie, M. E. *et al.* A comparison of background correction methods for two-colour microarrays. *Bioinformatics* **23,** 2700–7 (2007).

18. Ramasamy, A., Mondry, A., Holmes, C. C. & Altman, D. G. Key issues in conducting a meta-analysis of gene expression microarray datasets. *PLoS Med.* **5,** e184 (2008).

19. Cancer, T. & Atlas, G. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* **455,** 1061–8 (2008).

20. National Cancer Institute. RNASeq Version 2. at <https://wiki.nci.nih.gov/display/TCGA/RNASeq+Version+2>

21. Mose, L. & Parker, J. V2_MapSpliceRSEM: UNC V2 RNA-Seq Workflow - MapSplice genome alignment and RSEM estimation of GAF 2.1. at <https://confluence.broadinstitute.org/download/attachments/29790363/DESCRIPTION.txt>

22. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* **11,** 94 (2010).