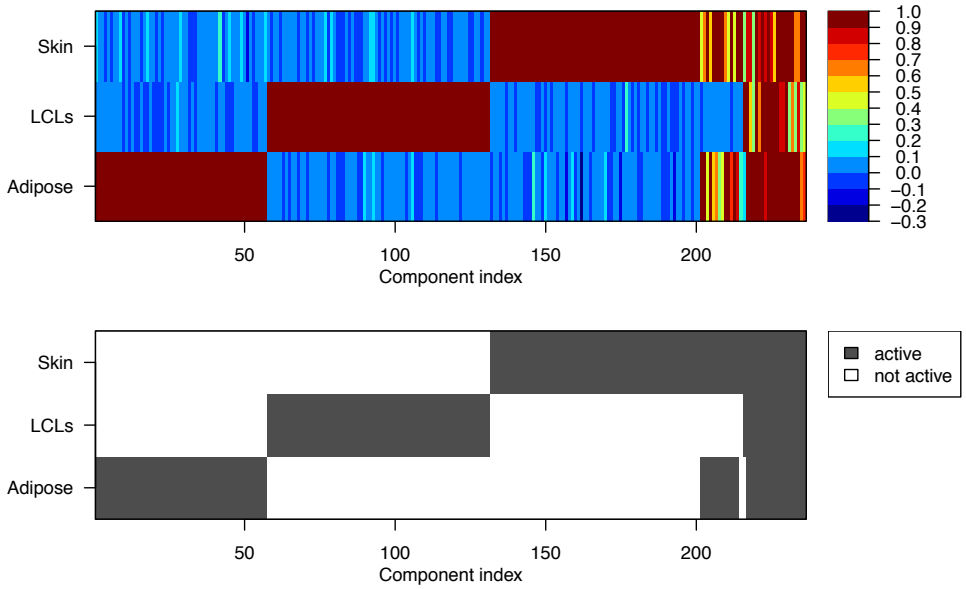
Supplementary Information for "Tensor decomposition for multi-tissue gene expression experiments"

Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight,
Mark I McCarthy, Kerrin Small, Jonathan Marchini

June 12, 2016

Contents

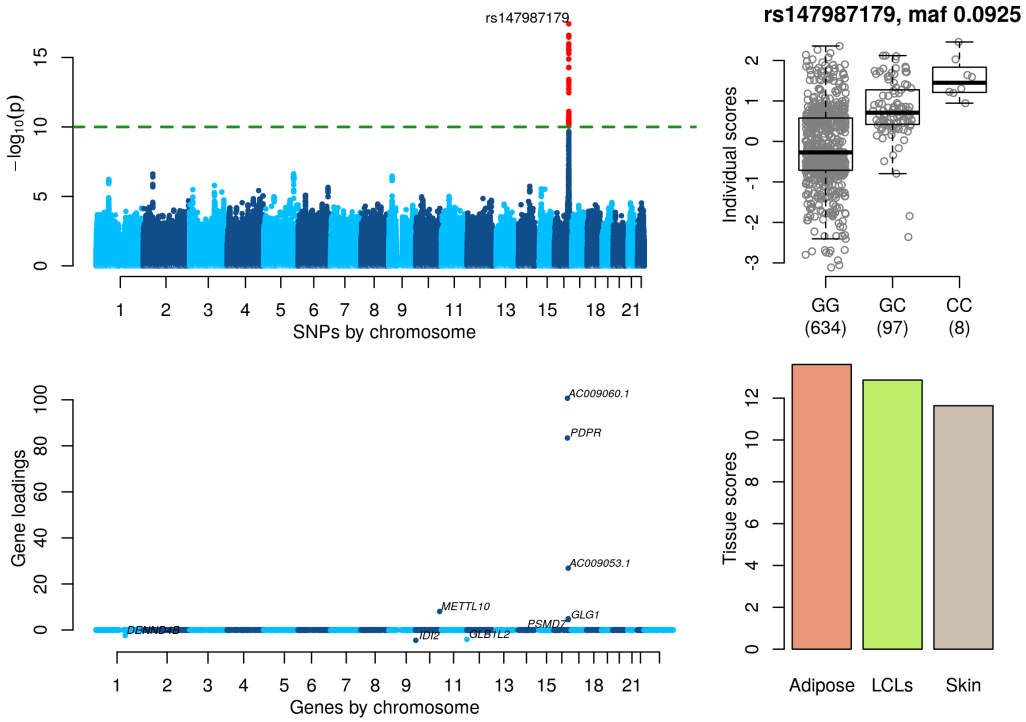
Supplementary Figures



Supplementary Figure 1

Tissue scores matrix for 236 robustly identified components.

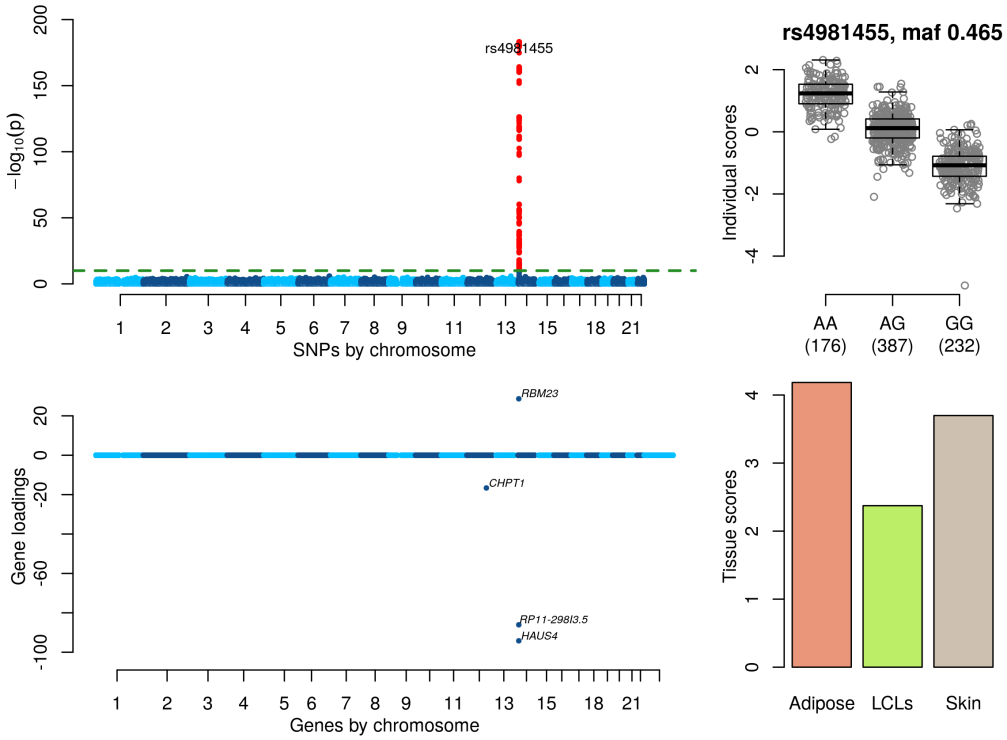
(Top) Each column shows the tissue scores for a component, scaled so that the largest score equals 1. Columns have been arranged to group components with similar tissue patterns. (Bottom) Binary representation of scaled tissue scores (obtained by thresholding scores at 0.5) to highlight the tissue specificity of the components.



Supplementary Figure 2

Robust component describing a *cis* effect.

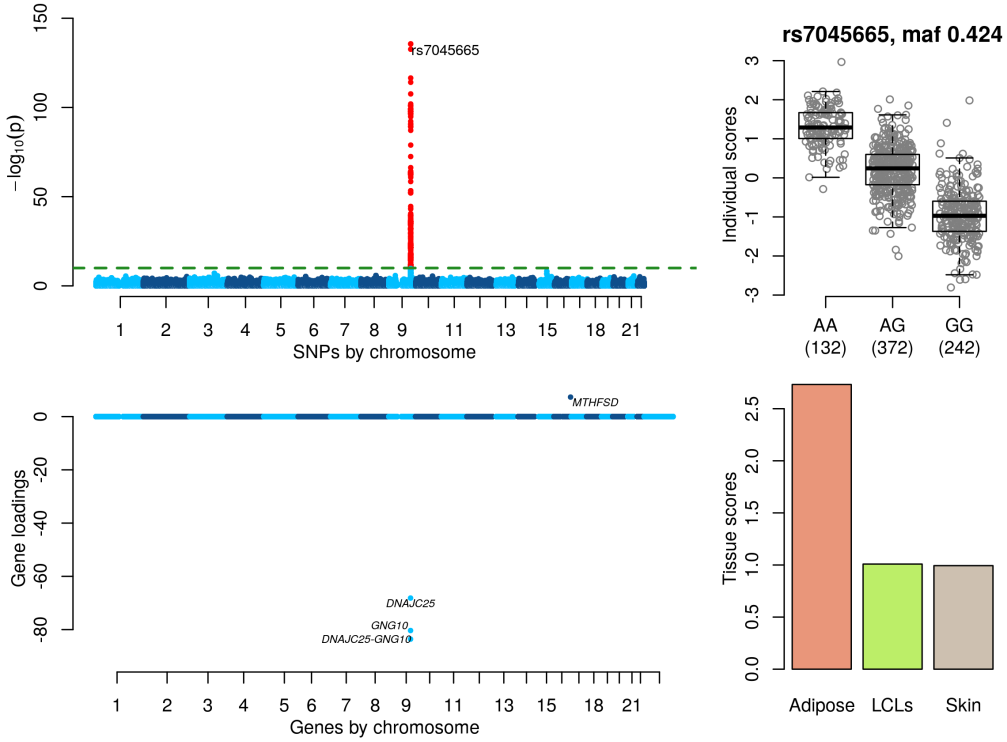
(Top left) GWAS with the component's individual scores vector as a phenotype. (Top right) Boxplot of individual scores stratified by genotype at the lead GWAS SNP. Boxplots show the median, upper and lower quartiles, with whiskers extending to either 1.5 times the inter quartile range (IQR), or to the most extreme data point if this lies within 1.5 times IQR. (Bottom left) Gene loadings for the component. Only gene loadings with a PIP>0.5 are shown. (Bottom right) Tissue scores vector for the component shown as a barplot.



Supplementary Figure 3

Robust component describing a *cis* effect.

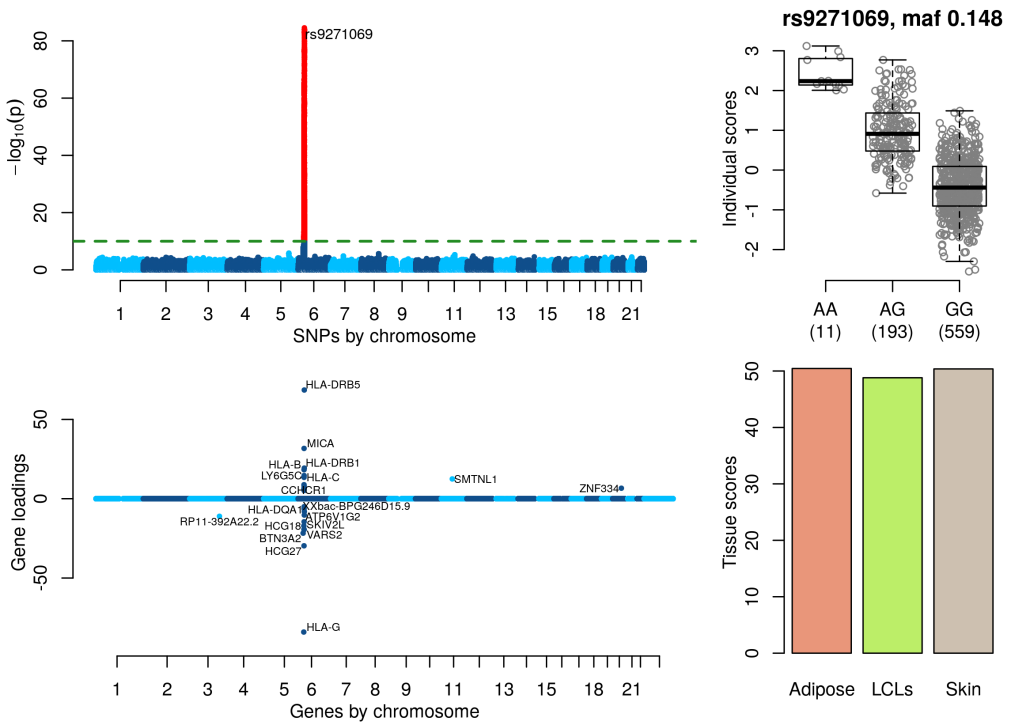
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 4

Robust component describing a *cis* effect.

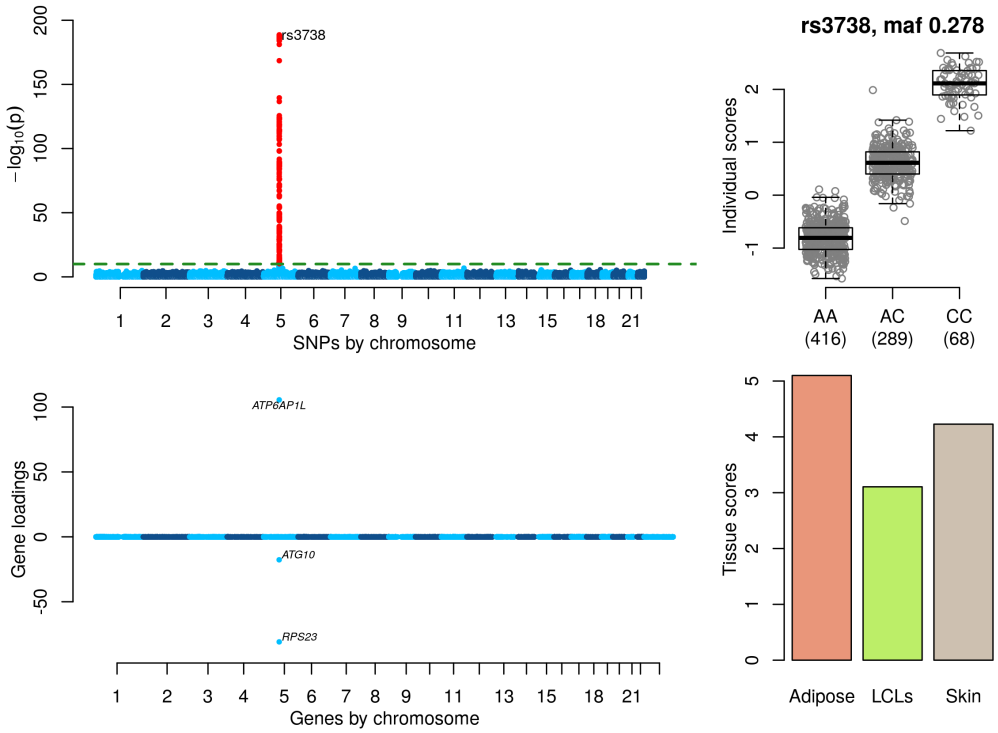
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 5

Robust component describing a *cis* effect.

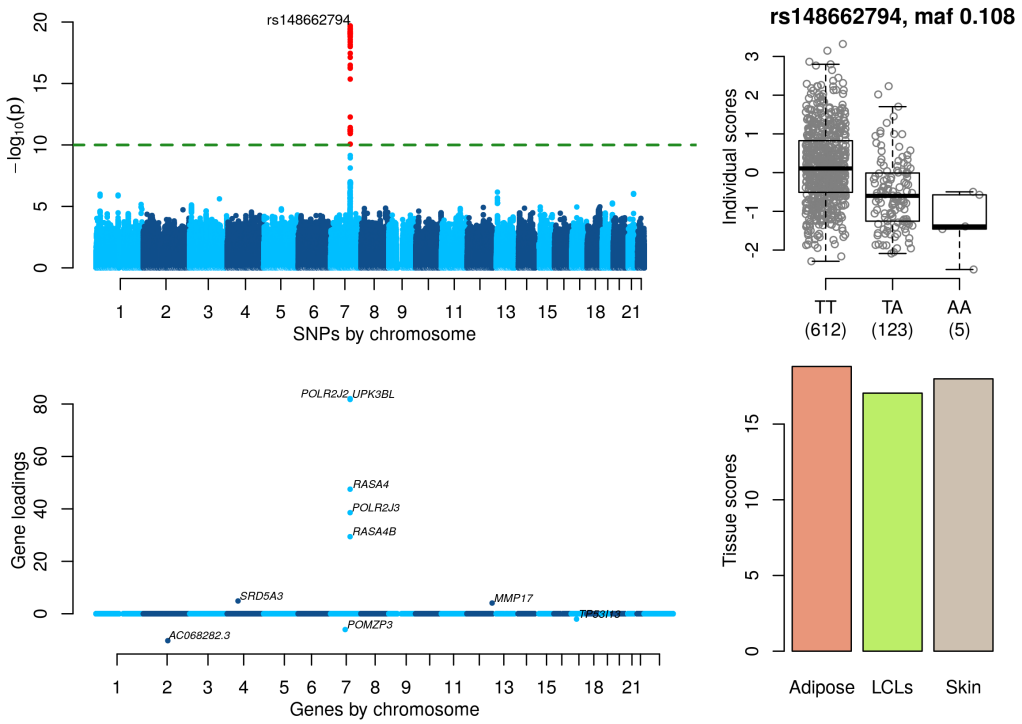
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 6

Robust component describing a *cis* effect.

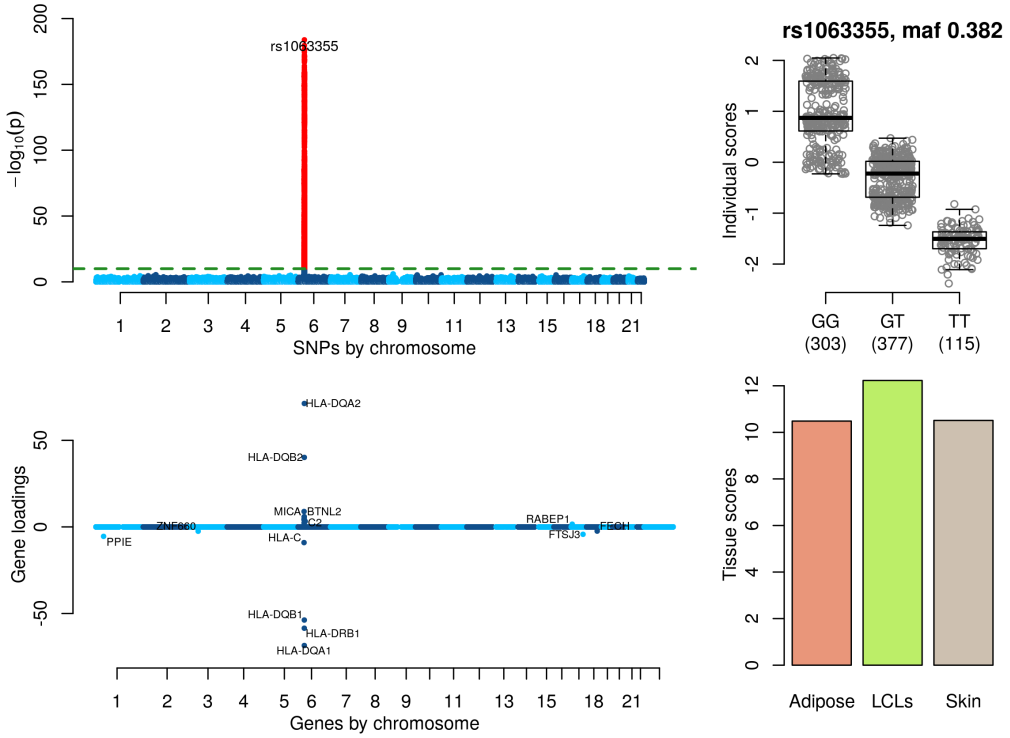
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 7

Robust component describing a *cis* effect.

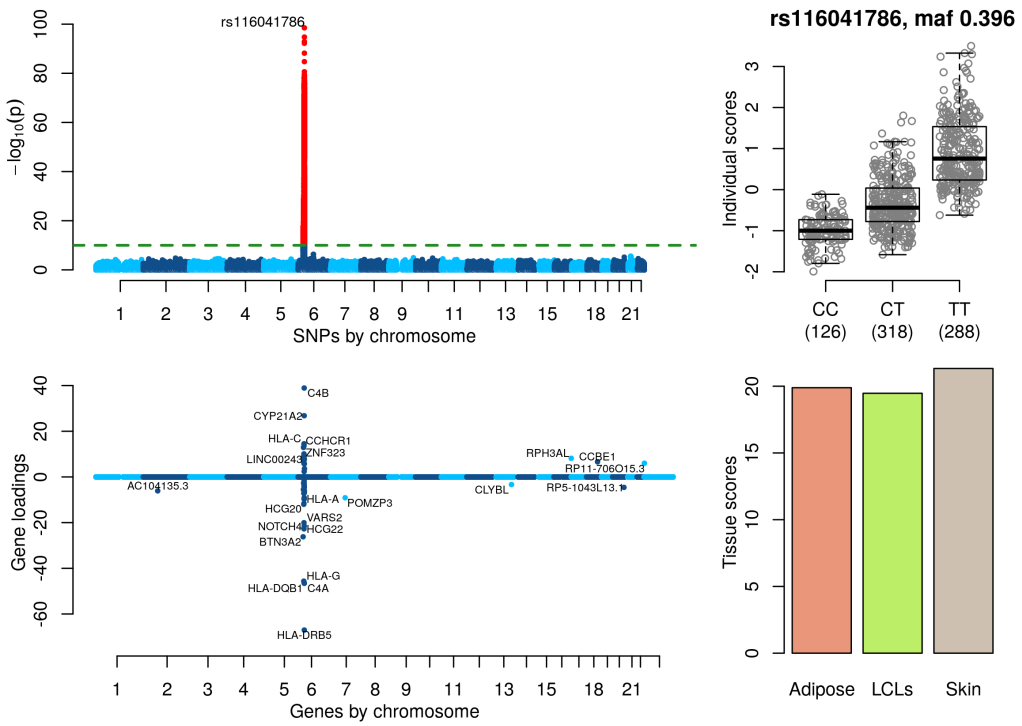
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 8

Robust component describing a *cis* effect.

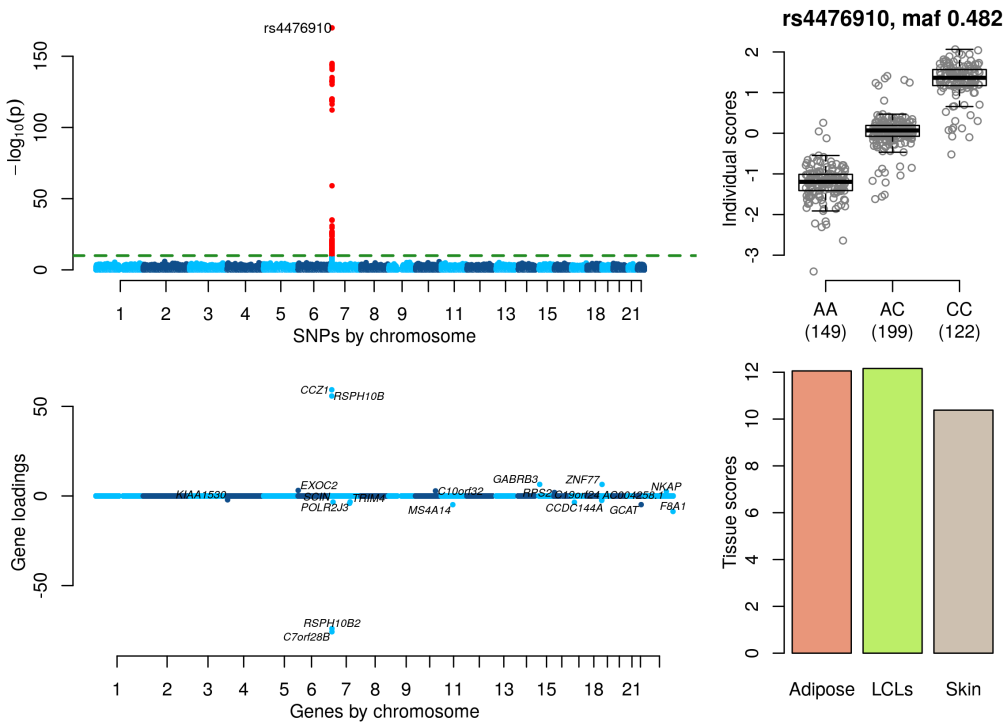
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 9

Robust component describing a *cis* effect.

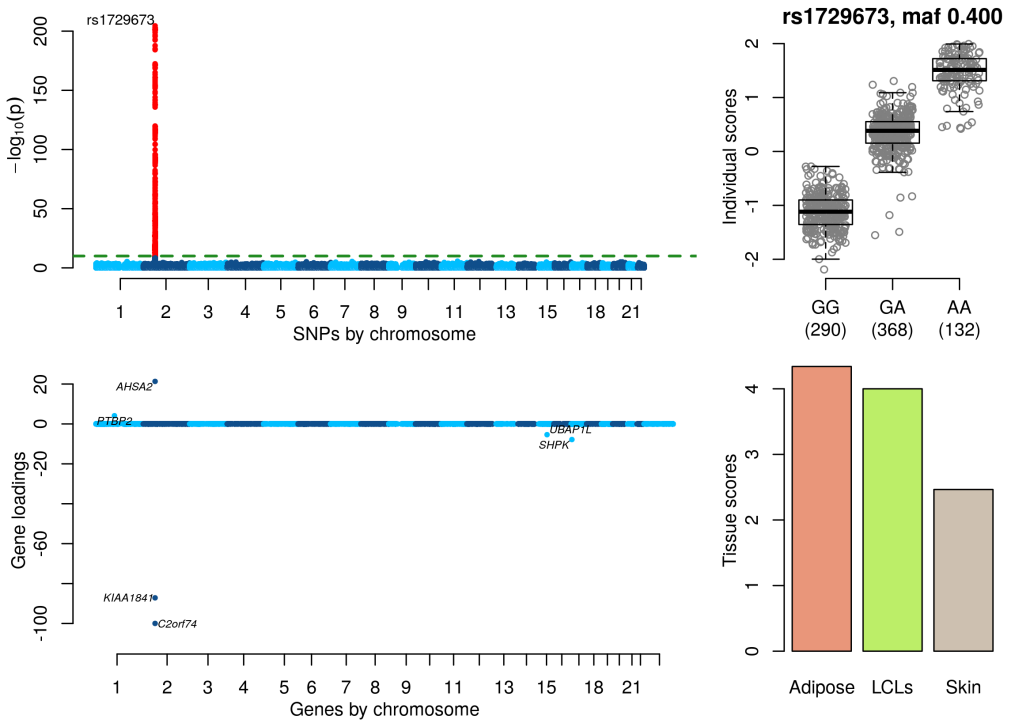
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 10

Robust component describing a *cis* effect.

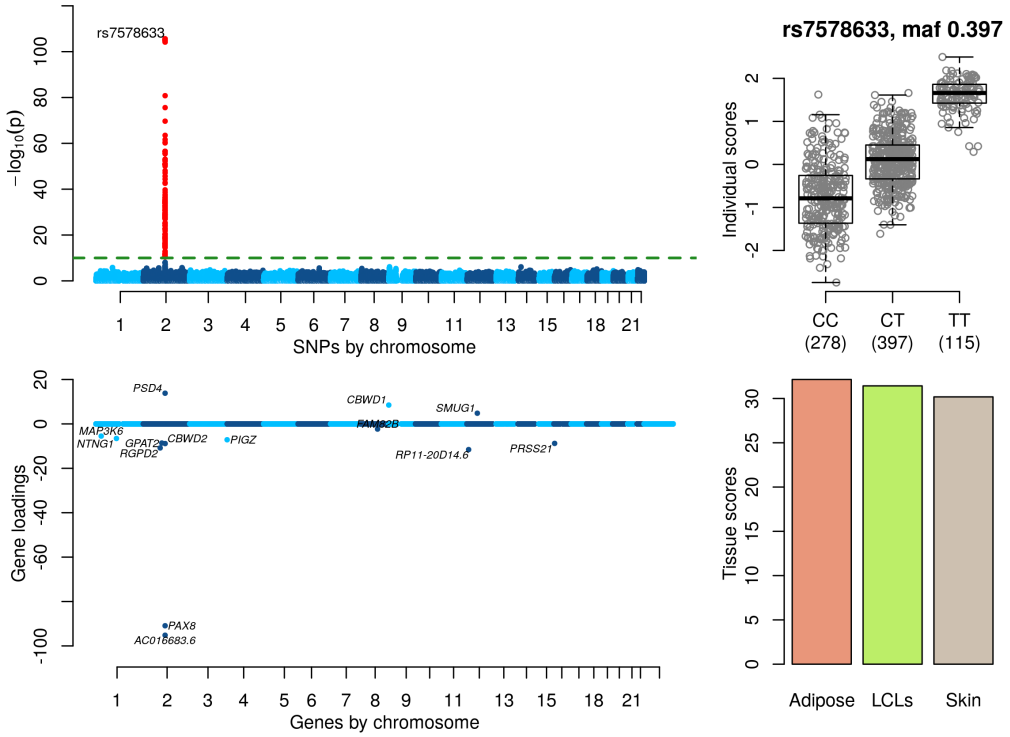
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 11

Robust component describing a *cis* effect.

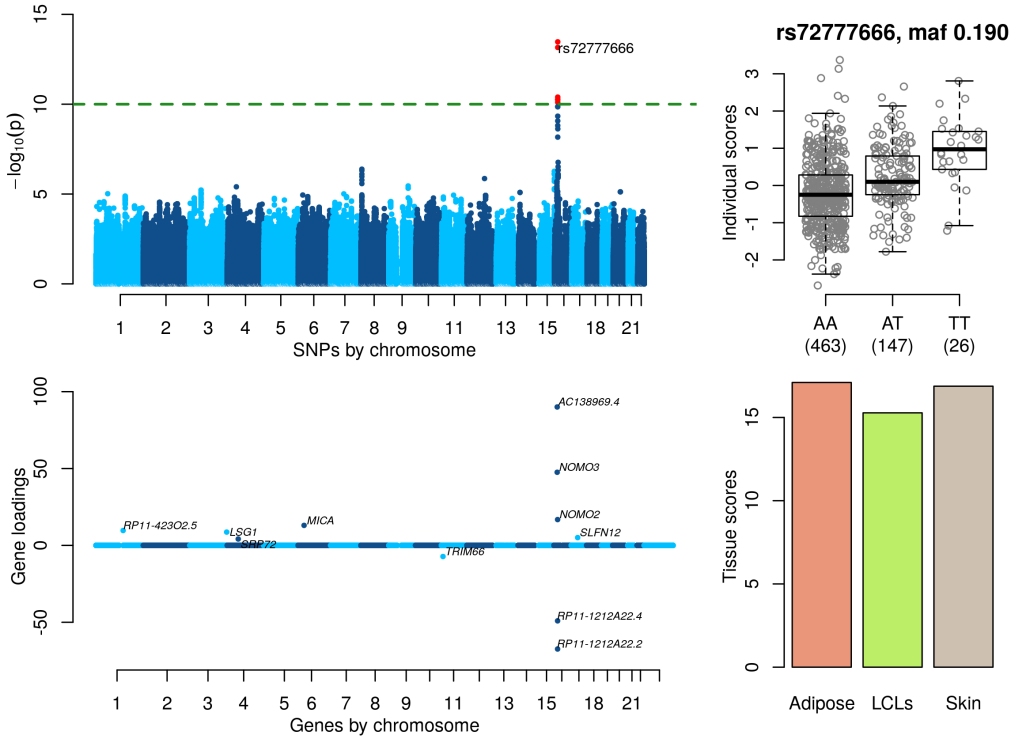
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 12

Robust component describing a *cis* effect.

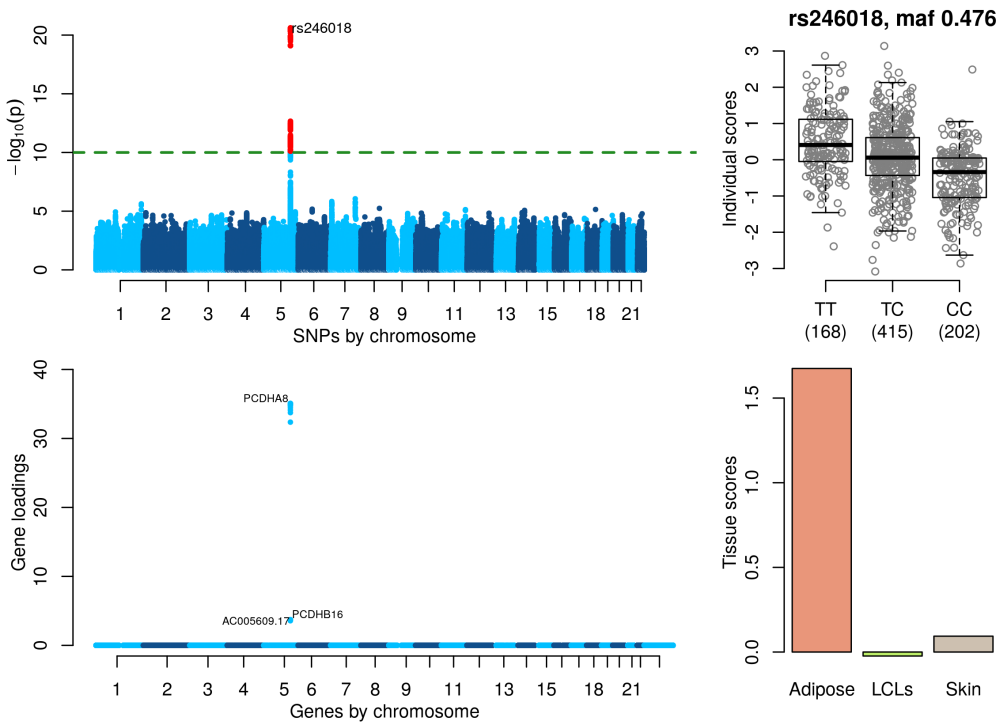
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 13

Robust component describing a *cis* effect.

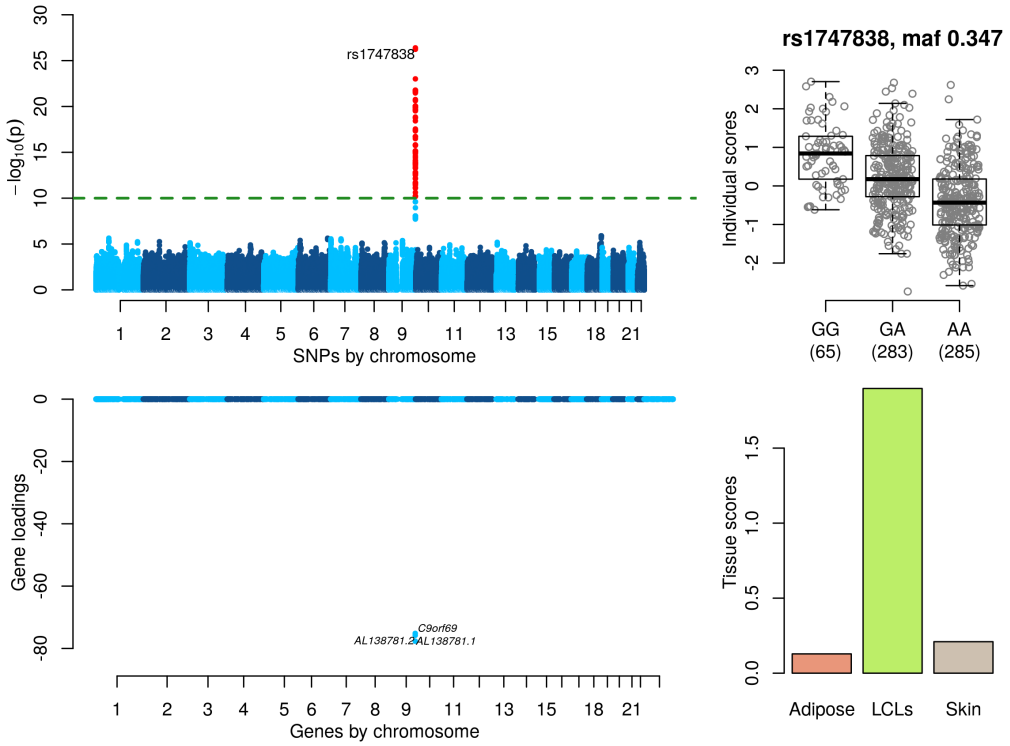
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 14

Robust component describing a *cis* effect.

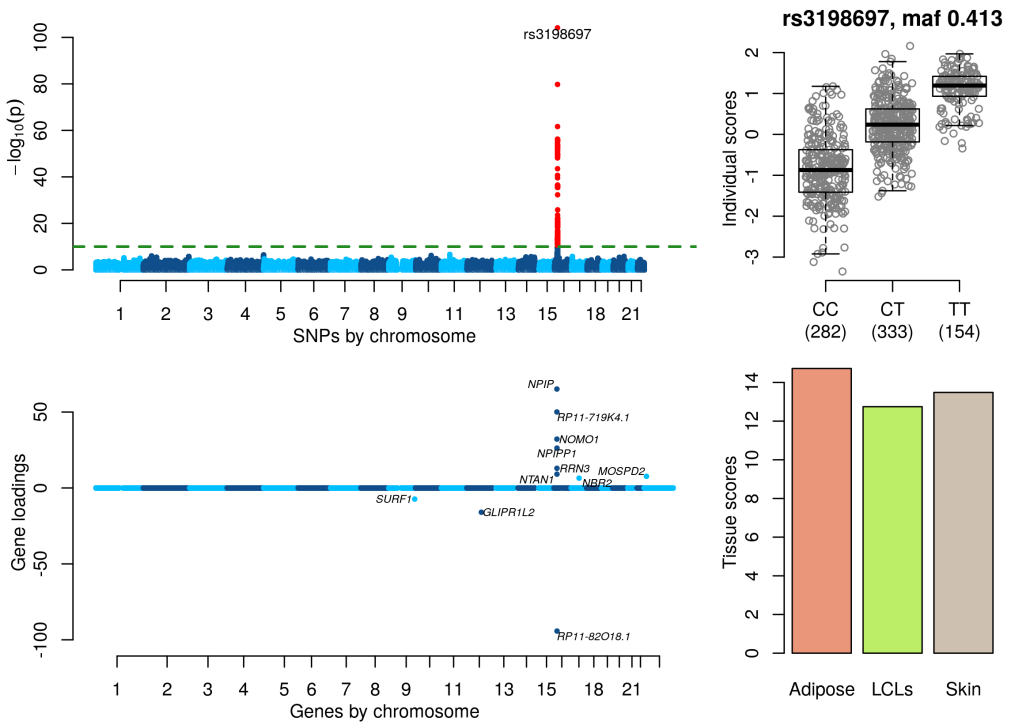
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 15

Robust component describing a *cis* effect.

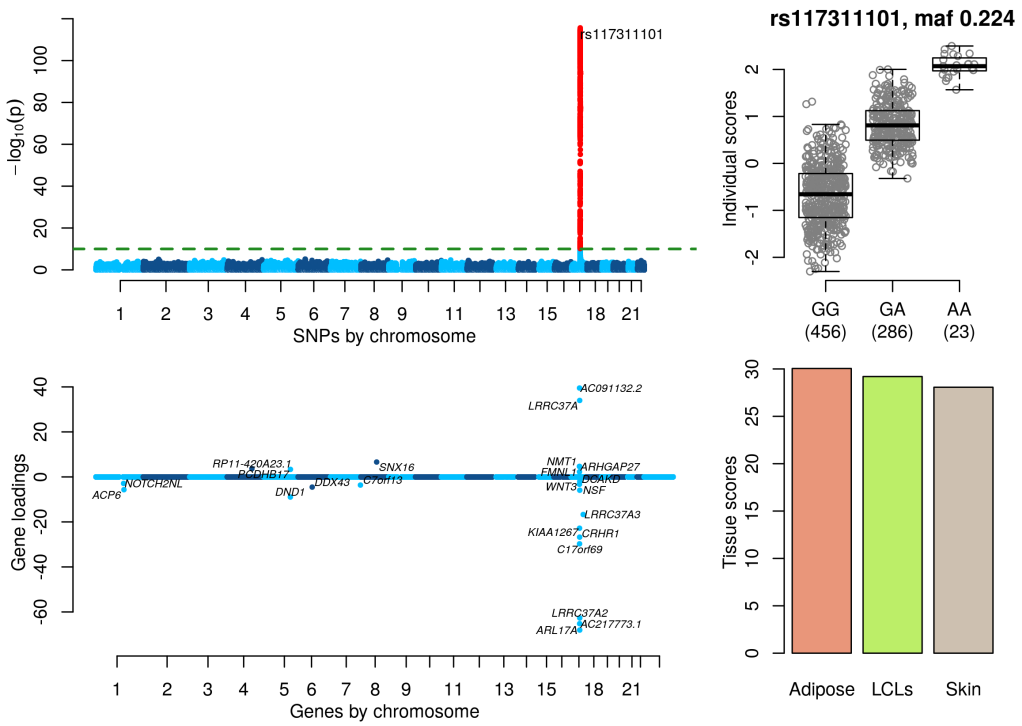
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 16

Robust component describing a *cis* effect.

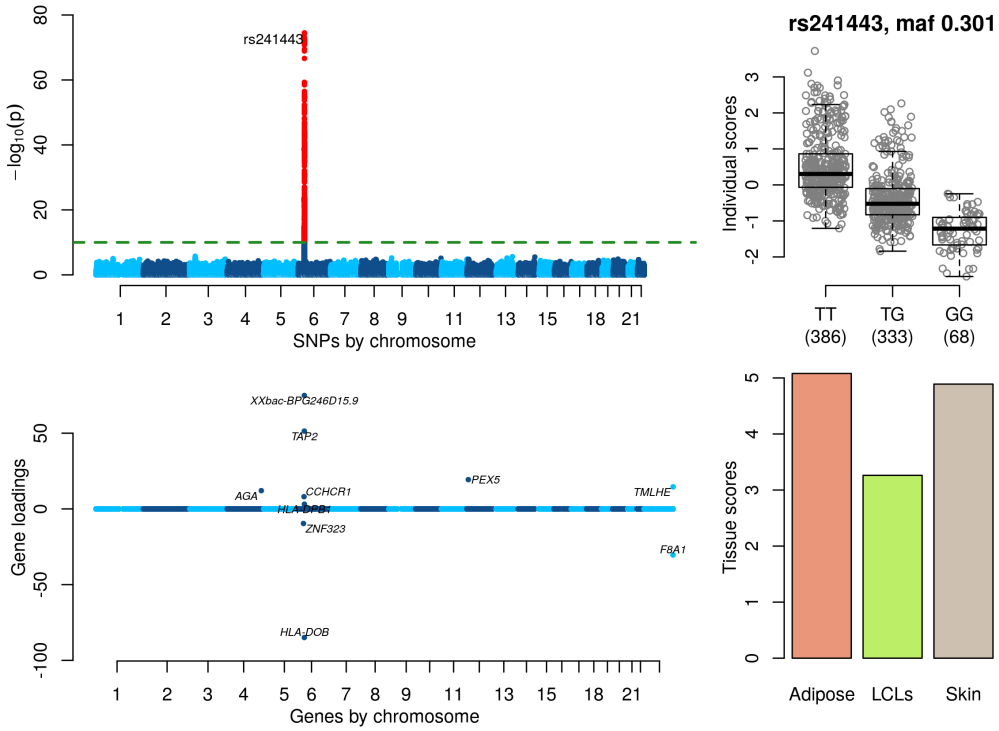
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 17

Robust component describing a *cis* effect.

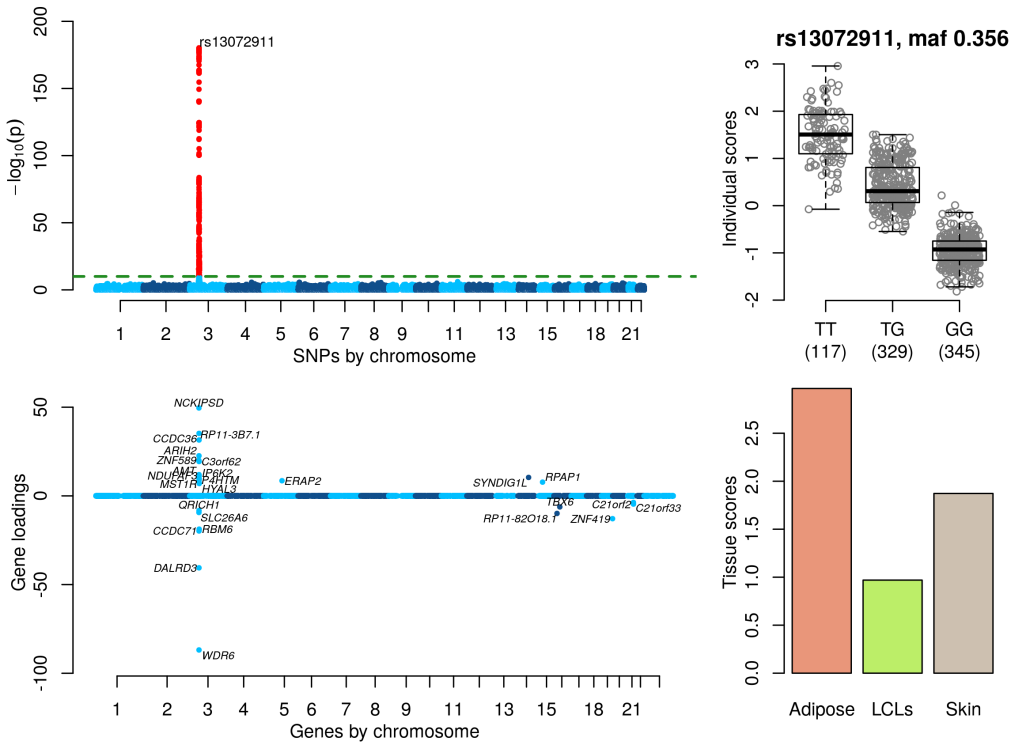
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 18

Robust component describing a *cis* effect.

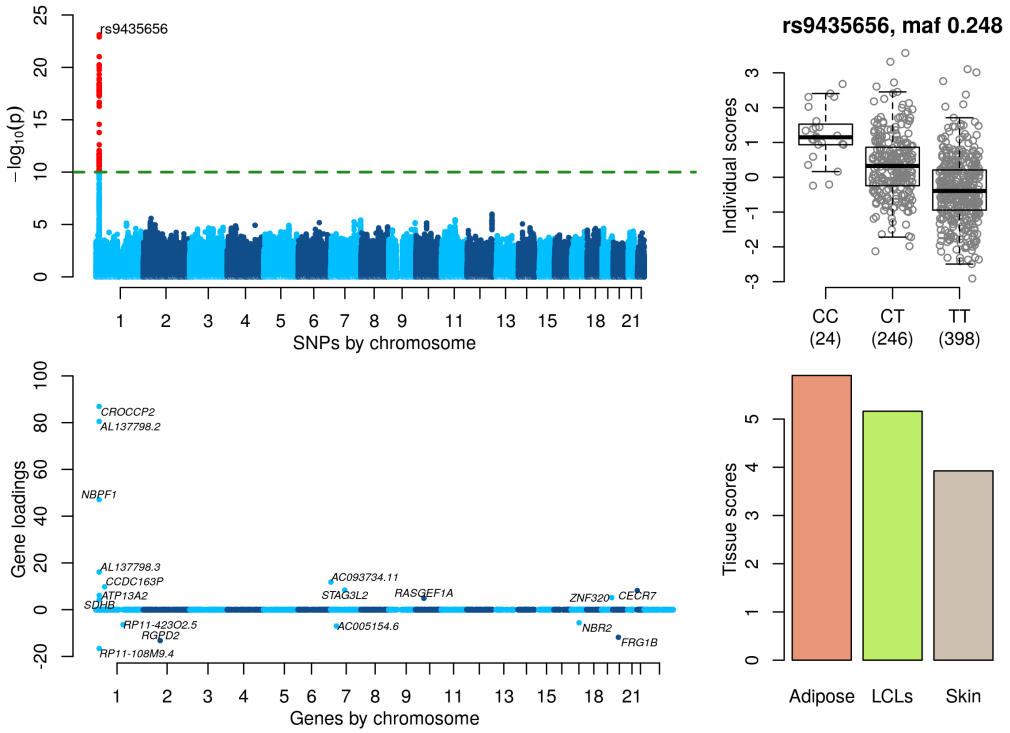
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 19

Robust component describing a *cis* effect.

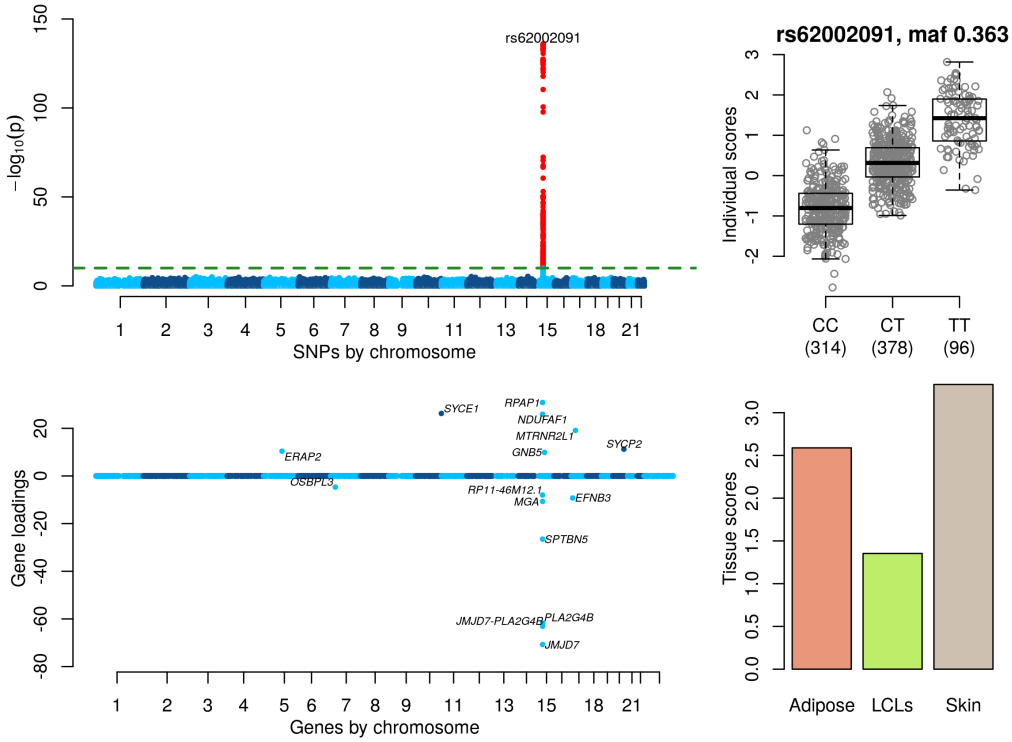
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 20

Robust component describing a *cis* effect.

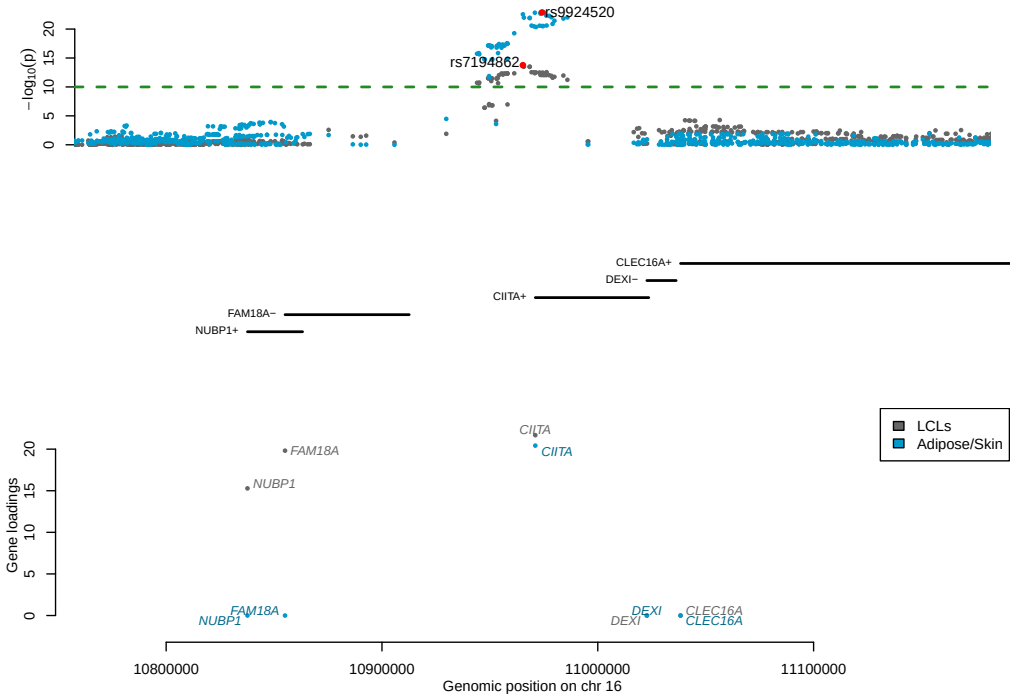
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 21

Robust component describing a *cis* effect.

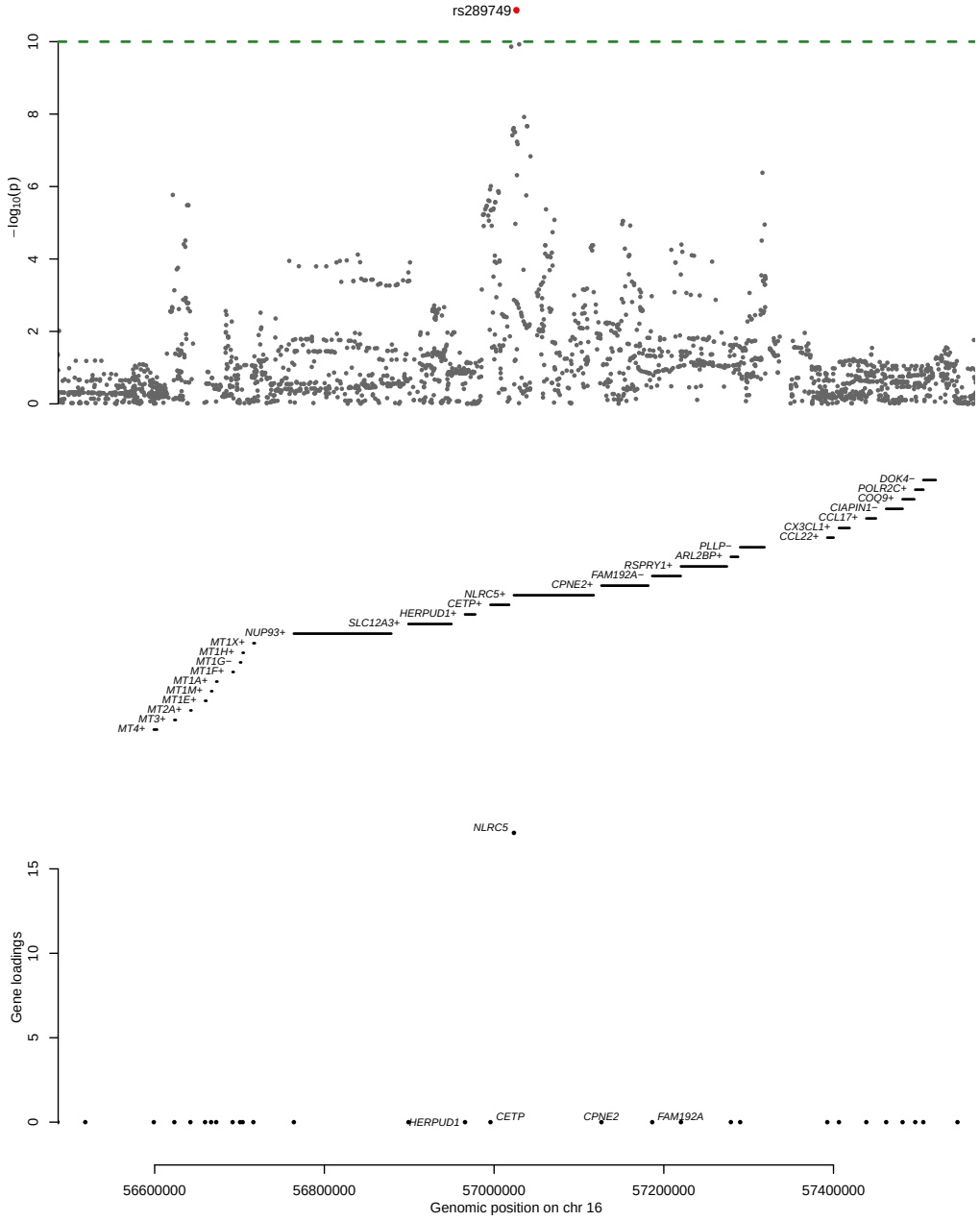
See Supplementary Figure 2 for explanation of the figure.



Supplementary Figure 22

Association plot for the MHC class II regulation components.

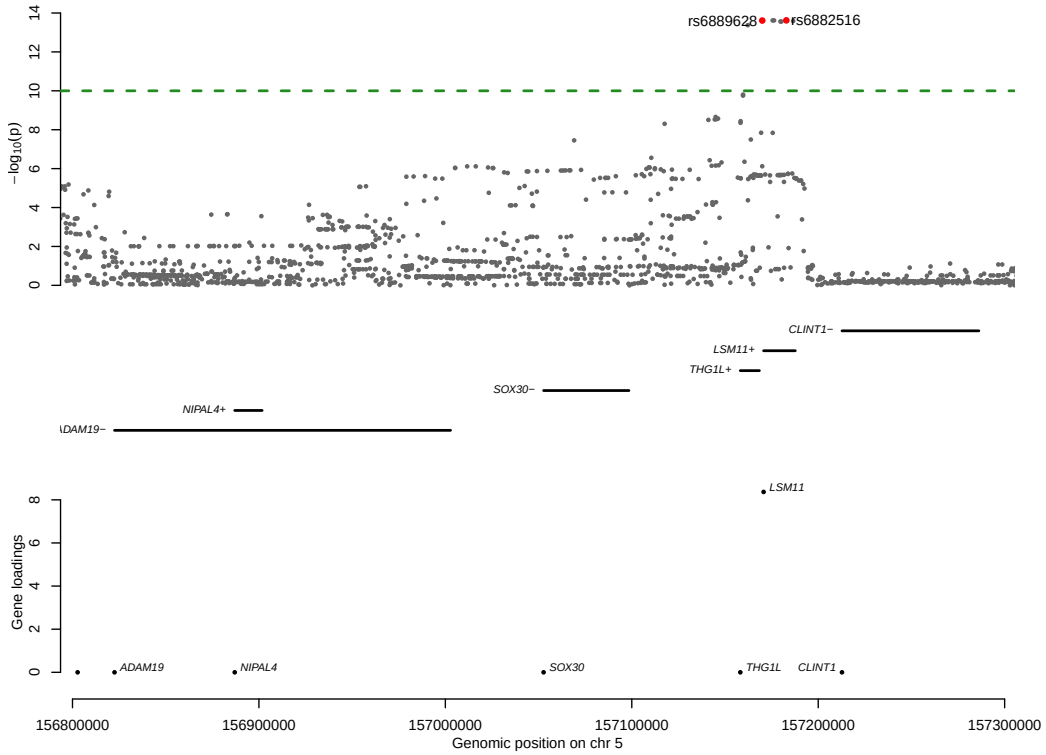
(Top) $-\log_{10}(p\text{-value})$ for association between individual scores and SNPs around region of the significant GWAS signal. (Bottom) Gene loadings in the same region. Results for the component identified in Adipose and Skin are shown in blue and results for the component identified in LCLs are shown in gray.



Supplementary Figure 23

Association plot for the MHC class I regulation component.

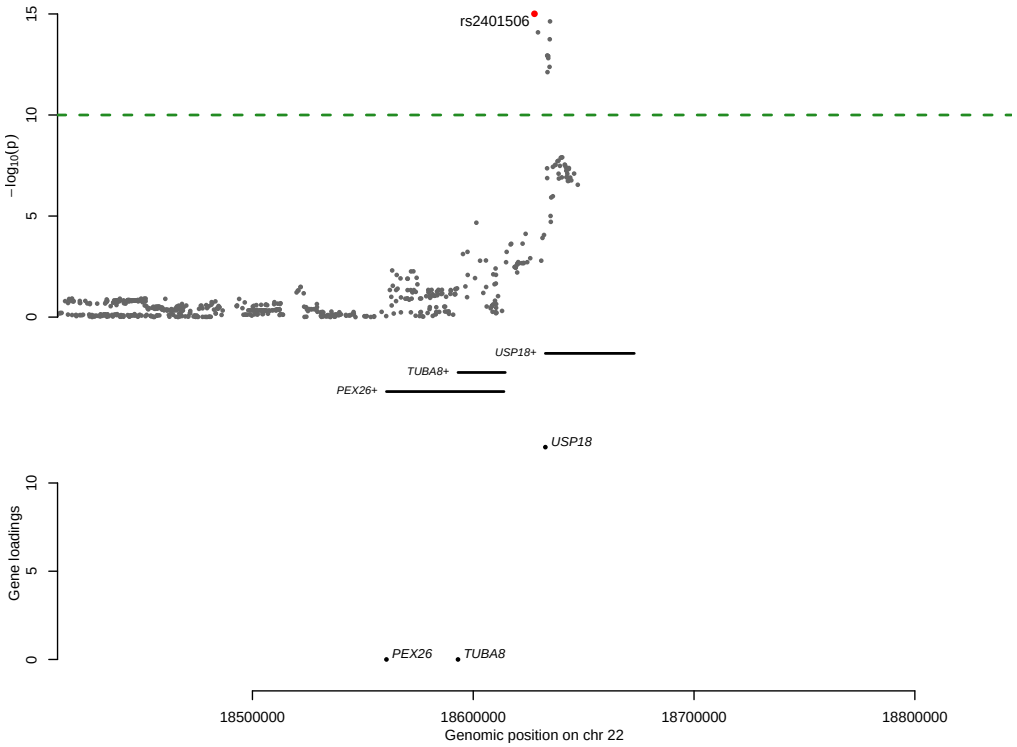
(Top) $-\log_{10}(p\text{-value})$ for association between individual scores and SNPs around region of the significant GWAS signal. (Bottom) Gene loadings in the same region.



Supplementary Figure 24

Association plot for the Histone RNA processing component.

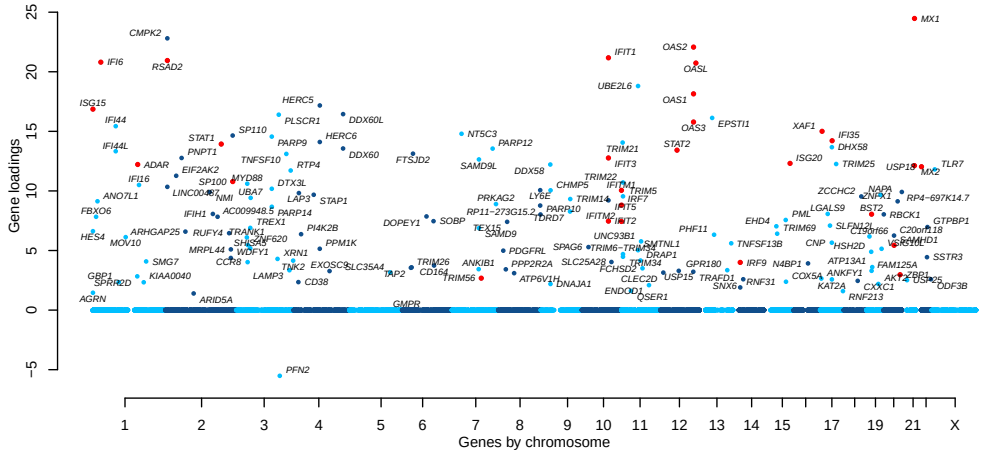
(Top) $-\log_{10}(p\text{-value})$ for association between individual scores and SNPs around region of the significant GWAS signal. (Bottom) Gene loadings in the same region.



Supplementary Figure 25

Association plot for the Type I interferon response component.

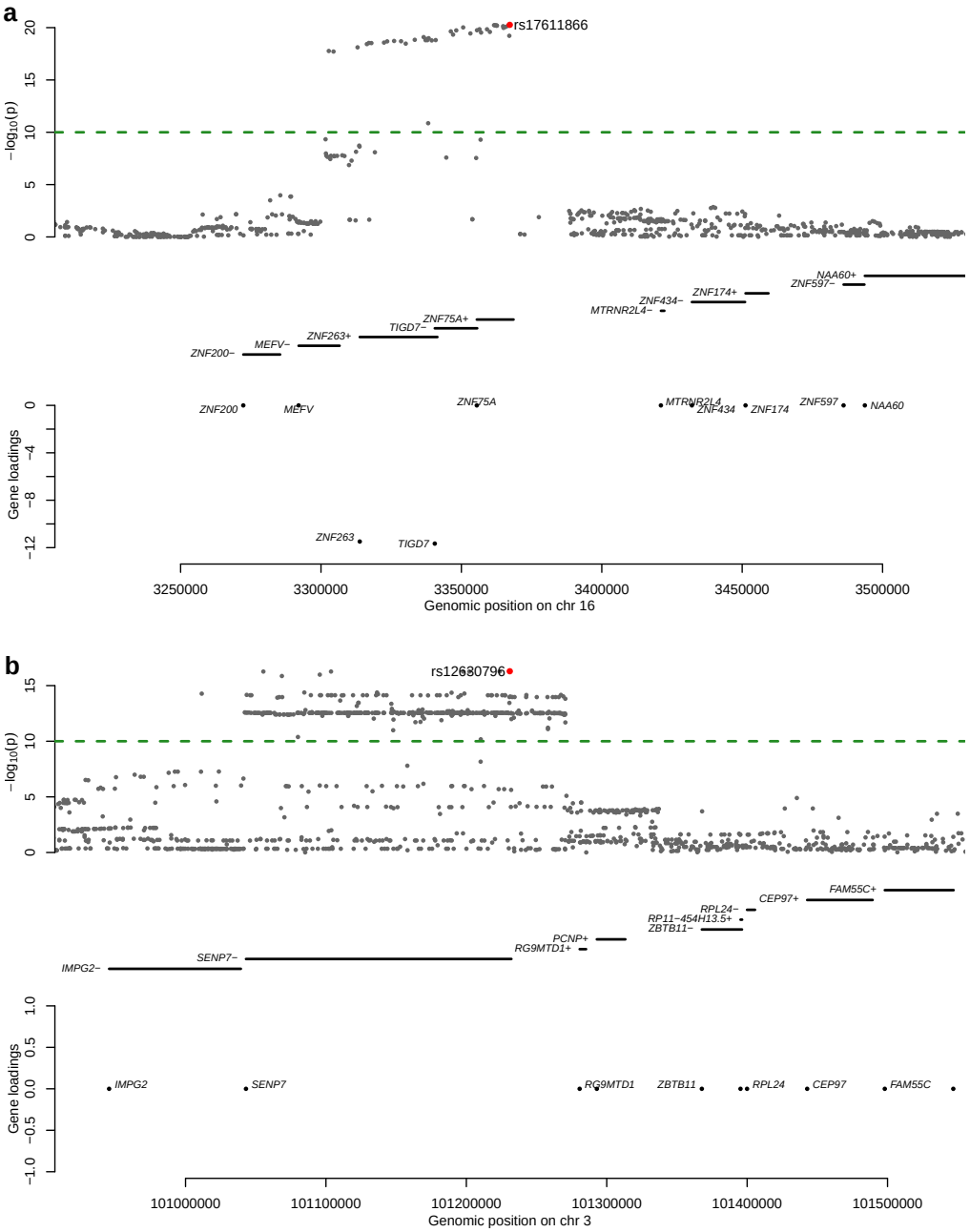
(Top) $-\log_{10}(p\text{-value})$ for association between individual scores and SNPs around region of the significant GWAS signal. (Bottom) Gene loadings in the same region.



Supplementary Figure 26

Gene loadings for Type I Interferon response component.

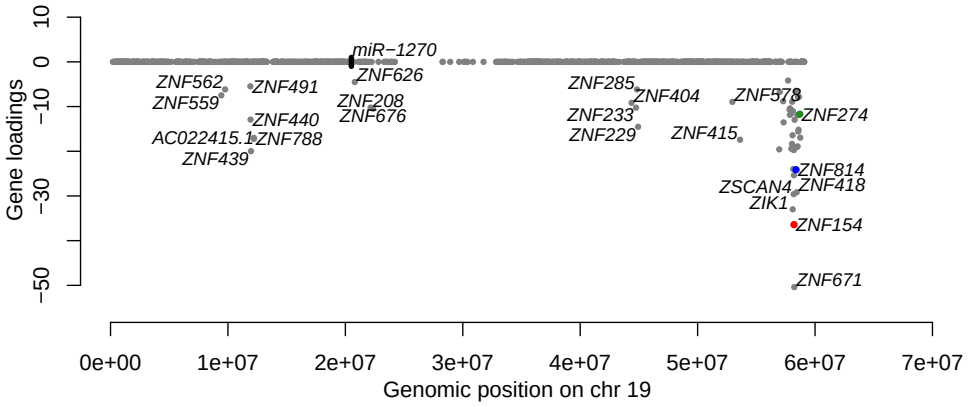
Genes highlighted in red are annotated for the 'response to interferon' gene ontology term (GO:0034340).



Supplementary Figure 27

Association plots for the zinc finger gene network component.

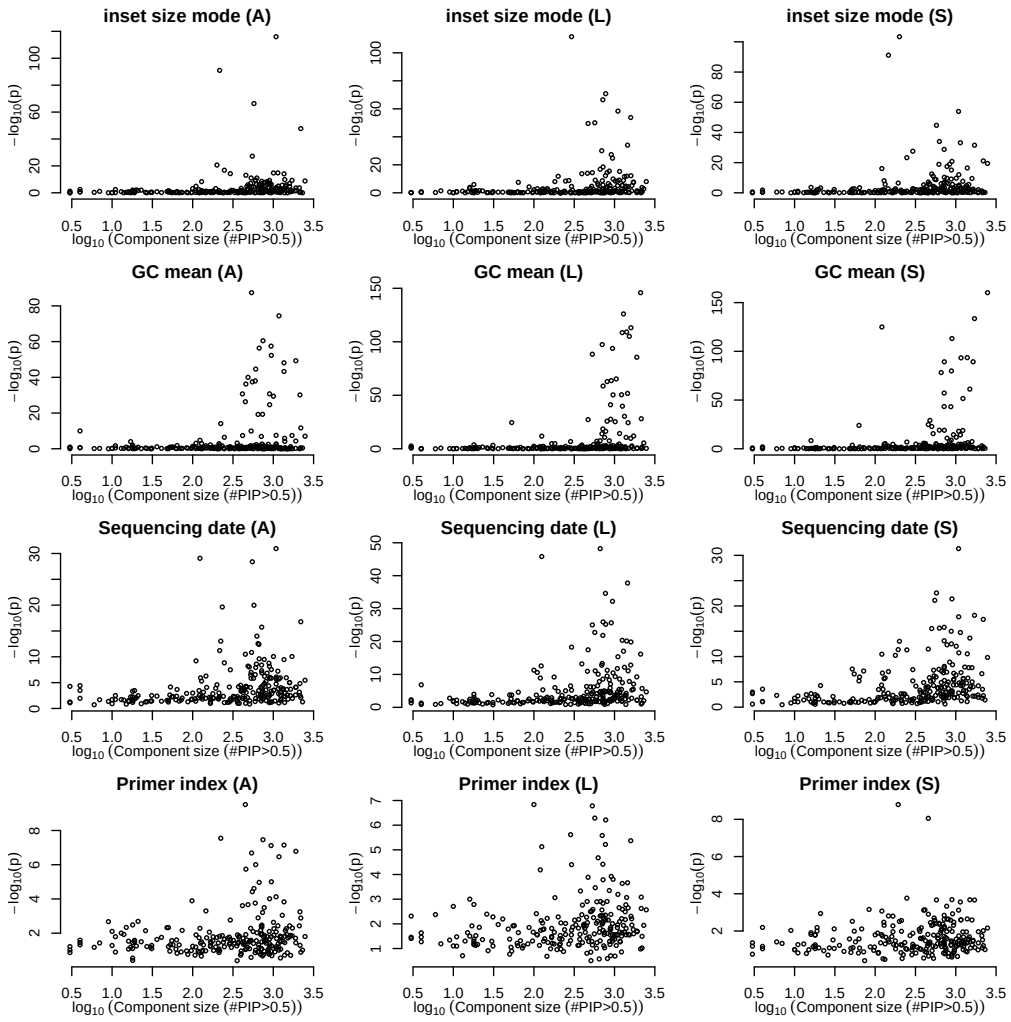
(Top) $-\log_{10}(p\text{-value})$ for association between individual scores and SNPs around region of the significant GWAS signal on chromosome 16 (figure **a**), and chromosome 3 (figure **b**). (Bottom) Gene loadings in the same region.



Supplementary Figure 28

Chromosome 19 gene loadings for zinc finger gene network component.

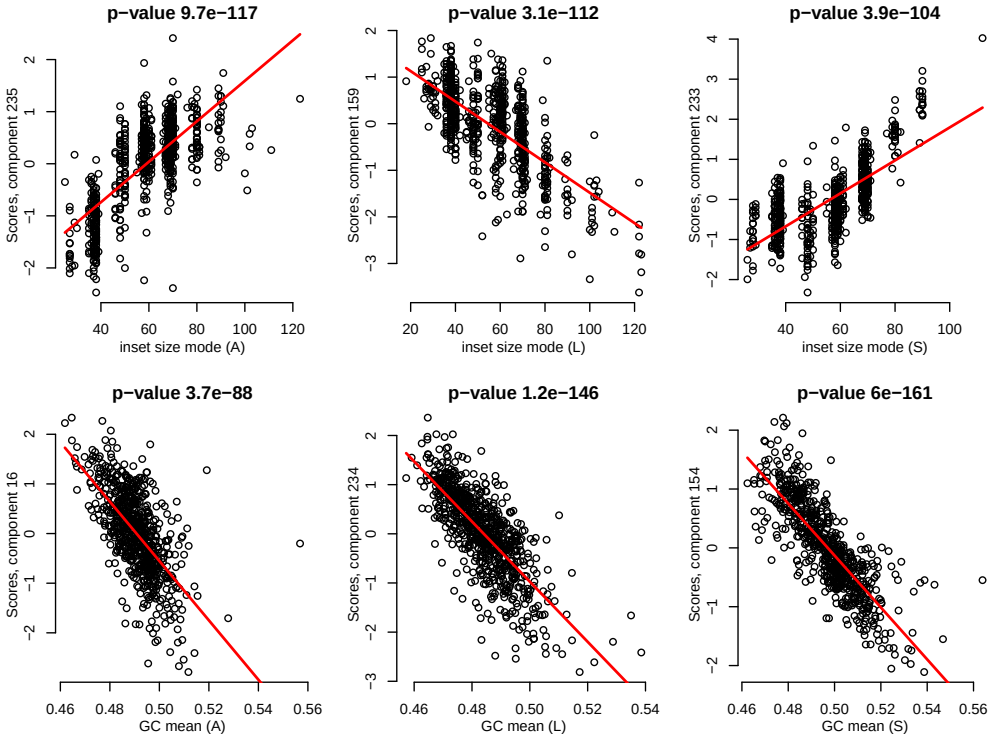
ZNF274, *ZNF154* and *ZNF814* are highlighted in green, red and blue respectively. Position of *miR-1270* on chromosome 19 shown as dash.



Supplementary Figure 29

Association of 236 robust components with batch variables.

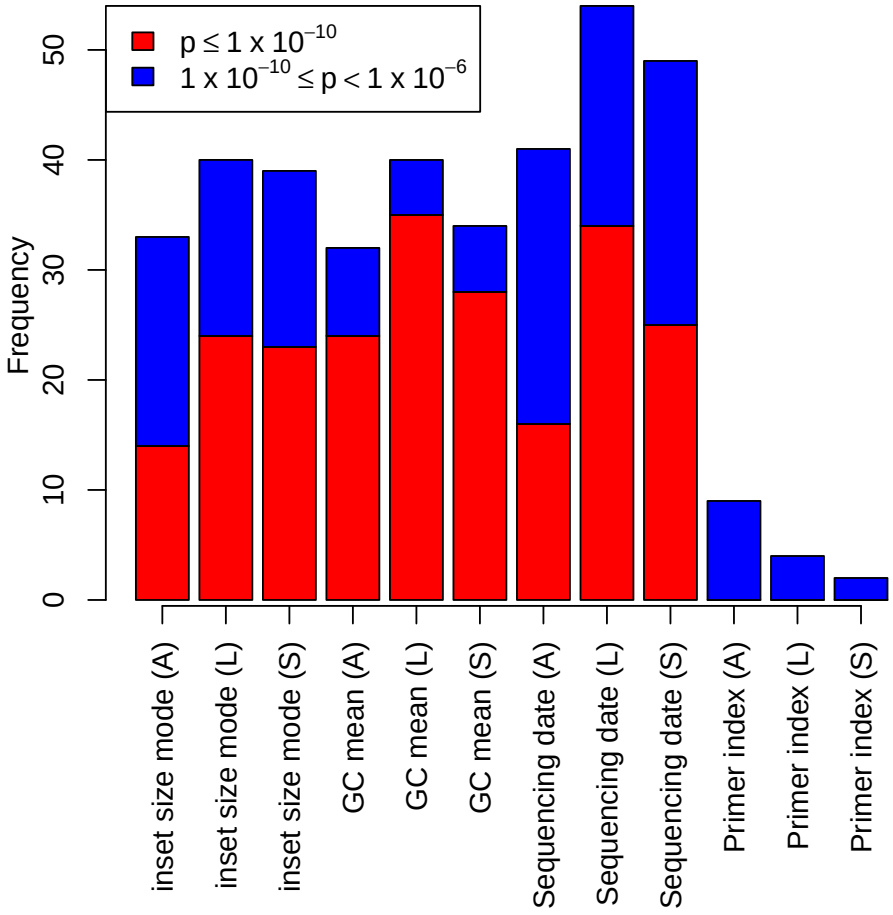
Each plot shows p-values for association between individual scores vectors and 4 sequencing variables in Adipose (A), LCLs (L) or Skin (S). The x-axis shows the $\log_{10}$ component size defined as the number of genes with a PIP>0.5; $-\log_{10}(p\text{-value})$ is plotted on the y-axis.



Supplementary Figure 30

Association of 236 robust components with batch variables.

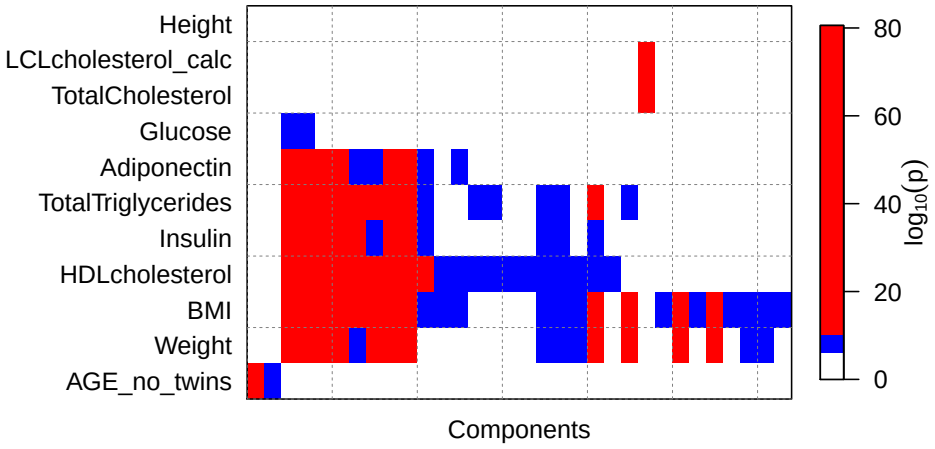
Scatter plots of most significant association of sequencing variables (GC mean and inset size mode) with robust component scores in all three tissues (A: Adipose, L: LCLs, S: Skin). A regression line is shown in red.



Supplementary Figure 31

Association of 236 robust components with batch variables.

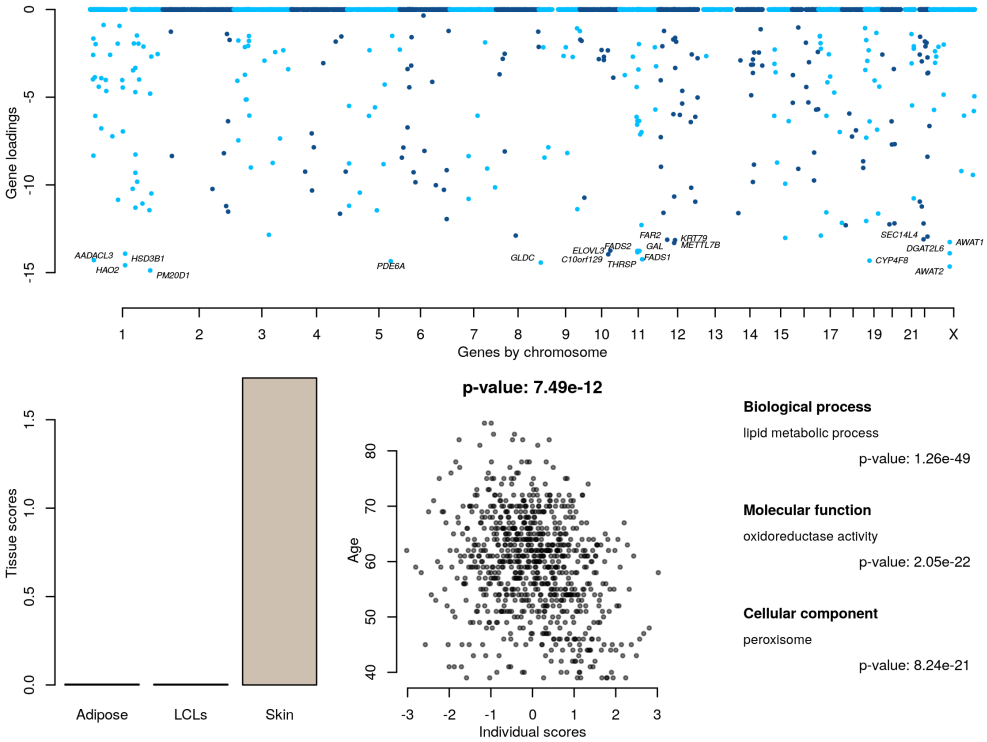
Barplot showing the number of robust components with a significant association with four batch variables measuring properties of RNA sequencing across three tissues (A:Adipose, L: LCLs, S:Skin). The plot shows the numbers of associations with p-values between 1×10^{-6} and 1×10^{-10} in blue and less than 1×10^{-10} in red.



Supplementary Figure 32

Summary of associations between 236 robust components and 11 measured phenotypes.

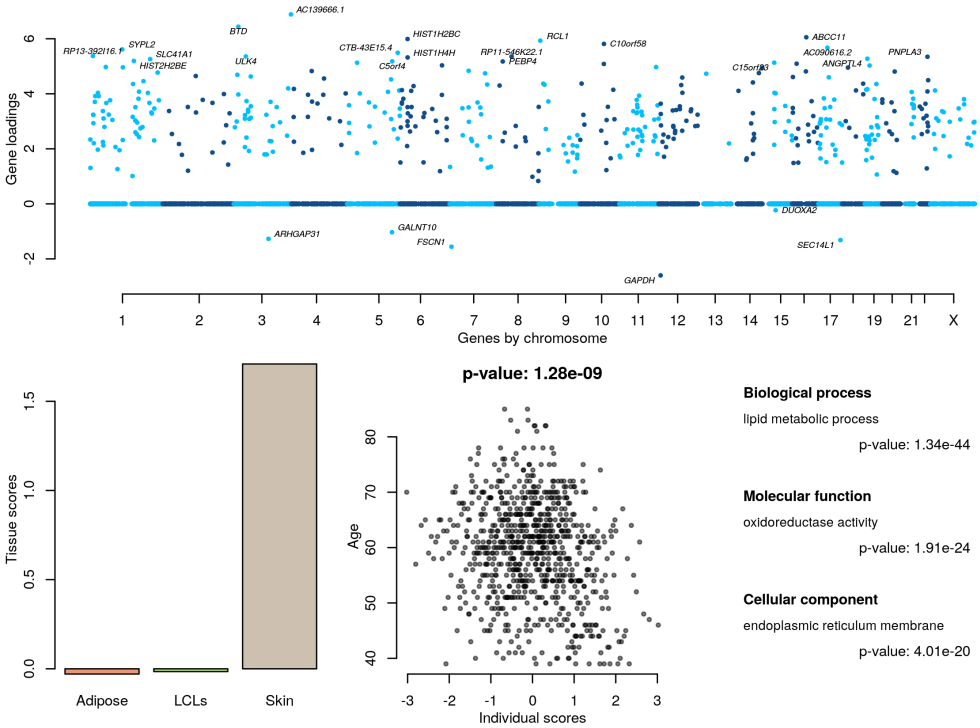
p-values less than 1×10^{-6} are shown in blue and those less than 1×10^{-10} shown in red. Only component with a significant association ($< 1 \times 10^{-6}$) (32 in total) have been plotted and components with similar patterns of association have been placed nearby. Association for age has been performed with one member of each twin pair removed.



Supplementary Figure 33

Robust component associated with age.

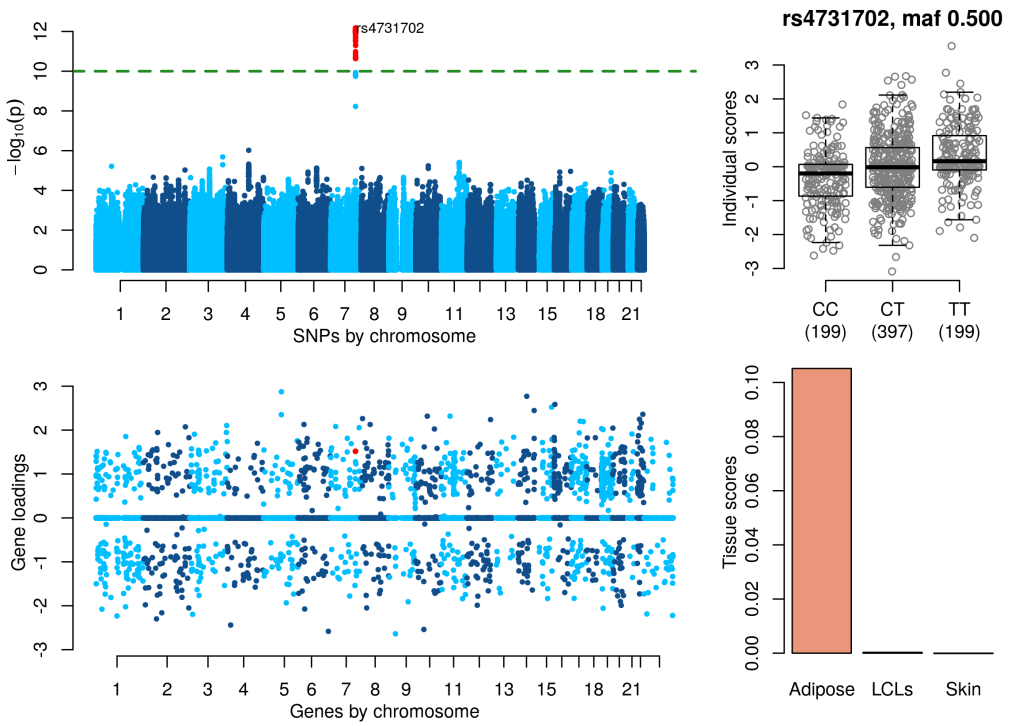
(Top) Gene loadings for the component. (Bottom left) Tissue scores vector for the component shown as a barplot. (Bottom middle) Scatter plot with the component's individual scores on the x-axis and age on the y-axis. (Bottom right) Most enriched gene ontology term and corresponding *p*-value for each ontology category.



Supplementary Figure 34

Robust component associated with age.

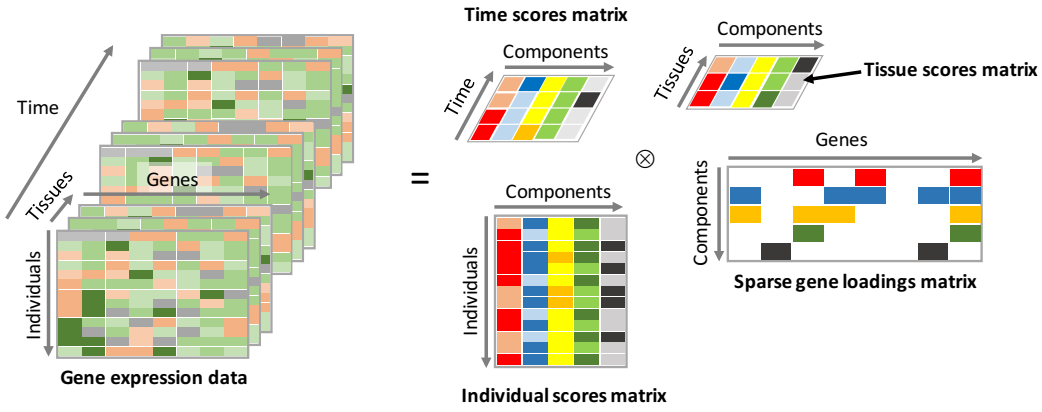
(Top) Gene loadings for the component. (Bottom left) Tissue scores vector for the component shown as a barplot. (Bottom middle) Scatter plot with the component's individual scores on the x-axis and age on the y-axis. (Bottom right) Most enriched gene ontology term and corresponding p-value for each ontology category.



Supplementary Figure 35

Component identifying KLF14 as a *trans*-regulator from the run with the highest value of the model's negative free energy.

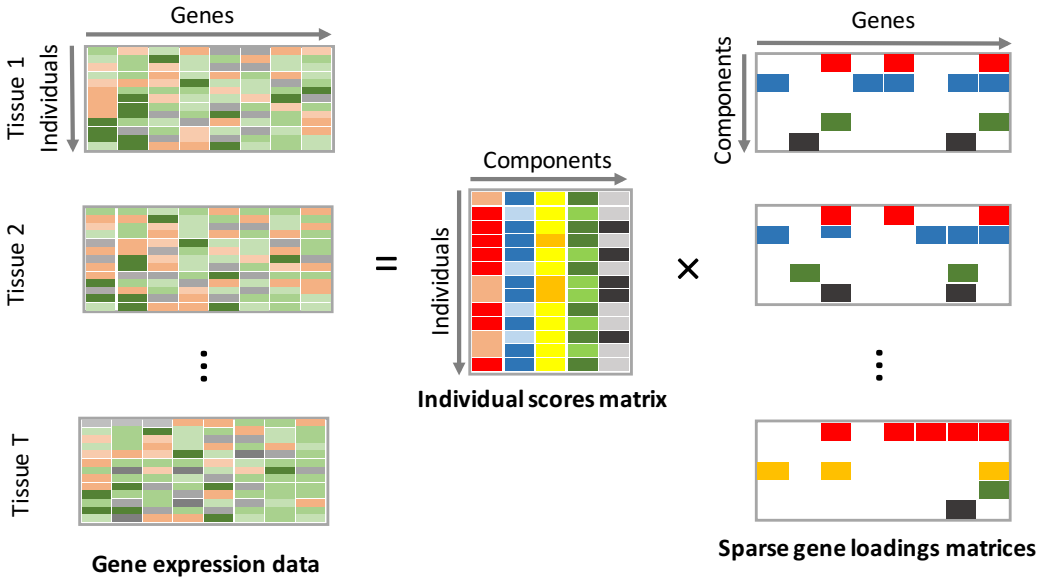
(Top left) Results for a GWAS with the individual scores vector as a phenotype. (Top right) Boxplot of individual scores stratified by genotypes at the lead GWAS SNP rs4731702. (Bottom left) Gene loadings for the component with KLF14 highlighted in red. (Bottom right) Tissue scores vector for the component shown as a barplot.



Supplementary Figure 36

Graphical representation of a four-dimensional decomposition.

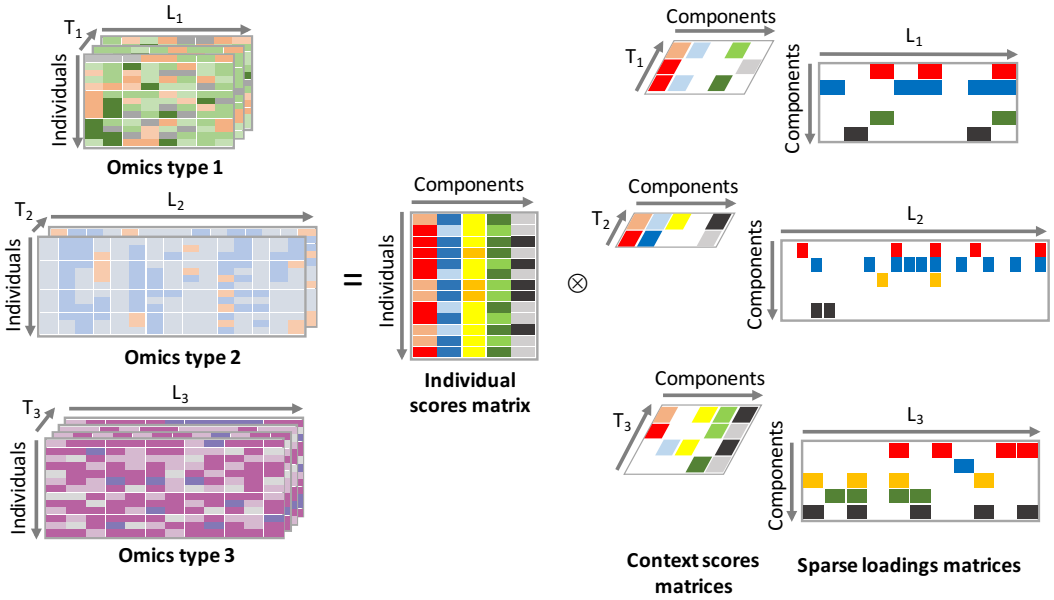
The left of the figure shows a 4D data set consisting of gene expression in multiple tissues at multiple time points. As in Figure 1from the main paper, the data is decomposed into a sparse gene loadings matrix, an individual scores matrix and a tissue scores matrix. We additionally estimate a time scores matrix to deal with the added dimension, which describes the activity of each component at different time points.



Supplementary Figure 37

Graphical representation of a linked decomposition for gene expression data in multiple tissues.

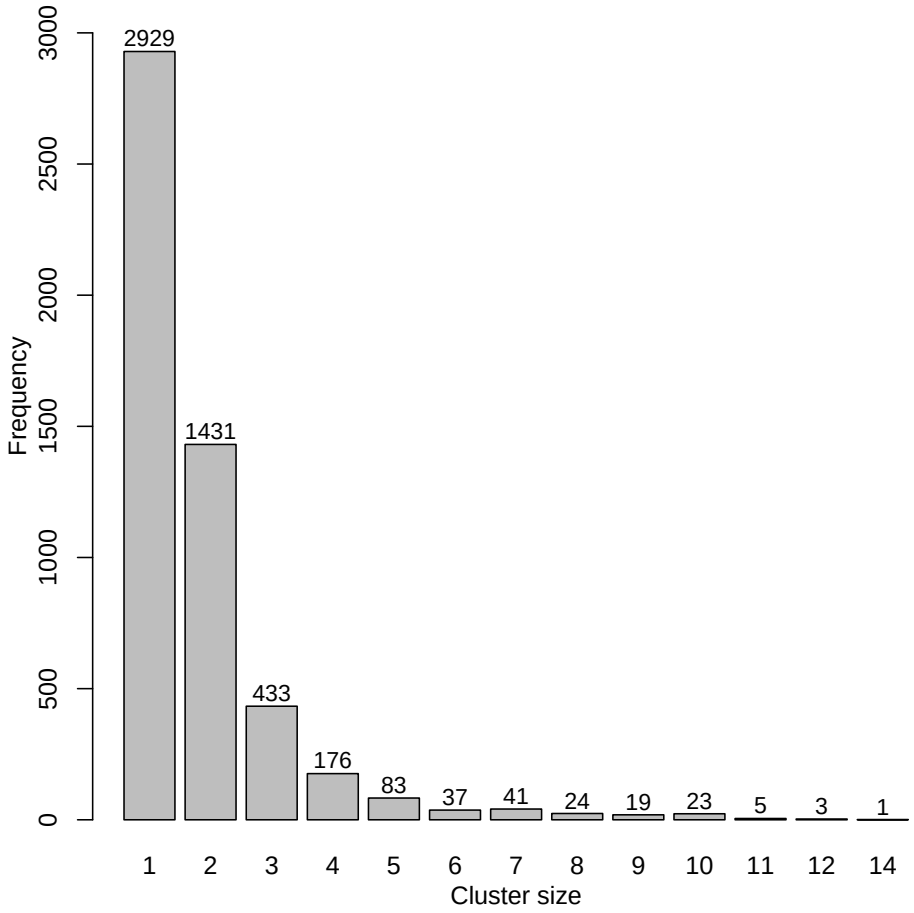
A matrix decomposition is applied to gene expression data for each tissue. The matrix decompositions identify a tissue specific gene loadings matrix with a shared individual scores matrix common to all decompositions.



Supplementary Figure 38

Graphical representation of a linked decomposition for several genomic assays that have a 3D array of data available.

A tensor decomposition is applied to each data type. The decompositions identify different loadings matrices for each data type and a shared individual scores matrix.



Supplementary Figure 39

Distribution of cluster size obtained when clustering components across 10 runs.

Robust components are defined as those with a cluster of size of 5 or more.

Supplementary Tables

Component	SNP (chr)	No. genes in component (PIP>0.5)	No. genes in component and associated with SNP (p-value < 0.05) (A, L, S)
MHC class II*	rs7194862 (16)	18	15, 13, 12
	rs9924520 (16)	31	19, 18, 15
MHC class I	rs289749 (16)	20	1, 0, 17
Histone RNA processing	rs6882516 (5)	31	1, 27, 3
Type I interferon	rs2401506 (22)	160	12, 122, 13
ZNF gene network**	rs12630796 (3)	57	15, 24, 37
	rs17611866 (16)		23, 20, 24

Supplementary Table 1: Summary of components shown in Figures 2-6 in main paper. The table gives the number of genes identified within each of the components we identify, and the number of these genes associated with the component's lead SNP via marginal associations. For each component, column 3 contains the number of genes in the component with a PIP>0.5. Column 4 gives the number of these genes that are marginally associated with the component's lead SNP using a p-value threshold of 0.05 (A:Adipose, L:LCLs, S:Skin). *Results for two components describing a regulation pathway involving the MHC class II genes are given. **For the ZNF gene network component, results for the most significant SNPs from clusters on chromosome 3 and chromosome 16 are presented.

	rs7194862			rs9924520		
	Adipose	LCLs	Skin	Adipose	LCLs	Skin
PI4KB	6.93e-03	5.49e-01	3.79e-02	5.21e-03	8.78e-02	1.72e-01
RFX5	2.33e-03	1.25e-03	5.02e-03	1.08e-03	1.50e-02	2.93e-03
LY9	6.49e-01	4.01e-02	1.45e-01	8.68e-01	2.37e-02	7.80e-01
ZNF672	9.23e-01	1.51e-02	5.03e-01	7.95e-01	2.82e-03	9.82e-01
GRK7	1.37e-02	1.58e-01	2.08e-01	4.34e-02	4.93e-01	3.28e-01
CD74	6.80e-09	8.18e-06	1.23e-05	1.75e-08	1.98e-04	6.23e-05
BTN2A2	6.59e-01	1.43e-02	7.61e-02	5.39e-01	5.27e-03	1.14e-01
HLA-E	8.14e-01	7.55e-01	7.28e-01	8.60e-01	8.88e-01	6.33e-01
HLA-DRA	4.02e-09	4.35e-09	1.96e-08	1.37e-08	2.39e-08	1.94e-07
HLA-DRB5	5.37e-02	3.15e-02	7.57e-02	6.29e-03	3.75e-03	3.55e-02
HLA-DRB1	1.75e-02	1.31e-02	1.15e-02	1.21e-02	2.62e-02	1.93e-02
HLA-DQA1	4.27e-04	1.90e-01	5.59e-03	5.39e-04	2.33e-01	1.80e-02
HLA-DQB1	1.20e-01	8.48e-02	7.94e-01	4.38e-02	1.12e-01	3.88e-01
HLA-DQA2	2.88e-02	3.53e-01	1.34e-01	2.13e-02	1.80e-01	2.19e-02
HLA-DQB2	2.74e-04	2.45e-03	4.68e-01	6.69e-04	8.19e-03	2.24e-01
HLA-DOB	7.12e-03	1.04e-01	6.81e-02	6.40e-04	1.49e-01	6.39e-02
HLA-DMB	7.34e-08	2.32e-05	3.95e-07	1.23e-08	1.17e-04	3.77e-07
XXbac-BPG181M17.5	8.37e-06	3.61e-05	1.61e-06	4.55e-06	1.35e-04	9.96e-06
HLA-DMA	1.51e-08	2.46e-04	2.22e-14	2.89e-08	6.39e-04	3.23e-13
HLA-DOA	5.66e-15	8.18e-07	9.27e-19	3.48e-16	7.41e-07	5.56e-19
HLA-DPA1	3.74e-11	3.21e-07	1.66e-05	4.63e-10	1.43e-05	1.71e-04
HLA-DPB1	2.64e-08	1.77e-04	7.66e-06	3.89e-08	4.68e-04	2.16e-05
TPP1	2.23e-01	1.81e-01	2.15e-01	1.71e-01	3.55e-01	4.75e-01
GOLGA8B	4.84e-01	2.07e-01	1.16e-01	1.18e-01	4.91e-02	9.01e-02
NUBP1	5.58e-01	9.60e-03	5.11e-01	1.62e-01	2.53e-02	6.86e-01
FAM18A	3.58e-01	5.93e-03	8.40e-01	7.27e-01	5.84e-02	7.31e-01
CIITA	1.44e-12	2.33e-09	2.26e-06	2.08e-11	2.84e-07	8.93e-07
HAUS5	4.21e-01	2.74e-02	7.23e-01	6.54e-01	3.23e-01	2.71e-01
MIA	3.61e-01	8.94e-02	6.53e-02	6.84e-01	3.51e-01	3.85e-02
AC008537.2	8.98e-01	5.27e-01	8.79e-01	9.88e-01	7.13e-01	7.77e-01
PARVG	6.68e-01	7.38e-02	7.90e-02	8.77e-01	6.58e-02	1.97e-01
FATE1	4.63e-01	7.94e-01	2.32e-01	2.26e-01	7.91e-01	1.85e-01

Supplementary Table 2: Direct associations between rs9924520 and rs7194862 on chromosome 16 and genes with a PIP>0.5 in the MHC Class II regulation components.

	rs289749		
	Adipose	LCLs	Skin
BTN3A2	3.81e-01	5.70e-01	3.92e-04
BTN2A2	3.18e-01	6.14e-01	3.97e-04
BTN3A1	2.13e-01	3.03e-01	3.48e-05
BTN3A3	6.78e-01	4.79e-01	1.55e-03
BTN2A1	7.65e-01	7.42e-02	6.78e-05
HLA-F	9.93e-01	1.45e-01	3.02e-12
HLA-A	2.95e-01	6.80e-01	1.22e-09
HLA-E	2.19e-01	7.60e-02	8.85e-05
HLA-C	5.92e-01	1.19e-01	1.46e-07
HLA-B	8.35e-01	9.59e-01	1.35e-10
TAP2	6.38e-01	3.66e-01	1.56e-01
XXbac-BPG246D15.9	8.29e-01	8.82e-01	1.62e-01
PSMB8	8.17e-01	4.12e-01	2.56e-06
PSMB9	5.59e-01	5.74e-01	2.09e-05
TAP1	4.17e-01	3.92e-01	4.30e-07
PATL2	3.44e-01	9.33e-01	2.92e-02
B2M	9.04e-01	4.87e-01	4.72e-08
TRIM69	2.77e-01	7.97e-01	4.18e-01
SPG21	2.70e-02	9.69e-01	2.11e-02
NLRC5	7.40e-02	1.63e-01	1.37e-28

Supplementary Table 3: Direct associations between rs289749 on chromosome 16 and genes with a PIP>0.5 in the MHC Class I regulation component.

	rs6882516		
	Adipose	LCLs	Skin
HIST2H2BE	4.23e-01	5.40e-09	8.79e-01
HIST3H2A	6.48e-01	4.02e-03	8.56e-01
LSM11	9.77e-01	5.57e-33	2.64e-01
HIST1H1C	3.01e-02	1.17e-12	8.07e-01
HIST1H2BC	5.37e-01	4.66e-06	4.63e-01
HIST1H2AC	5.88e-01	2.87e-09	2.47e-01
HIST1H1E	4.41e-01	6.08e-01	4.95e-01
HIST1H2BD	9.39e-01	9.65e-09	9.27e-01
HIST1H3D	1.62e-01	2.70e-07	2.04e-01
HIST1H2AD	5.22e-01	3.15e-06	9.91e-01
HIST1H2BF	7.65e-01	4.87e-02	6.90e-01
HIST1H4E	1.84e-01	6.20e-03	4.83e-02
HIST1H2BG	4.65e-01	6.48e-05	4.05e-01
HIST1H2AE	1.24e-01	1.12e-05	5.04e-01
HIST1H3E	6.05e-01	1.37e-03	6.38e-01
HIST1H2BH	2.14e-01	1.45e-02	3.14e-01
HIST1H3G	6.58e-01	1.28e-02	3.04e-02
HIST1H4H	5.40e-01	9.65e-09	3.37e-01
HIST1H2BJ	8.38e-01	1.81e-12	5.11e-01
HIST1H2AG	6.28e-01	2.02e-03	2.34e-01
HIST1H2BK	6.04e-01	2.65e-12	7.33e-01
HIST1H4I	9.37e-01	6.58e-07	6.88e-01
HIST1H3H	9.11e-01	2.73e-05	1.71e-01
HIST1H2BN	5.73e-01	3.90e-02	6.99e-01
HIST1H1B	2.41e-01	3.40e-01	7.09e-01
HIST1H2BO	6.12e-01	1.37e-08	1.70e-01
OR2B6	1.97e-01	1.47e-05	3.39e-01
H2AFX	1.97e-01	2.48e-02	2.73e-01
LRRC23	4.43e-01	6.72e-01	4.31e-01
HIST4H4	9.90e-01	2.53e-06	7.52e-01
EYA2	1.48e-01	1.85e-01	1.67e-02

Supplementary Table 4: Direct associations between rs6882516 on chromosome 5 and genes with a PIP>0.5 in the histone RNA processing component.

	rs2401506		
	Adipose	LCLs	Skin
HES4	4.33e-01	1.77e-02	6.06e-01
ISG15	1.46e-01	1.14e-08	1.31e-01
AGRN	9.51e-02	1.07e-01	6.88e-01
FBXO6	3.82e-01	7.12e-01	2.40e-01
ANO7L1	9.55e-01	1.81e-01	7.72e-01
IFI6	3.16e-01	6.67e-05	6.12e-02
IFI44L	4.19e-02	3.93e-05	1.10e-01
IFI44	4.41e-02	2.42e-07	1.81e-01
GBP1	2.28e-01	3.37e-02	9.55e-01
MOV10	2.59e-01	3.68e-05	2.63e-01
SPRR2D	4.53e-01	3.90e-01	4.61e-01
ADAR	7.74e-01	4.10e-06	3.31e-02
IFI16	9.30e-01	5.46e-04	6.94e-01
KIAA0040	4.37e-01	4.36e-01	8.16e-01
SMG7	2.58e-01	5.17e-02	5.28e-02
LINC00487	9.60e-02	9.54e-04	7.07e-03
CMPK2	2.29e-01	1.43e-10	1.28e-01
RSAD2	8.46e-02	1.41e-11	1.39e-01
EIF2AK2	5.07e-01	2.83e-09	3.87e-01
PNPT1	5.26e-01	8.01e-05	9.95e-01
ARHGAP25	8.03e-02	1.28e-02	6.00e-02
ARID5A	6.40e-01	9.94e-01	8.88e-01
NMI	2.85e-01	1.97e-05	8.34e-01
IFIH1	1.97e-01	2.41e-06	9.91e-01
AC009948.5	1.25e-01	3.39e-01	5.61e-01
STAT1	6.57e-01	9.88e-09	8.93e-01
RUFY4	4.70e-01	3.05e-02	2.72e-01
WDFY1	4.09e-03	3.08e-03	6.90e-02
MRPL44	4.78e-01	1.99e-02	3.32e-01
SP110	7.14e-01	4.88e-02	8.76e-01
SP100	7.76e-01	3.89e-01	7.92e-01
TRANK1	1.39e-01	3.83e-02	7.60e-02
MYD88	5.86e-01	4.85e-06	1.70e-01
CCR8	1.37e-01	8.40e-01	6.41e-01
ZNF620	9.73e-01	7.63e-04	9.10e-01
TREX1	6.50e-01	7.25e-03	1.21e-01
SHISA5	1.34e-01	8.58e-03	4.54e-01
UBA7	5.02e-02	4.93e-03	8.95e-01
PARP9	8.11e-01	1.10e-02	2.38e-01
DTX3L	7.98e-01	5.21e-03	6.57e-01
PARP14	2.70e-01	6.83e-02	4.07e-01
XRN1	2.24e-01	1.55e-03	4.82e-01
PLSCR1	3.33e-01	4.53e-09	6.96e-01
PFN2	2.79e-01	3.47e-02	2.47e-01
TNFSF10	3.90e-01	2.93e-03	6.86e-01
LAMP3	4.08e-01	6.80e-02	5.93e-01
RTP4	1.61e-01	2.52e-04	1.46e-01
TNK2	5.35e-01	4.16e-03	1.43e-01
CD38	8.46e-01	7.33e-01	2.25e-01
LAP3	8.12e-01	1.17e-02	8.71e-01
PI4K2B	3.13e-02	1.16e-02	2.86e-01
STAP1	8.68e-01	2.38e-01	1.92e-02
PPM1K	4.97e-01	4.79e-04	6.34e-01
HERC6	1.95e-01	3.54e-04	1.25e-01
HERC5	8.33e-01	3.31e-07	6.23e-01
EXOSC9	9.77e-02	4.30e-01	1.07e-01
DDX60	2.19e-01	1.27e-06	3.77e-01
DDX60L	9.80e-01	2.12e-06	8.15e-01
SLC35A4	3.38e-01	2.71e-02	8.52e-01
TRIM26	2.13e-01	2.36e-01	8.43e-01
TAP2	9.40e-01	2.79e-01	7.53e-01
FTSJD2	6.01e-01	3.07e-05	4.84e-01
DOPEY1	1.27e-01	1.50e-02	1.90e-01
SOBP	3.26e-01	2.21e-04	8.00e-01

CD164	2.58e-01	4.94e-03	8.79e-01
NT5C3	5.93e-01	1.61e-05	4.68e-01
ANKIB1	5.77e-02	4.51e-07	6.65e-01
SAMD9	9.44e-01	7.04e-04	5.23e-01
SAMD9L	7.69e-01	1.81e-05	7.24e-01
TRIM56	3.65e-01	7.82e-02	1.69e-01
PARP12	8.06e-01	3.32e-07	9.93e-01
PRKAG2	2.28e-01	4.66e-03	1.50e-01
PDGFRL	6.62e-01	2.42e-03	4.53e-01
PPP2R2A	7.53e-01	7.80e-03	4.43e-01
TEX15	6.75e-01	5.89e-01	1.61e-01
ATP6V1H	9.92e-03	4.33e-01	7.53e-01
RP11-273G15.2	5.63e-01	4.60e-02	7.40e-01
LY6E	4.08e-01	8.99e-07	3.17e-02
PARP10	6.83e-01	2.83e-04	4.59e-01
DDX58	7.04e-01	1.14e-04	7.73e-01
DNAJA1	6.20e-01	2.79e-02	4.65e-01
CHMP5	8.30e-01	1.82e-07	3.64e-01
TDRD7	1.30e-01	1.74e-04	4.32e-01
TRIM14	8.89e-01	2.29e-01	2.58e-01
SPAG6	4.34e-01	4.96e-01	6.87e-01
IFIT2	1.08e-02	1.52e-05	1.34e-01
IFIT3	9.48e-02	1.84e-03	3.71e-01
IFIT1	3.45e-01	8.56e-10	2.05e-01
IFIT5	4.01e-01	3.53e-05	9.97e-01
SLC25A28	8.65e-01	1.45e-01	7.51e-01
IFITM2	2.33e-01	4.36e-04	1.04e-01
IFITM1	1.31e-01	3.92e-07	4.63e-01
IRF7	5.27e-01	1.27e-06	6.59e-02
TRIM21	8.96e-01	3.42e-05	4.51e-02
TRIM34	2.91e-01	1.22e-04	1.63e-01
TRIM6-TRIM34	2.55e-01	1.19e-04	1.52e-01
TRIM5	8.84e-01	1.45e-02	8.85e-01
TRIM22	2.54e-01	6.94e-08	3.40e-01
QSER1	2.86e-04	3.74e-01	5.07e-01
SMTNL1	1.65e-01	8.08e-03	6.97e-01
UBE2L6	2.04e-01	2.57e-07	1.05e-01
DRAP1	5.79e-01	5.50e-04	4.95e-01
UNC93B1	9.64e-01	5.49e-03	8.22e-01
FCHSD2	1.24e-01	3.08e-01	6.53e-01
ENDOD1	3.34e-01	2.89e-02	1.55e-01
CLEC2D	3.82e-01	6.93e-01	9.79e-01
STAT2	5.33e-01	1.06e-07	8.03e-01
USP15	8.83e-01	2.34e-01	7.49e-01
TRAFD1	9.21e-01	2.92e-01	6.86e-03
OAS1	5.86e-01	6.26e-04	5.25e-01
OAS3	7.69e-02	1.75e-07	5.90e-02
OAS2	4.13e-01	1.59e-08	6.84e-02
OASL	6.58e-03	5.66e-07	7.89e-02
EPSTI1	5.76e-02	9.44e-03	1.75e-01
PHF11	9.81e-01	3.20e-03	3.58e-02
GPR180	9.52e-01	1.02e-01	3.75e-01
TNFSF13B	1.06e-01	1.18e-01	6.20e-01
RNF31	1.67e-01	7.08e-02	8.11e-01
IRF9	4.58e-01	3.86e-02	7.13e-01
SNX6	7.72e-01	6.45e-03	2.78e-01
EHD4	3.67e-01	8.54e-02	4.19e-02
TRIM69	1.52e-01	2.24e-02	7.37e-01
PML	3.16e-01	4.01e-05	1.51e-01
COX5A	3.34e-01	6.43e-01	1.41e-01
ISG20	4.15e-01	7.56e-05	5.28e-01
N4BP1	4.90e-01	4.82e-04	7.11e-02
ANKFY1	2.57e-01	2.98e-03	4.10e-01
XAF1	8.87e-01	2.33e-04	6.60e-01
LGALS9	8.17e-01	4.49e-02	7.03e-01
SLFN12L	9.99e-01	1.15e-02	8.40e-01
CNP	7.20e-01	2.55e-01	6.26e-01
DHX58	4.64e-01	1.53e-02	3.81e-01
KAT2A	2.08e-01	4.20e-01	9.84e-01
IFI35	6.15e-01	7.12e-07	6.83e-01
TRIM25	3.43e-01	3.60e-04	7.30e-01

RNF213	5.24e-01	1.59e-03	8.64e-01
CXXC1	1.23e-01	1.39e-02	5.89e-01
ZCCHC2	8.83e-01	3.56e-04	5.19e-01
C19orf66	4.43e-01	2.82e-02	8.82e-01
HSH2D	1.58e-01	8.50e-02	7.54e-01
BST2	2.46e-01	6.98e-07	1.07e-02
FAM125A	8.94e-01	3.95e-02	8.61e-01
ATP13A1	4.47e-01	4.17e-03	9.44e-03
AKT2	3.30e-01	9.26e-01	3.20e-01
NAPA	1.10e-01	2.14e-02	8.17e-01
VSIG10L	7.25e-01	1.54e-02	6.46e-01
RBCK1	6.57e-01	1.76e-03	3.51e-01
C20orf118	7.06e-01	1.20e-01	3.32e-02
SAMHD1	6.67e-02	1.51e-04	2.52e-03
ZNFX1	2.69e-01	3.98e-04	2.97e-01
ZBP1	2.92e-02	2.35e-03	5.70e-01
RP4-697K14.7	7.17e-01	2.49e-07	1.16e-01
USP25	3.54e-02	1.51e-03	6.24e-01
MX2	3.19e-01	6.67e-03	2.31e-01
MX1	3.25e-02	8.88e-07	1.41e-01
USP18	1.83e-01	3.44e-03	5.22e-01
SSTR3	1.14e-01	6.94e-03	4.87e-01
GTPBP1	8.24e-01	6.14e-03	6.75e-01
ODF3B	6.06e-01	5.90e-01	2.23e-01
TLR7	9.56e-03	1.62e-05	1.87e-03

Supplementary Table 5: Direct associations between rs2401506 on chromosome 22 and genes with a PIP>0.5 in the type I interferon response component.

	rs12630796			rs17611866		
	Adipose	LCLs	Skin	Adipose	LCLs	Skin
SENP7	3.17e-06	4.02e-21	7.88e-02	7.74e-01	7.06e-01	1.57e-01
PCDHB14	9.13e-02	8.77e-01	1.25e-02	1.27e-01	6.82e-01	4.35e-01
MICA	4.48e-01	2.53e-01	7.39e-02	4.42e-01	5.11e-01	6.92e-01
ZNF250	3.84e-01	5.34e-01	1.76e-01	6.22e-01	6.00e-02	3.94e-01
CACNB2	2.91e-01	7.08e-01	2.96e-03	2.03e-01	3.89e-01	5.53e-02
ZNF263	7.09e-01	5.53e-03	1.32e-02	6.51e-18	5.45e-27	1.59e-16
TIGD7	1.86e-01	7.67e-01	2.50e-01	2.18e-36	2.39e-07	2.94e-07
ZNF559	3.60e-01	7.79e-01	6.25e-02	3.46e-01	2.33e-01	4.31e-01
ZNF562	3.41e-02	1.03e-01	2.13e-02	5.21e-01	6.65e-01	1.71e-03
ZNF491	9.50e-01	8.84e-01	6.27e-01	5.31e-01	5.00e-02	6.17e-02
ZNF440	9.84e-01	1.85e-01	2.96e-02	4.13e-01	4.74e-02	3.08e-04
ZNF439	7.64e-01	1.44e-03	5.35e-04	3.18e-02	1.46e-02	5.68e-07
ZNF788	6.83e-01	1.09e-01	3.54e-03	6.82e-04	2.19e-04	1.37e-06
AC022415.1	7.67e-01	8.92e-02	2.46e-03	1.16e-03	1.79e-04	7.24e-07
ZNF626	9.35e-01	1.17e-04	8.63e-02	3.16e-01	3.08e-01	2.65e-01
ZNF208	6.50e-01	5.35e-02	2.64e-02	2.12e-01	6.48e-01	3.94e-01
ZNF676	3.20e-01	1.16e-01	1.68e-01	9.34e-01	9.74e-01	8.85e-01
ZNF404	5.63e-01	9.10e-01	9.95e-01	2.26e-02	5.94e-01	2.46e-04
ZNF233	7.66e-01	1.14e-01	2.91e-01	1.69e-03	4.31e-01	1.61e-01
ZNF285	5.37e-01	1.40e-02	7.12e-01	8.48e-01	1.44e-02	4.85e-01
ZNF229	6.57e-01	1.59e-02	1.57e-03	8.65e-01	4.53e-02	3.25e-01
ZNF578	4.53e-01	1.44e-01	8.75e-03	3.69e-01	4.18e-02	1.92e-01
ZNF415	1.90e-01	8.80e-03	1.88e-04	5.84e-02	9.97e-01	1.05e-01
ZNF667	5.59e-01	2.36e-01	2.41e-05	9.37e-02	4.27e-08	6.46e-01
AC004696.1	6.87e-01	2.11e-02	2.55e-03	1.68e-01	1.76e-06	7.69e-01
ZIM2	3.62e-01	1.74e-03	7.61e-03	2.46e-01	9.24e-01	8.47e-01
ZNF264	1.08e-01	6.36e-01	1.28e-01	9.65e-01	7.78e-01	6.79e-01
ZNF543	2.00e-02	3.30e-01	2.03e-04	4.21e-01	1.62e-01	3.04e-01
ZNF304	7.09e-01	5.53e-02	7.69e-02	4.44e-03	1.53e-01	3.50e-01
ZNF547	2.47e-02	6.74e-04	6.70e-02	9.55e-01	1.36e-01	7.23e-02
VN1R1	1.33e-01	1.25e-01	1.90e-02	4.57e-02	3.69e-01	2.01e-02
ZNF772	2.62e-02	1.16e-02	2.45e-03	9.70e-02	3.90e-02	4.52e-02
ZNF549	1.00e-01	1.30e-01	3.26e-06	5.41e-02	1.92e-01	2.16e-03
AC003682.1	2.81e-01	1.32e-02	1.55e-01	3.97e-03	7.72e-01	1.26e-01
ZNF416	7.76e-01	7.29e-01	1.22e-01	1.59e-01	3.38e-01	2.08e-03
ZIK1	1.48e-02	1.31e-05	1.36e-04	2.63e-02	2.65e-01	6.58e-03
ZNF530	4.80e-01	8.52e-01	7.16e-02	3.42e-01	4.73e-04	6.29e-01
ZNF134	3.08e-01	1.73e-02	1.47e-05	1.04e-04	5.27e-01	6.60e-08
ZNF211	3.94e-01	7.94e-01	1.67e-08	5.99e-03	5.66e-01	3.93e-04
ZSCAN4	8.21e-01	9.76e-01	8.77e-08	3.44e-05	6.98e-01	2.06e-03
ZNF551	5.19e-03	7.71e-04	5.93e-03	5.14e-04	3.80e-01	3.86e-05
AC004017.1	3.50e-04	6.76e-05	4.20e-03	2.65e-04	1.49e-01	1.06e-05
ZNF154	5.94e-04	4.13e-08	1.71e-06	3.38e-14	9.68e-04	6.57e-07
ZNF671	6.09e-05	1.07e-04	2.09e-15	3.97e-10	5.22e-01	2.21e-10
ZNF776	2.34e-03	2.48e-04	2.88e-04	1.26e-03	1.38e-01	5.75e-04
ZNF814	2.64e-04	1.54e-06	4.30e-06	4.52e-02	1.54e-02	4.37e-04
ZNF417	2.06e-02	5.21e-02	4.92e-03	3.36e-02	4.30e-01	9.12e-02
ZNF418	3.44e-04	2.05e-10	3.29e-04	1.46e-05	7.78e-02	1.27e-02
ZNF256	3.09e-01	4.21e-02	8.43e-07	2.66e-03	5.64e-01	1.48e-01
ZNF606	6.43e-01	4.15e-01	2.25e-01	7.12e-01	8.14e-01	2.85e-01
CTD-2368P22.1	9.84e-01	4.13e-01	1.33e-01	1.60e-01	9.87e-01	2.25e-02
ZSCAN1	5.76e-01	8.82e-01	2.79e-02	5.94e-01	5.95e-01	8.95e-01
ZNF135	3.55e-01	7.82e-03	1.12e-04	4.28e-01	2.47e-02	6.65e-01
ZSCAN18	8.77e-01	2.60e-02	8.47e-05	8.10e-01	1.28e-02	8.67e-01
ZNF329	9.54e-01	8.57e-01	6.19e-02	1.63e-01	2.08e-03	1.12e-01
ZNF274	4.49e-02	1.77e-01	2.96e-08	6.88e-01	4.62e-03	1.82e-01
ZNF544	1.21e-01	4.08e-01	1.39e-02	3.72e-01	1.84e-04	4.60e-01

Supplementary Table 6: Direct associations between rs12630796 on chromosome 3 and rs17611866 on chromosome 16 and genes with a PIP>0.5 in the zinc finger network component. *SENP7* is also included in this table despite it having a near zero gene loading in the component.

	Tissue activation pattern							Row totals
	A	L	S	AL	AS	LS	ALS	
Number of components	57	74	70	0	14	1	20	236
SNP (1×10^{-10}) *cis*	1	1	0	0	0	0	18	20
SNP (1×10^{-10}) *trans*	0	3	0	0	1	0	2	6
Phenotype (1×10^{-6})	21	0	8	0	3	0	0	32
Sequencing (1×10^{-6})	37	53	39	0	2	0	0	131
GO term (1×10^{-6})	49	68	63	0	14	1	5	200

Supplementary Table 7: Summary of 236 robust components obtained when clustering components across 10 runs of the method. Components are categorized according to which set of tissues they are active in (A : Adipose L : LCLs, S : Skin) using a threshold of 0.5 on the tissue scores matrix. The first row of data gives the number of components with each activation pattern; subsequent rows summarize the number of component associated with SNPs, phenotypes, batch variables and enriched for GO terms (with significance levels given in brackets).

Supplementary Note

1 Bayesian Sparse Tensor Decomposition Model

1.1 Notation

Capital letters denote matrices; suppose Y is a matrix with dimensions $I \times J$. We reference the (i, j)th element of the matrix Y by y_{ij}. The ith row of Y is a row vector of length J denoted by $\mathbf{y}_{i\cdot}$ and the jth column of Y is a column vector of length I denoted by $\mathbf{y}_{\cdot j}$ or $\mathbf{y}_j$.

Tensors (by which we mean a 3-dimensional array) are represented by curly letters; for example, $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ is a tensor with dimensions I by J by K. An element of the tensor can be referenced using three indices, e.g. y_{ijk} for $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$ and $k \in \{1, \ldots, K\}$. We would also like to represent various subsets of the data in the tensor. Let us consider an example; suppose that $\mathcal{Y} \in \mathcal{R}^{N \times L \times T}$ contains gene expression data for N individuals, L genes and T tissues. A 2-dimensional slice of the tensor can be obtained by specifying one index, for example, the data for tissue t is a matrix given by $Y_{\cdot\cdot t} \in \mathcal{R}^{N \times L}$; similarly, all the data for an individual n is given by the matrix $Y_{n\cdot\cdot} \in \mathcal{R}^{L \times T}$. The tensor analog of a matrix row or column is the vector obtained when two indices are specified, for example, $\mathbf{y}_{\cdot lt} \in \mathcal{R}^N$ represents the data (for all individuals) for gene l in tissue t.

In the following sections, we use c and k as an index over components, n as an index over individuals, l as an index over genes and t as an index over tissues.

1.2 Model description

Let $\mathcal{Y} \in \mathbb{R}^{N \times L \times T}$ be a tensor containing expression data for N individuals at L genes in T tissues. For now we will assume that there is no missing data. (In section 1.10 we describe extensions to the model to deal with missing tissue samples and randomly missing elements in the tensor.) We also assume that the data for each gene (in each tissue) has been mean centred and variance normalised (i.e. $\mathbf{y}_{\cdot lt}$ has zero mean and unit variance).

We aim to find an alternative representation of the data in terms of C latent components. Specifically, the data is modelled as a linear combination of C components and additive noise,

$$y_{nlt} = \sum_{c=1}^{C} a_{nc} b_{tc} x_{cl} + \epsilon_{nlt}, \tag{1}$$

where $X \in \mathbb{R}^{C \times L}$ is a gene loadings (or scores) matrix; each row of X defines the relative contribution of each measured gene in the component. A is an N by C matrix of individual scores which describes the individual specific mixing weights for each component. Similarly, the tissue scores matrix $B \in \mathbb{R}^{T \times C}$ contains tissue specific mixing weights. Finally, $\mathcal{E}$ is an N by L by T tensor of noise. This model is known in the literature as the PARAFAC (parallel factors) decomposition or CANDECOMP (canonical decomposition) (Carroll and Chang, 1970; Harshman and Lundy, 1994).

We fit the model in a Bayesian framework; the next section defines priors for the model, including a shrinkage prior on X.

1.3 Priors

Sparsity prior on the gene loadings matrix

We expect biological processes to only involve a relatively small subset of the total number of genes in the genome and therefore want to identify components with sparse loadings vectors. In order to encourage sparsity in the model, we use a spike and slab prior on the gene loadings matrix (Lucas et al., 2006; Mitchell and Beauchamp, 1988). The spike and slab distribution consists of a mixture of a point mass at zero (the "spike") and a Gaussian (the "slab"). Genes involved in the component will have gene loadings modelled by the Gaussian distribution, while genes with zero effect should be captured by the delta function.

The prior on x_{cl} is given by

$$P(x_{cl}|p_{cl}, \beta_c) = p_{cl}\mathcal{N}(x_{cl}|0, \beta_c^{-1}) + (1 - p_{cl})\delta_0(x_{cl}), \tag{2}$$

where p_{cl} is a mixing weight and β_c is the precision of the Gaussian (Lucas et al., 2006). This prior is a more general alternative to the original spike and slab distribution from Mitchell and Beauchamp (1988) where a single mixing parameter is specified for each component i.e. $p_{cl} = p_c$. As in Lucas et al. (2006) we found that this more general spike and slab allowed us to model sparser signals in the data and resulted in lower false positive rates.

Following convention, a gamma prior is placed on the precision parameters $\beta_c \sim \mathcal{G}(\beta_c|e, f)$ where e and f are hyper-parameters. The prior on p_{cl} is given by

$$p_{cl} \sim \rho_c\mathcal{B}(p_{cl}|g, h) + (1 - \rho_c)\delta_0(p_{cl}) \tag{3}$$

where ρ_c is component-level mixing parameter. p_{cl} encodes sparsity of the (c, l)th element in the loadings matrix. A spike and slab prior on p_{cl}, with a Beta distribution for the slab, reflects our belief that some elements in the loadings matrix should be zero, and others should have an non-zero value. If ρ_c takes a value close to 0, then the expression for p_{cl} will be dominated by the delta function at zero and the majority of the loadings vector will take values close to, or equal to, zero, resulting in a sparse component. We learn ρ_c alongside the other parameters. To complete the prior on x_{cl} we place a beta distribution on $\rho_c \sim \mathcal{B}(\rho_c|r, z)$ with hyperparameters r and z.

In order to make inference easier, we follow the approach used in Titsias and Lázaro-Gredilla (2011) and factorise the spike and slab distribution as $x_{cl} = w_{cl}s_{cl}$ where

$$w_{cl} \sim \mathcal{N}(w_{cl}|0, \beta_c^{-1}), \tag{4}$$

$$s_{cl} \sim \mathcal{B}\text{ernoulli}(s_{cl}|p_{cl}). \tag{5}$$

The random variable s_{cl} reflects the model's evidence that the (c, l)th element of X is non-zero. The magnitude of w_{cl} can be thought of as an effect size for the (c, l)th element.

We use the same trick to make inference on p_{cl} tractable; let $p_{cl} = \psi_{cl}\phi_{cl}$ where

$$\psi_{cl} \sim \mathcal{B}(\psi_{cl}|g, h), \tag{6}$$

$$\phi_{cl} \sim \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c). \tag{7}$$

Prior on the individual and tissue scores matrices

We put a standard multivariate normal prior on the component vectors in both scores matrices.

$$\begin{aligned} P(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c}|0, I_N), \\ P(\mathbf{b}_{\cdot c}) &= \mathcal{N}_T(\mathbf{b}_{\cdot c}|0, I_T). \end{aligned} \tag{8}$$

This corresponds to a prior belief that individuals and tissues are independent. Without loss of generality, we can fix the variance of these distributions to 1, because of the scaling indeterminacy of factor analysis models. A scaling can be incorporated into the precision parameters (β_c) in the gene loadings matrix.

Prior on noise precision

To complete the model specification, we use a Gaussian error term. The noise levels for each gene are modelled independently, where λ_{lt} is the noise precision for each gene and tissue combination,

$$\boldsymbol{\epsilon}_{\cdot lt} \sim \mathcal{N}_N(\boldsymbol{\epsilon}_{\cdot lt}|0, \lambda_{lt}^{-1} I_N). \tag{9}$$

The precision parameters are given a Gamma distribution with hyper-parameters u and v,

$$\lambda_{lt} \sim \mathcal{G}(\lambda_{lt}|u, v). \tag{10}$$

1.4 Full model

The full model can be written as

$$\begin{aligned} P(\mathcal{Y}|\theta) &= \prod_{lt} \mathcal{N}_N\Big(\mathbf{y}_{\cdot lt}|\sum_c \mathbf{a}_{\cdot c} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1} I_N\Big) \\ P(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c}|0, I_N) \\ P(\mathbf{b}_{\cdot c}) &= \mathcal{N}_N(\mathbf{b}_{\cdot c}|0, I_T) \\ P(w_{cl}|\beta_c) &= \mathcal{N}\big(w_{cl}|0, \beta_c^{-1}\big) \\ P(s_{cl}|\psi_{cl}, \phi_{cl}) &= \mathcal{B}\text{ernoulli}\big(s_{cl}|\psi_{cl}, \phi_{cl}\big) \\ P(\beta_c) &= \mathcal{G}(\beta_c|e, f) \\ P(\psi_{cl}) &= \mathcal{B}\text{eta}(\psi_{cl}|g, h) \\ P(\phi_{cl}|\rho_c) &= \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c) \\ P(\rho_c) &= \mathcal{B}\text{eta}(\rho_c|r, z) \\ P(\lambda_{lt}) &= \mathcal{G}(\lambda_{lt}|u, v) \end{aligned} \tag{11}$$

where θ denotes the set of all parameters.

1.5 Hyperparameters

We place uninformative priors on the noise precision, λ_{lt}, and the 'slab' precision β_c by setting $u = 10^{-6}$, $v = 10^6$, $e = 10^{-6}$ and $f = 10^6$. We put a flat (uniform) prior on the component sparsity parameters (ρ_c) by setting $r = z = 1$. To encourage sparsity in the gene loadings we use a prior on ψ_{cl} with $g = h = 0$.

1.6 Inference via variational Bayes

Inference is performed using an Bayesian technique called variational Bayes (VB) which allows us to evaluate an approximation to the posterior distribution. Suppose the approximate posterior distribution is given by $Q(\theta)$. VB aims to minimise the Kullbeck-Lieber (KL) divergence between $Q(\theta)$ and the true posterior $P(\theta|\mathcal{Y})$ given by

$$\mathrm{KL}(Q|P) = \int Q(\theta) \log \frac{Q(\theta)}{P(\theta|\mathcal{Y})} d\theta. \tag{12}$$

The KL divergence takes positive values, or a value of 0 if and only if $Q(\theta)$ is identical to $P(\theta|\mathcal{Y})$. We can write the marginal log-likelihood in terms of the KL divergence and a term called the negative free energy denoted by $F(Q)$,

$$\log P(\mathcal{Y}) = \underbrace{\int Q(\theta) \log \frac{P(\mathcal{Y}, \theta)}{Q(\theta)} d\theta}_{:=F(Q)} + \mathrm{KL}(Q|P). \tag{13}$$

Minimising the KL divergence is equivalent to maximising $F(Q)$. Also note that because the KL divergence can not take a negative value, $F(Q)$ is a lower bound to the log-marginal likelihood.

A common approach to optimising $F(Q)$ is the mean field VB algorithm where the approximate posterior is assumed to fully factorise. If the model priors are chosen to be conjugate, then (conditional) analytic solutions can be obtained for the variational parameters that increase $F(Q)$ and the algorithm consists of iteratively updating parameters until convergence. An alternative approach can be used if the priors are not conjugate; in this scenario, a fixed form for the posterior distribution is used and the parameters of this approximate distribution are estimated in order to maximise the negative free energy.

The majority of the parameters in our model have conjugate priors. However, in some cases, fully factorising over parameters may be too strong an assumption and also an unnecessary assumption. We retain dependence between w_{cl} and s_{cl}, rather than assuming they are independent (Titsias and Lázaro-Gredilla, 2011). Titsias and Lázaro-Gredilla (2011) show that this approach results in more robust and accurate estimates. All other parameters with conjugate priors are assumed to fully factorise in the approximate posterior distribution.

Unfortunately the parameters ψ_{cl}, ϕ_{cl} and ρ_c do not have conjugate priors and we are not able to use the results from mean field VB. Instead, we specify that their posterior distributions are point masses and optimise the free energy to find these point estimates.

The approximate posterior distribution $Q(\theta)$ for the model takes the following form

$$Q(\theta) = \prod_c Q(\mathbf{a}_{\cdot c}) \prod_{t,c} Q(b_{tc}) \prod_{c,l} Q(w_{cl}|s_{cl}) Q(s_{cl}) \prod_c Q(\beta_c) \prod_{c,l} \delta_{\psi^*_{cl}}(\psi_{cl}) \prod_{c,l} \delta_{\phi^*_{cl}}(\phi_{cl}) \prod_c \delta_{\rho^*_c}(\rho_c) \prod_{l,t} Q(\lambda_{lt}) \tag{14}$$

Our VB algorithm consists of iteratively updating each parameter given current estimates of the other parameters. All updates are guaranteed to increase (or at least not decrease) the negative free energy. All parameters are initialised randomly from their prior distribution other than parameters s_{cl}, ψ_{cl} and ϕ_{cl}, which are initialised to 0.5.

1.6.1 Variational Bayes updates

Parameters of the approximate posterior distributions are denoted using an asterix (*).

Loadings matrix

$$\begin{aligned} Q(w_{cl}|s_{cl}) &= \mathcal{N}\Big(w_{cl}\Big|s_{cl}m^*_{cl}, \big(s_{cl}\sigma^*_{cl} + (1-s_{cl})\langle\beta_c\rangle\big)^{-1}\Big) \\ \sigma^*_{cl} &= \langle\beta_c\rangle + \sum_{nt}\langle\lambda_{lt}\rangle\langle a^2_{nc}\rangle\langle b^2_{tc}\rangle \\ m^*_{cl} &= \sigma^{*-1}_{cl}\big(\sum_{n,t}\langle\lambda_{lt}\rangle y_{nlt}\langle a_{nc}\rangle\langle b_{tc}\rangle - \sum_{n,t}\langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle b_{tc}\rangle\sum_{k\neq c}\langle w_{kl}s_{kl}\rangle\langle a_{nk}\rangle\langle b_{tk}\rangle\Big) \end{aligned} \tag{15}$$

$$\begin{aligned} Q(s_{cl}) &= \mathcal{B}\text{ernoulli}(s_{cl}|\gamma^*_{cl}) \\ \gamma^*_{cl} &= \frac{1}{1+e^{-u^*_{cl}}} \\ u^*_{cl} &= \log(\psi^*_{cl}\phi^*_{cl}) - \frac{1}{2}\log\sigma^*_{cl} + \frac{\sigma^*_{cl}}{2}{m^*_{cl}}^2 - \log(1-\psi^*_{cl}\phi^*_{cl}) + \frac{1}{2}\log\langle\beta_c\rangle \end{aligned} \tag{16}$$

Sparsity parameters

We derive point estimates for the parameters ψ_{cl}, ϕ_{cl} and ρ_c by directly optimising the negative free energy. The relevant terms of the negative free energy are given by $\tilde{F}$.

$$\begin{aligned} \tilde{F} &:= \sum_{c,l}\log P(s_{cl}|\psi_{cl},\phi_{cl}) + \sum_{c,l}\log P(\psi_{cl}) + \sum_{c,l}\log P(\phi_{cl}|\rho_c) + \sum_c \log P(\rho_c) \\ &= \sum_{c,l}\big(\langle s_{cl}\rangle\log(\psi_{cl}\phi_{cl}) + \langle 1-s_{cl}\rangle\log(1-\psi_{cl}\phi_{cl})\big) \\ &\quad + \sum_{c,l}\big((g-1)\log\psi_{cl} + (h-1)\log(1-\psi_{cl})\big) \\ &\quad + \sum_{c,l}\big(\phi_{cl}\log\rho_c + (1-\phi_{cl})\log(1-\rho_c)\big) \\ &\quad + \sum_c\big((r-1)\log\rho_c + (z-1)\log(1-\rho_c)\big) \end{aligned} \tag{17}$$

The equation $\frac{\delta F}{\delta\rho_c} = 0$ has a closed form solution so we can find ρ^*_c as follows,

$$\rho^*_c = \frac{\sum_l \phi^*_{cl} + r - 1}{L + r + z - 2} \tag{18}$$

Since we expect ψ_{cl} and ϕ_{cl} to be highly coupled, we use Newton's method to simultaneously find $(\psi^*_{cl}, \phi^*_{cl})$ to optimise $\tilde{F}$. The optimisation problem we need to solve is

$$(\psi^*_{cl}, \phi^*_{cl}) = \text{argmax}_{(\psi_{cl}, \phi_{cl})} \tilde{F} \tag{19}$$

The gradient and Hessian matrix of $\tilde{F}$ are given by

$$\mathbf{g} = \begin{pmatrix} \frac{\langle s_{cl} \rangle}{\psi_{cl}} - \frac{\langle 1 - s_{cl} \rangle \phi_{cl}}{1 - \psi_{cl}\phi_{cl}} + \frac{g-1}{\psi_{cl}} - \frac{h-1}{1-\psi_{cl}} \\ \frac{\langle s_{cl} \rangle}{\phi_{cl}} - \frac{\langle 1 - s_{cl} \rangle \psi_{cl}}{1 - \psi_{cl}\phi_{cl}} + \log \rho_c - \log(1 - \rho_c) \end{pmatrix} \tag{20}$$

$$H = \begin{pmatrix} -\frac{\langle s_{cl} \rangle}{\psi^2_{cl}} - \frac{\langle 1 - s_{cl} \rangle \phi^2_{cl}}{(1 - \psi_{cl}\phi_{cl})^2} - \frac{qr-1}{\psi^2_{cl}} - \frac{q(1-r)-1}{(1-\psi_{cl})^2} & -\frac{\langle 1 - s_{cl} \rangle}{(1 - \psi_{cl}\phi_{cl})^2} \\ -\frac{\langle 1 - s_{cl} \rangle}{(1 - \psi_{cl}\phi_{cl})^2} & -\frac{\langle s_{cl} \rangle}{\phi^2_{cl}} - \frac{\langle 1 - s_{cl} \rangle \psi^2_{cl}}{(1 - \psi_{cl}\phi_{cl})^2} \end{pmatrix} \tag{21}$$

We update (ψ_{cl}, ϕ_{cl}) as follows,

$$\begin{pmatrix} \psi^{i+1}_{cl} \\ \phi^{i+1}_{cl} \end{pmatrix} = \begin{pmatrix} \psi^{i}_{cl} \\ \phi^{i}_{cl} \end{pmatrix} - \alpha H^{i-1} \mathbf{g}^i \tag{22}$$

where α is a step-size determined using a backtracking line search, i.e. we start with $\alpha = 1$ then reduce α until we satisfy $\tilde{F}^{i+1} > \tilde{F}^i$

Update for $\mathbf{a}_c$

$$\begin{aligned} Q(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c} | \boldsymbol{\mu}^*_c, \Omega^{*-1}_c) \\ \Omega^*_c &= (1 + \sum_{l,t} \langle \lambda_{lt} \rangle \langle b^2_{tc} \rangle \langle w^2_{cl} s^2_{cl} \rangle) I_N \\ \boldsymbol{\mu}^*_c &= \Omega^{*-1}_c \Big(\sum_{l,t} \langle \lambda_{lt} \rangle \mathbf{y}_{\cdot lt} \langle b_{tc} \rangle \langle w_{cl} s_{cl} \rangle - \sum_{l,t} \langle \lambda_{lt} \rangle \langle b_{tc} \rangle \langle w_{cl} s_{cl} \rangle \sum_{k \neq c} \langle \mathbf{a}_{\cdot k} \rangle \langle b_{tk} \rangle \langle w_{kl} s_{kl} \rangle \Big) \end{aligned} \tag{23}$$

Update for b_{tc}

$$\begin{aligned} Q(b_{tc}) &= \mathcal{N}(b_{tc} | \nu^*_{tc}, \tau^{*-1}_{tc}) \\ \tau^*_{tc} &= 1 + \sum_{n,l} \langle \lambda_{lt} \rangle \langle a^2_{nc} \rangle \langle w^2_{cl} s^2_{cl} \rangle \\ \nu^*_{tc} &= \tau^{*-1}_{tc} \Big(\sum_{n,l} \langle \lambda_{lt} \rangle y_{nlt} \langle a_{nc} \rangle \langle x_{cl} \rangle - \sum_{n,l} \langle \lambda_{lt} \rangle \langle a_{nc} \rangle \langle w_{cl} s_{cl} \rangle \sum_{k \neq c} \langle a_{nk} \rangle \langle b_{tk} \rangle \langle w_{kl} s_{kl} \rangle \Big) \end{aligned} \tag{24}$$

Update for β_c

$$\begin{aligned} Q(\beta_c) &= \mathcal{G}(e^*_c, f^*_c) \\ e^*_c &= e + \frac{L}{2} \\ f^*_c &= \Big(\frac{1}{f} + \frac{1}{2} \sum_l \langle w^2_{cl} \rangle \Big)^{-1} \end{aligned} \tag{25}$$

Update for λ_{lt}

$$Q(\lambda_{lt}) = \mathcal{G}(\lambda_{lt}|u^*_{lt}, v^*_{lt}) \tag{26}$$

$$u^*_{lt} = u + \frac{NT}{2}$$

$$v^*_{lt} = \left(\frac{1}{v} + \frac{1}{2}\sum_n \left\langle \left(y_{nlt} - \sum_c a_{nc} b_{tc} w_{cl} s_{cl}\right)^2 \right\rangle\right)^{-1} \tag{27}$$

1.6.2 Negative free energy

The negative free energy is a lower bound of the model evidence (marginal likelihood). The updates given above are guaranteed to increase the free energy.

$$\begin{aligned}
F(Q) = &-\frac{NLT}{2}\log 2\pi + \frac{N}{2}\sum_{l,t}\langle \log \lambda_{lt}\rangle - \frac{1}{2}\sum_{n,l,t}\langle\lambda_{lt}\rangle\big\langle (y_{nlt} - \Sigma_c a_{nc} b_{tc} w_{cl} s_{cl})^2\big\rangle \\
&- \frac{1}{2}\sum_c \langle \mathbf{a}_{\cdot c}^\top \mathbf{a}_{\cdot c}\rangle - \frac{1}{2}\sum_c \log|\Omega^*_c| + \frac{NC}{2} \\
&- \frac{1}{2}\sum_{t,c}\langle b^2_{tc}\rangle - \frac{1}{2}\sum_{t,c}\log|\nu_{tc}| + \frac{TC}{2} \\
&+ \frac{L}{2}\sum_c \big\langle \log \beta_c\big\rangle + \frac{CL}{2} - \frac{1}{2}\sum_{c,l}\langle\beta_c\rangle\big\langle w^2_{cl}\big\rangle \\
&- \frac{1}{2}\sum_{c,l}\gamma^*_{cl}\log\sigma^*_{cl} + \frac{1}{2}\sum_{c,l}(1-\gamma^*_{cl})\log\langle\beta_c\rangle \\
&\sum_c \Bigg(-\log\Gamma(e) - e\log f + (e-1)(\psi(e^*_c) + \log\hat{f}_c) - \frac{e^*_c f^*_c}{f} \\
&+ e^*_c + \log f^*_c + \log\Gamma(e^*_c) - (e^*_c - 1)\psi(e^*_c)\Bigg) \\
&+ \sum_{c,l}\Bigg(\langle s_{cl}\rangle\langle\log\psi_{cl}\phi_{cl}\rangle + (1-\langle s_{cl}\rangle)\langle\log\big(1-\psi_{cl}\phi_{cl}\big)\rangle \\
&- \langle s_{cl}\rangle\log\langle s_{cl}\rangle - (1-\langle s_{cl}\rangle)\log(1-\langle s_{cl}\rangle)\Bigg) \\
&+ \sum_{c,l}\Big((g-1)\log\psi^*_{cl} + (h-1)\log(1-\psi^*_{cl})\Big) \\
&+ \sum_{cl}\Big(\phi^*_{cl}\log\rho^*_c + (1-\phi^*_{cl})\log(1-\rho^*_c)\Big) \\
&+ \sum_{cl}\Big((r-1)\log\phi^*_{cl} + (z-1)\log(1-\phi^*_{cl})\Big) \\
&+ \sum_{lt}\Bigg(-\log\Gamma(u) - u\log v + (u-1)(\psi(u^*_{lt}) + \log v^*_{lt}) - \frac{u^*_{lt}v^*_{lt}}{v} \\
&+ u^*_{lt} + \log v^*_{lt} + \log\Gamma(u^*_{lt}) - (u^*_{lt}-1)\psi(u^*_{lt})\Bigg)
\end{aligned} \tag{28}$$

1.7 Identifiability

The components estimated by our model are not completely identifiable. The sign of the gene loadings, individual scores and tissue scores is not fully determined by the model, so that swapping the sign of any two of these parts of the model will produce an equivalent model

fit. Scaling of the components is constrained to some extent by the fixed unit variances used in the priors on the individual and tissue scores. The use of sparsity in the gene loadings goes some way to ensure that components are not rotationally invariant, but dense factors will clearly suffer from this problem, especially if they are active in only one tissue.

1.8 Implementation and complexity

We implement the model in `C++` using a matrix library called Eigen (`http://eigen.tuxfamily.org`). The complexity of the algorithm is $\mathcal{O}(NLTC^2)$ but parallelisation of matrix multiplications (via Eigen) and parallel updates of elements in the gene loadings matrix (via openmp) speed up the code considerably.

1.9 Convergence

Based on experience of running this method on simulated and real data, we run the method for 3,000 iterations. We check for convergence by tracking the change in $\langle S \rangle$. After 3,000 iterations, the average number of elements in $\langle S \rangle$ that cross the threshold 0.5 drops to less than 1 per iteration.

1.10 Handling missing data

In this section we describe two extensions to the model which allow for missing data. We consider two scenarios in which missing data might arise, missing tissue samples and randomly missing data points.

1.10.1 Missing tissue samples

Missing tissue samples arise when only a subset of the tissues are collected for a particular individual, or if data for a whole tissue sample is removed due to experimental errors. Missing samples correspond to missing vectors within the data tensor; for example, if data for individual n in tissue t is missing, then $\mathbf{y}_{n \cdot t}$ will be missing. We can reformulate our model to ignoring these missing samples.

Let $\mathcal{J}$ be a binary indicator matrix of dimensions N by T, where $\mathcal{J}_{nt} = 1$ if data for individual n in tissue t exists and $\mathcal{J}_{nt} = 0$ otherwise. Based on the data that does exist, the likelihood is given by,

$$P(\mathcal{Y}|\theta) = \prod_{n,l,t} \mathcal{N}\Big(y_{nlt}|\Sigma_c a_{nc} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1}\Big)^{\mathcal{J}_{nt}} \tag{29}$$

Using this likelihood and the priors defined in section 1.3, we can derive updates for the model parameters in a similar way as above. The resulting updates are identical to those given in section 1.6.1 except that the indicator matrix $\mathcal{J}$ needs to be added into any expression with a sum over n or t.

1.10.2 Missing data points

We will now briefly describe how to deal with randomly missing elements in the data tensor that may have arisen due to experimental errors. In this scenario, we treat the missing data points as parameters in the model and learn their posterior distribution.

We create a partition of the data such that $\mathcal{Y} = \mathcal{Y}^o \cup \mathcal{Y}^m$ where $\mathcal{Y}^o$ denotes the set of observed data and $\mathcal{Y}^m$ denotes the set of missing data. Let $\mathcal{S}^m$ be the set of triplets $\{n, l, t\}$ for which data is missing.

The prior for the missing data is,

$$P(\mathcal{Y}^m|\theta) = \prod_{\{n,l,t\}\in\mathcal{S}^m} \mathcal{N}\Big(y^m_{nlt}|\Sigma_c a_{nc} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1}\Big), \tag{30}$$

and assuming that the posterior factorises fully, the posterior is given by

$$Q(\mathcal{Y}^m) = \prod_{\{n,l,t\}\in S^m} \mathcal{N}(y^m_{nlt}|\Sigma_c \langle a_{nc}\rangle \langle b_{tc}\rangle \langle w_{cl} s_{cl}\rangle, \langle \lambda_{lt}^{-1}\rangle). \tag{31}$$

With this extension, the updates for the other model parameter are similar to those given in section 1.6.1, altered to reflect the uncertainty in the estimates of the missing data points, if $\{n, l, t\} \in S^m$, we need to replace y_{nlt} by $\langle y_{nlt}\rangle$ and y^2_{nlt} by $\langle y^2_{nlt}\rangle = \langle y_{nlt}\rangle^2 + \langle \lambda_{lt}^{-1}\rangle$.

1.11 Allowing for related individuals

As it stands, our model ignores any relatedness between samples. However genetic studies often contain closely related individuals by design, or distantly related individuals by chance when recruitment occurs within a small geographical area. A kinship matrix $K \in \mathbf{R}^{N\times N}$ can be used to summarise this genetic relatedness between individuals where an element of K, k_{ij}, is a measure of the relatedness between individual i and individual j. Data from related individuals are likely to be correlated due to shared genetic material and explicitly modelling these correlations may lead to better results.

Our model identifies both genetic structure (e.g. a *trans* network or the genetic basis of ageing) and non-genetic structure (e.g. environment signals or batch effects) in the data. We can accommodate these different types of components by using the following prior on the individual scores matrix A, using the kinship matrix to inform the model about the relatedness between individuals,

$$A \sim \prod_c \mathcal{N}_N(\mathbf{a}_{\cdot c}|0, \alpha_c K + (1-\alpha_c) I_N), \tag{32}$$

where the covariance of each scores vector is a mixture of the kinship matrix and the identity matrix with a different mixing parameter α_c for each component. If α_c is close to 1 then the covariance matrix in the prior is approximately the kinship matrix K, which imposes a structure so that related individuals have more similar scores, resulting in a 'genetic' component. On the other hand, if α_c is close to 0 then the prior has no genetic basis and we recover the i.i.d Gaussian prior already described for A. The mixing parameter α_c is given an uninformative Beta prior,

$$\alpha_c \sim \mathcal{B}\text{eta}(\alpha_c|1, 1). \tag{33}$$

Implementing this model involves a change to the update for A and the addition of an update for α_c but the remaining parameter updates remain the same. We assume that $\mathbf{a}_{\cdot c}$ and α_c are independent in the approximate posterior distribution and that the posterior distribution of α_c is a delta function at α_c^*.

The update for A becomes,

$$
\begin{aligned}
Q(\mathbf{a}_{\cdot c}) &= \mathcal{N}_N(\mathbf{a}_{\cdot c}|\boldsymbol{\mu}_c^*, \Omega_c^{*-1}) \\
\Omega_c^* &= \Big(\alpha_c^* K + (1-\alpha_c^*) I_N\Big)^{-1} + \big(\sum_{l,t} \langle\lambda_{lt}\rangle\langle b_{tc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle\big) I_N \\
\boldsymbol{\mu}_c^* &= \Omega_c^{*-1}\Big(\sum_{l,t}\langle\lambda_{lt}\rangle \mathbf{y}_{\cdot lt}\langle b_{tc}\rangle\langle w_{cl}s_{cl}\rangle - \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}\rangle\langle w_{cl}s_{cl}\rangle \sum_{k\neq c}\langle\mathbf{a}_{\cdot k}\rangle\langle b_{tk}\rangle\langle w_{kl}s_{kl}\rangle\Big) \quad (34)
\end{aligned}
$$

An efficient implementation of this can be obtained by using the eigendecomposition of K to avoid inverting an N by N matrix in the calculation for Ω_c^*. In fact, we can avoid calculating Ω_c^* altogether as the expression for $\boldsymbol{\mu}_c$ only requires Ω_c^{*-1}. Using the eigendecomposition of $K = QDQ^t$ where Q is an orthonormal matrix of eigenvectors and D is a diagonal matrix with eigenvalues on the diagonal. We can now write,

$$
(\Omega_c^{*-1})_{nm} = Q\Big(((1-\alpha_c^*)I_N + \alpha_c^* D)^{-1} + \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle I_N\Big)^{-1} Q^t. \quad (35)
$$

Using the above expression, the complexity in N scales quadratically. We note that this expression will no longer apply when there are missing samples in the data. The combination of the genetic prior and missing samples makes the complexity cubic in N which is why we do not use this approach when analysing the TwinsUK data set.

We evaluate the point estimates α_c^* using gradient ascent,

$$
\alpha_c^* \leftarrow \alpha_c^* + \Delta \sum_n (-1 + D_{nn})\Big(-\frac{1}{1-\alpha_c^* + \alpha_c^* D_{nn}} + \frac{\big(Q(\mu_c^*\mu_c^{*t} + \Omega_c^{*-1})\big)_{nn}}{(1-\alpha_c^* + \alpha_c^* D_{nn})^2}\Big), \quad (36)
$$

where $\Delta = 0.0001$ is the step size.

1.12 Linked matrix/tensor decomposition

The 3D tensor decomposition method that we have described above is actually a special case of a more general model we have implemented for linked tensor decomposition (see Supplementary Figure 38). Consider a study consisting of D types of omics data for a set of N individuals. Let each data set d be represented by the tensor, $\mathcal{Y}^{(d)} \in \mathbb{R}^{N\times L_d \times T_d}$ where L_d is the number of variables measured for data type d and T_d is the number of contexts (or conditions) in which these variables were measured. If data for type d is collected in only a single context then $T_d = 1$. Importantly, all tensors are linked by their shared first dimension (N).

The data is modelled as follows (Groves et al., 2011),

$$
y_{nlt}^{(d)} = \sum_c a_{nc} b_{tc}^{(d)} x_{cl}^{(d)} + \epsilon_{nlt}^{(d)} \qquad \text{for } d \in \{1, \cdots, D\} \quad (37)
$$

where $A \in \mathbb{R}^{N\times C}$ is the individual scores matrix (shared across all data types), $B^{(d)} \in \mathbb{R}^{T_d \times C}$ is a context specific scores matrix for data type d and $X^{(d)} \in \mathbb{R}^{C\times L_d}$ is a loadings matrix for data type d. A noise tensor for each data type is given by $\mathcal{E}^{(d)} \in \mathbb{R}^{N\times L_d \times T_d}$. Each data tensor is decomposed using equation (1), with the constraint that a single individual scores matrix is common across all data types. In practice, if $T_d = 1$ for a data type d, then $B^{(d)}$ has dimensions 1 by C and is fixed to a vector of ones during inference.

Again, spike and slab priors are used for the loadings matrices to encourage sparsity. Updates for the loadings and context scores matrices for a data type d are effectively identical to the (single) tensor decomposition already considered. Importantly, updates for $X^{(d)}$ and $B^{(d)}$ do not depend on $X^{(d')}$ and $B^{(d')}$ for any $d' \neq d$. The update for A is dependent on all other current parameter estimates. One way to think about this update is that it averages over the estimates for A that one would get if performing separate decompositions for each data type. (In reality this is not quite the case because the prior also needs to be considered.)

It is important to note that equation (37) can model a variety of different types of underlying structure in the data. Components can be shrunk to zero for a particular data type allowing for the model to capture signals that exist in an arbitrary subset of the data. For example, in Supplementary Figure 38, the yellow component is active in only data types 2 and 3.

This linked tensor decomposition is a generalisation of several models. In particular, the single 3D tensor decomposition we focus on in this paper is recovered if $D = 1$. If $T_1 = D = 1$, then the model collapses to sparse factor analysis. Group factor analysis is recovered if $T_d = 1$ for all d.

2 Additional results

2.1 Marginal association for SNPs and gene identified in components

In order to better visualize the marginal associations for the lead SNPs from our components and the genes in our network (given in Supplementary Tables 2-6) we produced plots of the p-values. These plots are shown in Figures 1-5. On each plot we detail 3 different significance thresholds:

1. $3.3 \times 10^{-12} = 5 \times 10^{-8}/(3 \times 5000)$ - derived by taking an anti-conservative view, that due to correlation between genes there are only effectively only 5,000 genes.
2. $4.9 \times 10^{-11} = 0.05/(3 \times 18409 \times 18409)$ - derived by assuming that the best *cis* SNP for each gene was tested against all other genes
3. $9.05 \times 10^{-13} = 5 \times 10^{-8}/(3 \times 18409)$ - the Bonferroni correction of all SNPs versus all genes in all 3 tissues.

Table 1 shows the p-values for association between our components and lead SNPs, and also the best marginal association (in *trans*) between the SNP and set of genes identified in the component. In 5 out of 7 cases, our approach results in smaller p-values than the marginal associations.

In addition to performing marginal associations between the lead SNPs from our components and the genes identified in the network, we tested for associations between the lead SNP and all genes in the analysis (Figures 6 - 12). Genes identified in our components are shown in red. These plots suggest that our components are recovering the majority (if not all) of the genes involved in the *trans* effect. The genes with significant or near significant marginal associations are those identified in the components; an exception is *SENP7* which is significantly associated with rs12630796 in skin, but does not appear in our ZNF gene network component.

2.2 Discussion of direct associations for Zinc finger component

Supplementary Table 6 details the direct associations of SNPs rs12630796 and rs17611866 with *SENP7* on chromosome 3 and genes with non-zero gene loadings in the component in all three tissues. This analysis partially recovers the signal that we find using our method. rs12630796 on chromosome 3 is significantly associated with *ZNF671* on chromosome 19 in Skin (p-value = 2.0910-15) and there is some evidence of association in Adipose and LCLs (6.0910-5 and 1.0710-4 respectively) and associated with *ZNF418* (also on chromosome 19) in LCLs (p-value = 2.0510-10). However, the SNP varies considerably in its association with *SENP7* across tissues (LCLs p-value = 4.02×10^{-21}, Adipose p-value = 3.17×10^{-6}, Skin p-value = 7.88×10^{-2}). SNP rs17611866 on chromosome 16 shows a clear pattern of significant association with nearby gene *ZNF263* in all 3 tissues (p-values between 1.59×10^{-16} and 5.45×10^{-27}) and *T1GD7* (p-values between 2.94×10^{-7} and 2.18×10^{-36}) and a strong pattern of association with *ZNF671* on chromosome 19 in Skin (p-value = 2.21×10^{-10}) and Adipose tissue (p-value = 3.97×10^{-10}). Additionally, rs17611866 is associated with *ZNF154* (chromosome 19) in Adipose (p-value = 3.38×10^{-14}). This marginal analysis uncovers evidence of links between SNPs on chromosomes 3 and 16 with genes on chromosomes 3, 16 and 19, although not all of these associations reach a genome wide significance level

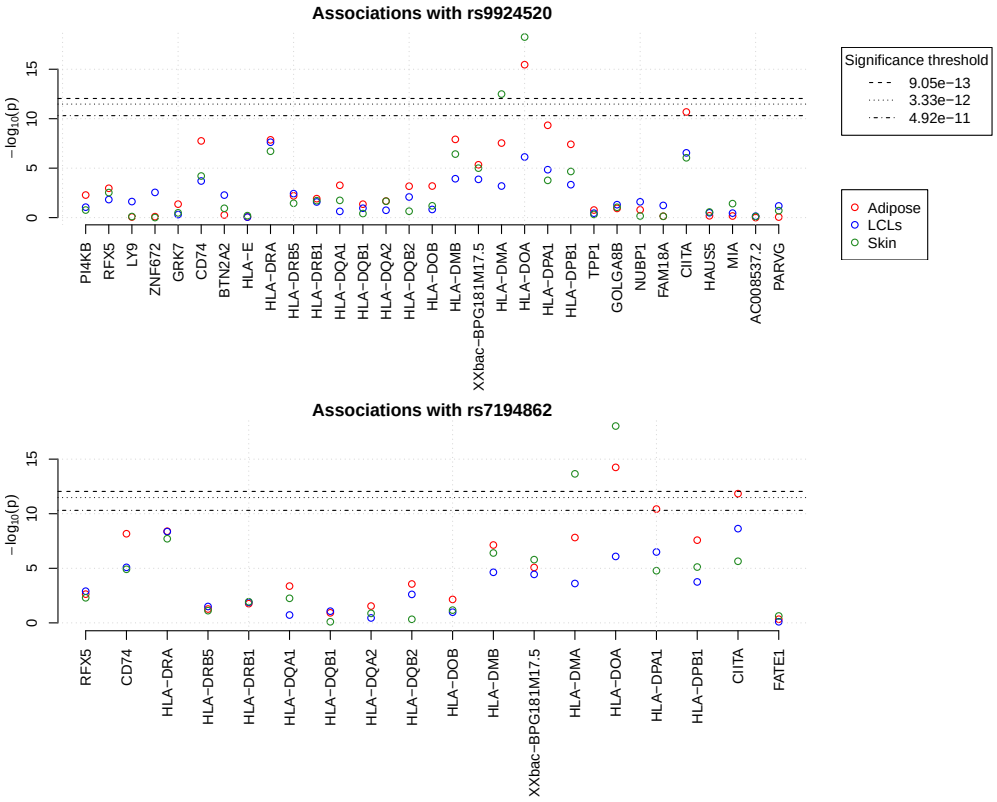


Figure 1: Marginal associations for MHC class II component.

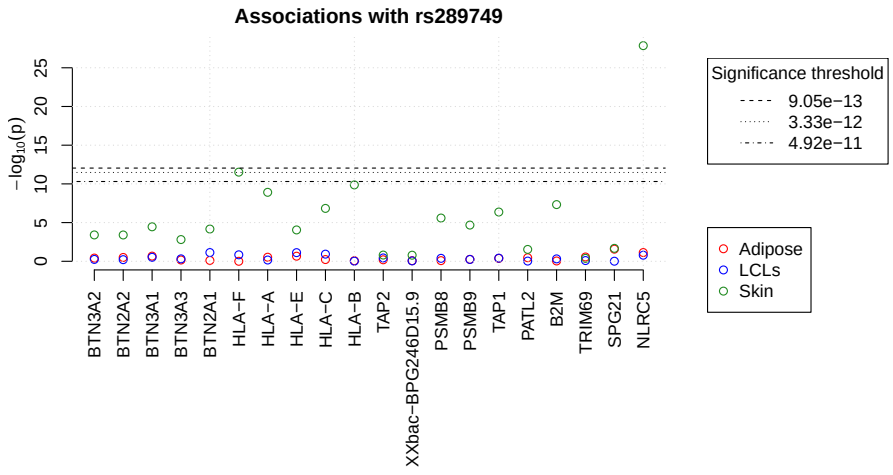


Figure 2: Marginal associations for MHC class I component.

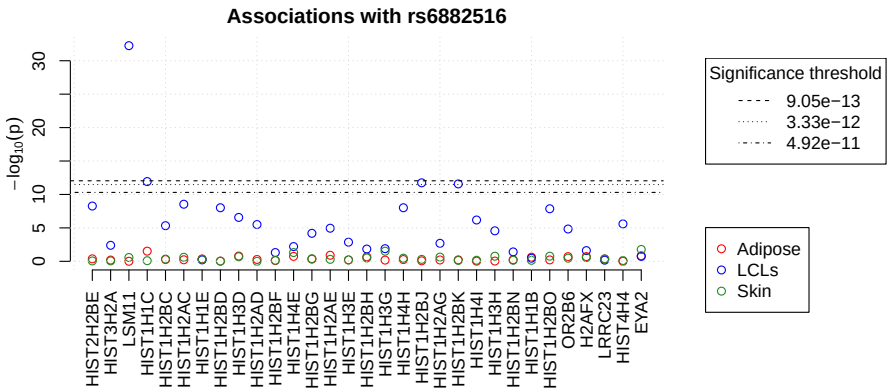


Figure 3: Marginal associations for RNA histone processing component.

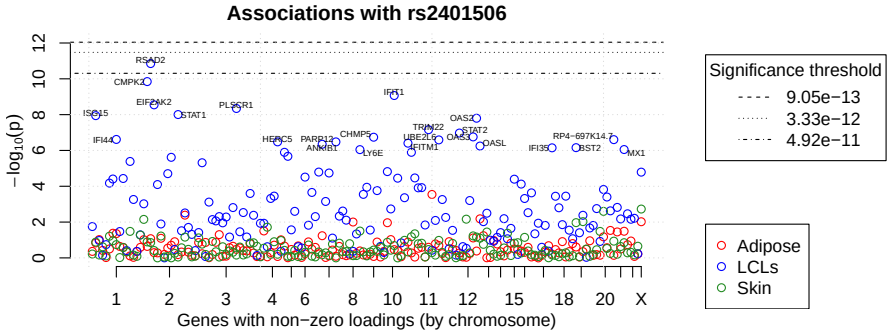


Figure 4: Marginal associations for Type I Inteferon component

Component	SNP (chr)	p-value for association between component's individual scores vector and SNP	Most associated gene in *trans* via marginal associations
MHC class II*	rs7194862 (16)	1.74e-14	9.30e-19
	rs9924520 (16)	1.33e-23	5.56e-19
MHC class I	rs289749 (16)	1.34e-11	3.02e-12
Histone RNA processing	rs6882516 (5)	2.39e-15	1.17e-12
Type I interferon	rs2401506 (22)	9.82e-16	1.41e-11
ZNF gene network**	rs12630796 (3)	5.10e-17	2.09e-15
	rs17611866 (16)	5.40e-21	3.38e-14

Table 1: Comparison of the p-values obtained using our approach and marginal associations. For each of the components identifying potential *trans* signals (column 1), the SNPs most associated with the component's individual scores are given in column 2. Column 3 contains the p-values for these associations. Marginal scans for *trans* associations between the SNP and genes in the component (PIP $>$ 0. 5) were performed and the smallest p-value across all three tissues given in column 4. The smallest p-value in each row is highlighted in red. *Results for two components describing a regulation pathway involving the MHC class II genes are given. **For the ZNF gene network component, results for most significant SNPs from clusters on chr 3 and chr 16 are presented.

of 9.05×10^{-13}. The SNPs rs17611866 and rs12630796 are uncorrelated (r^2=0.01). The model fit with both SNPs as predictors of the component scores is highly significant (p-value=1.8×10^{-34}).

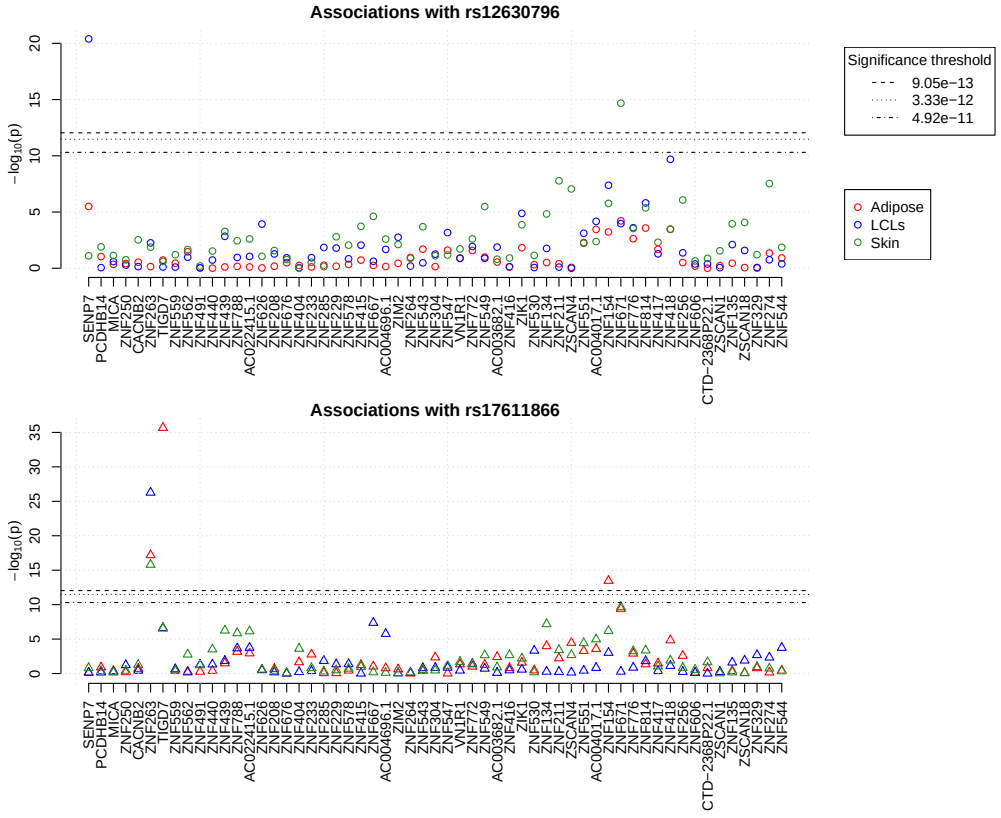


Figure 5: Marginal associations for Zinc finger component.

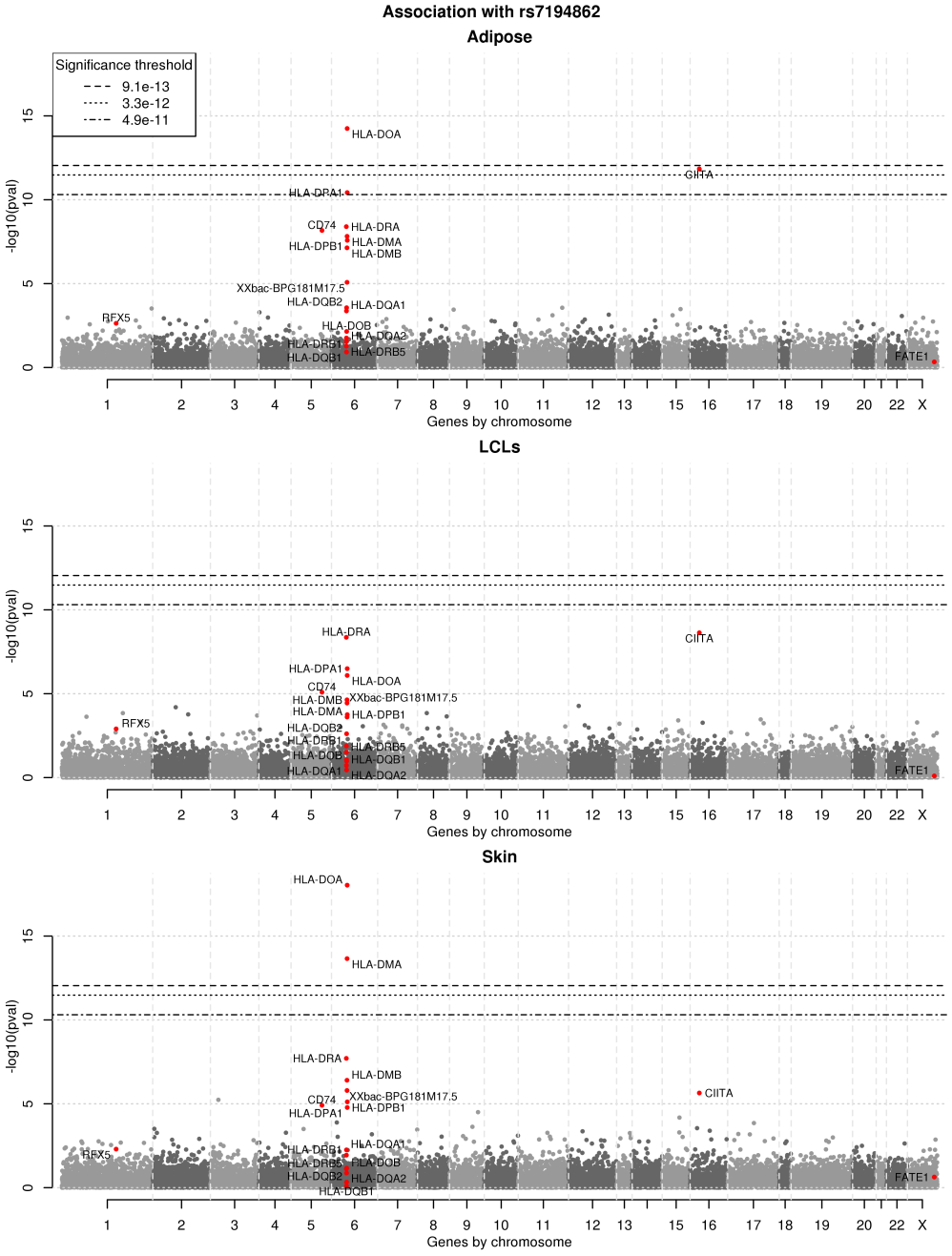


Figure 6: Marginal associations between lead SNP from MHC class II component active in LCLs, and all genes. Genes identified in the component are shown in red.

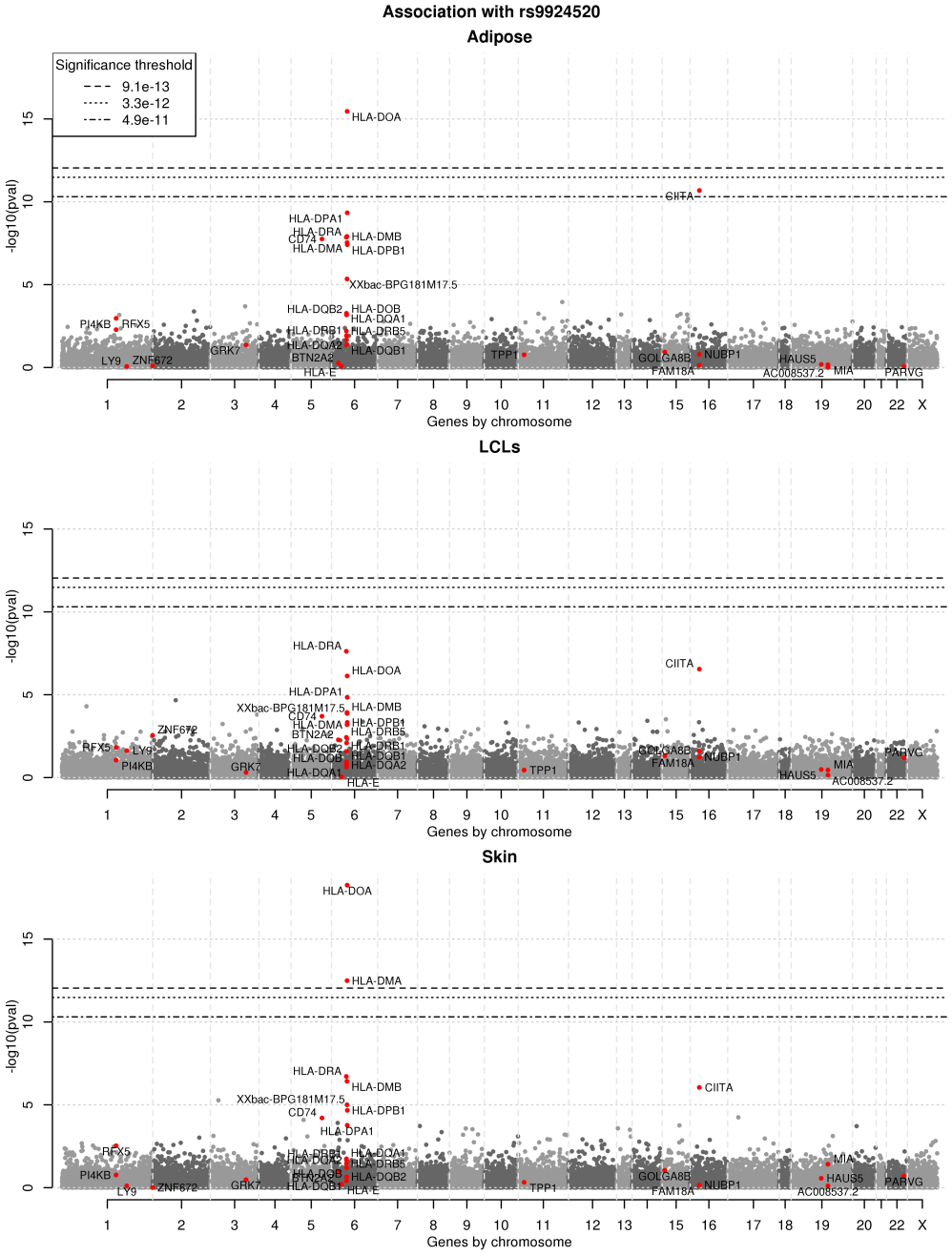


Figure 7: Marginal associations between lead SNP from MHC class II component active in adipose and skin, and all genes. Genes identified in the component are shown in red.

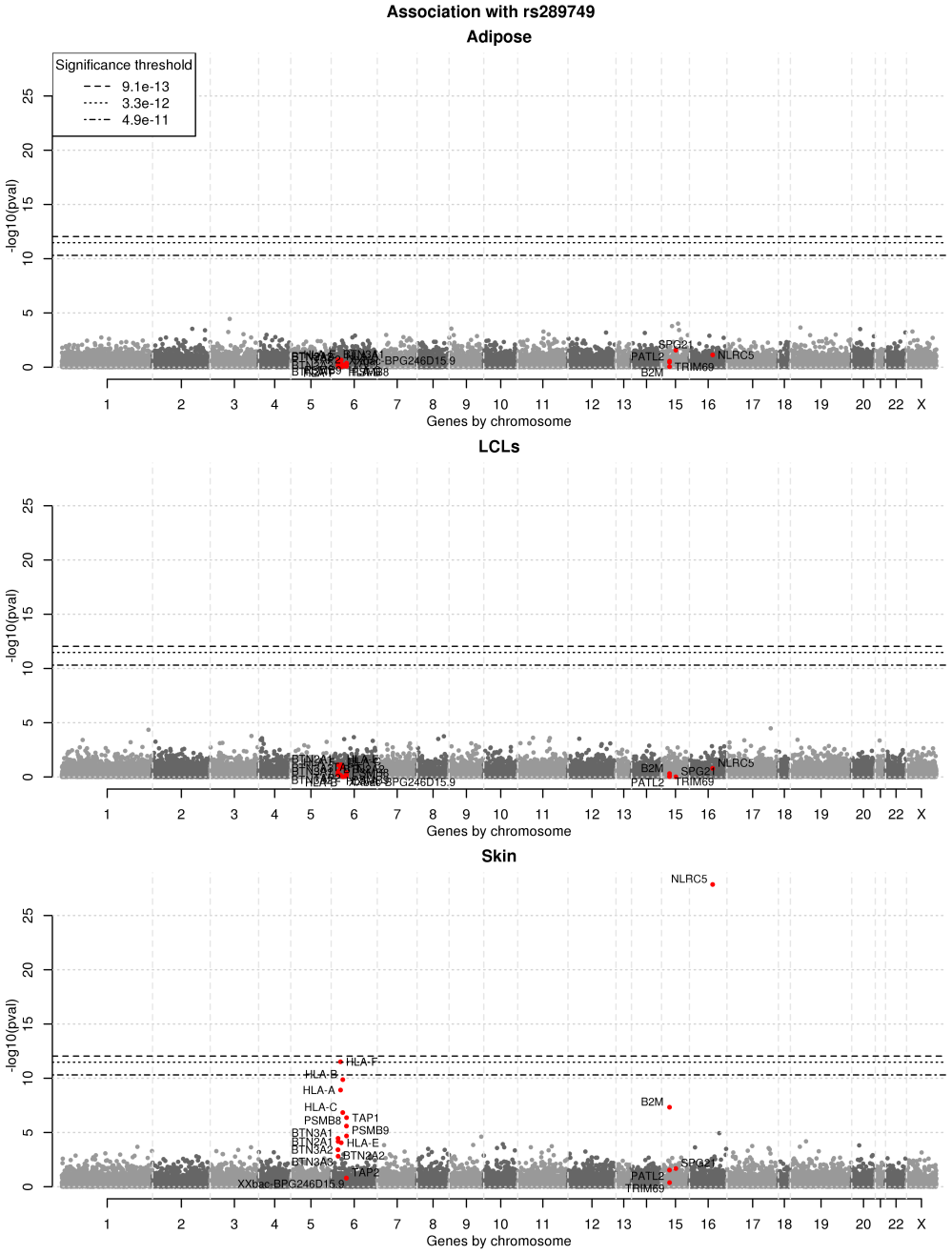


Figure 8: Marginal associations between lead SNP from MHC class I component and all genes. Genes identified in the component are shown in red.

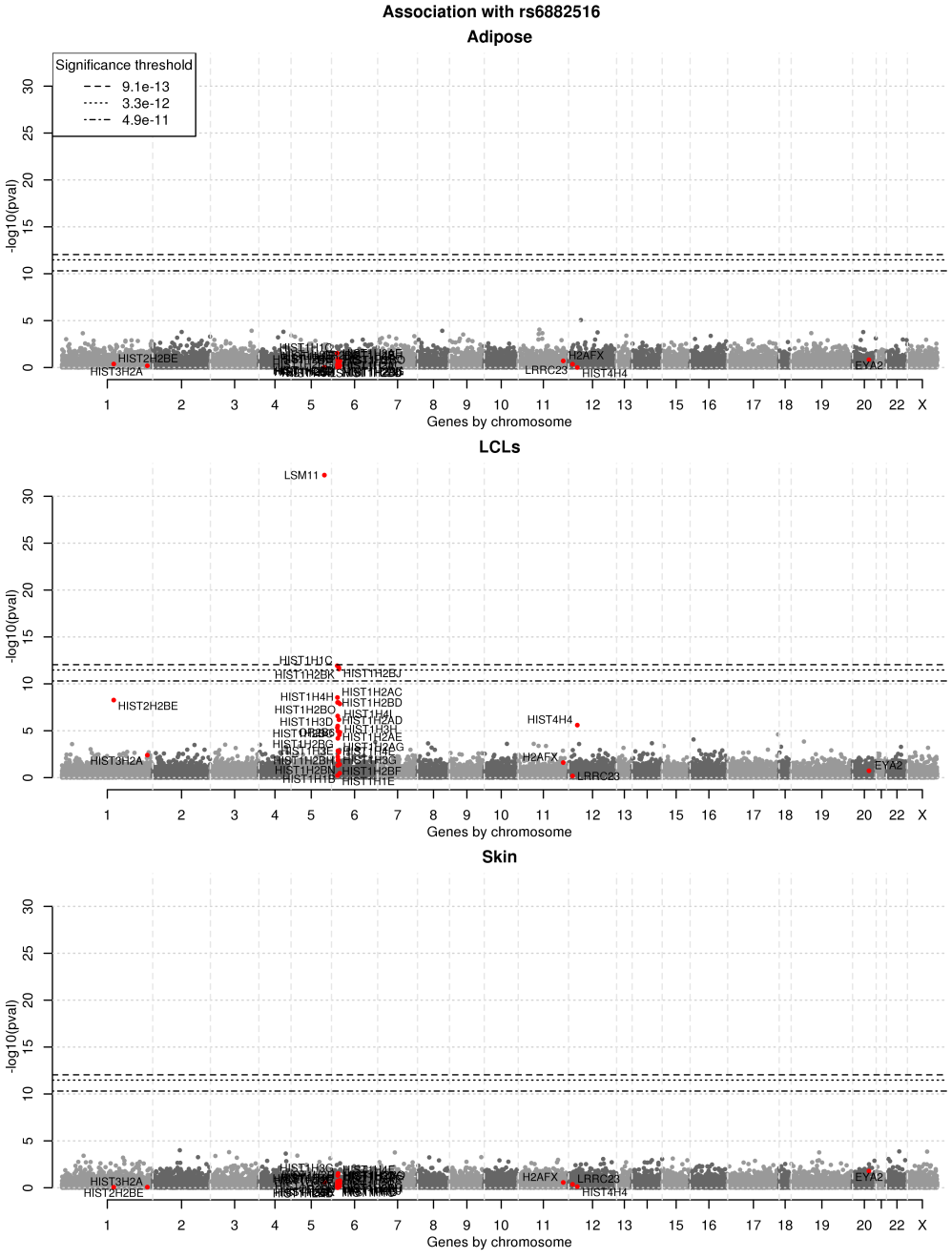


Figure 9: Marginal associations of histone RNA processing component lead SNP and all genes. Genes identified in the component are shown in red.

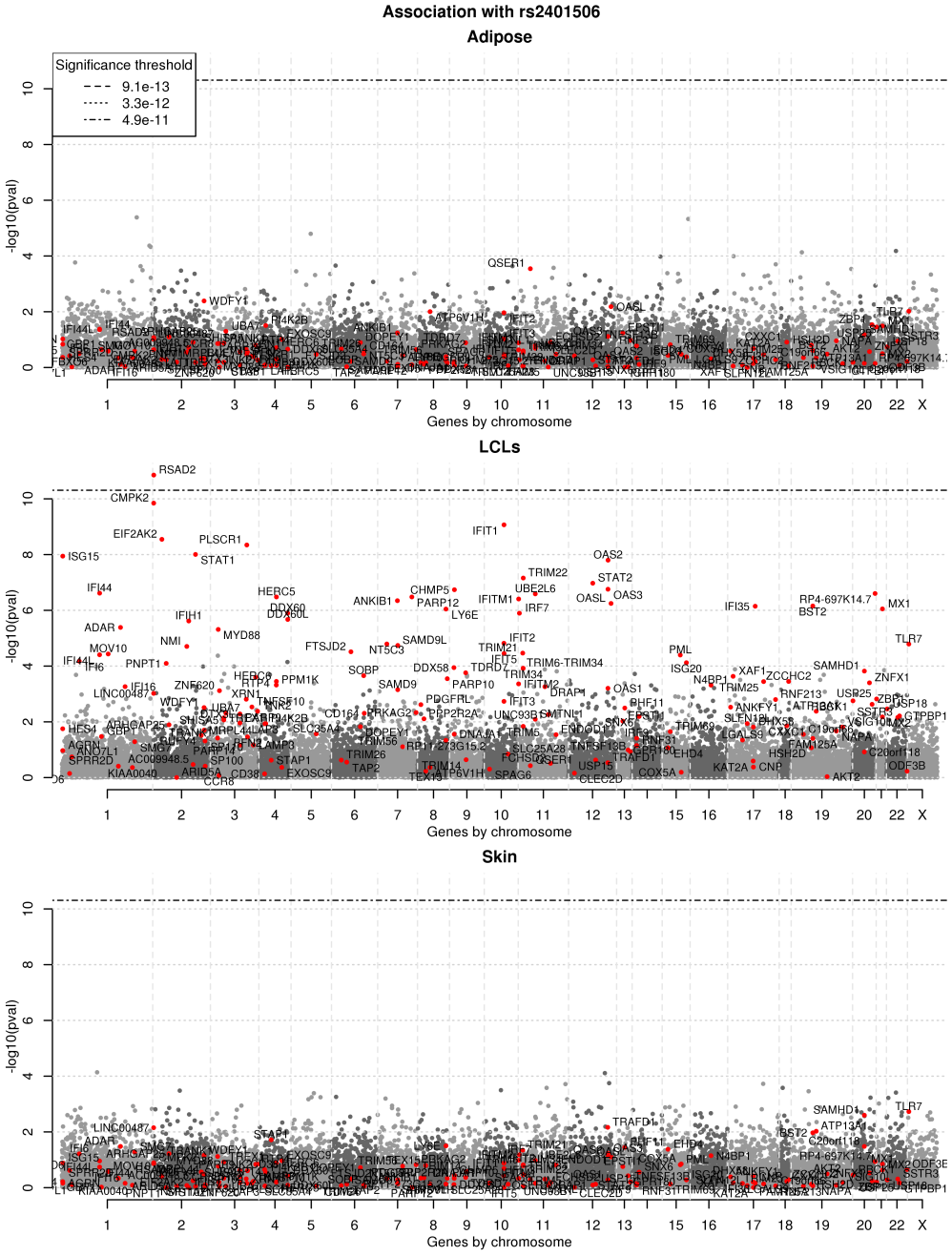


Figure 10: Marginal associations of Type I Inteferon component lead SNP with genes in that component. Genes identified in the component are shown in red.

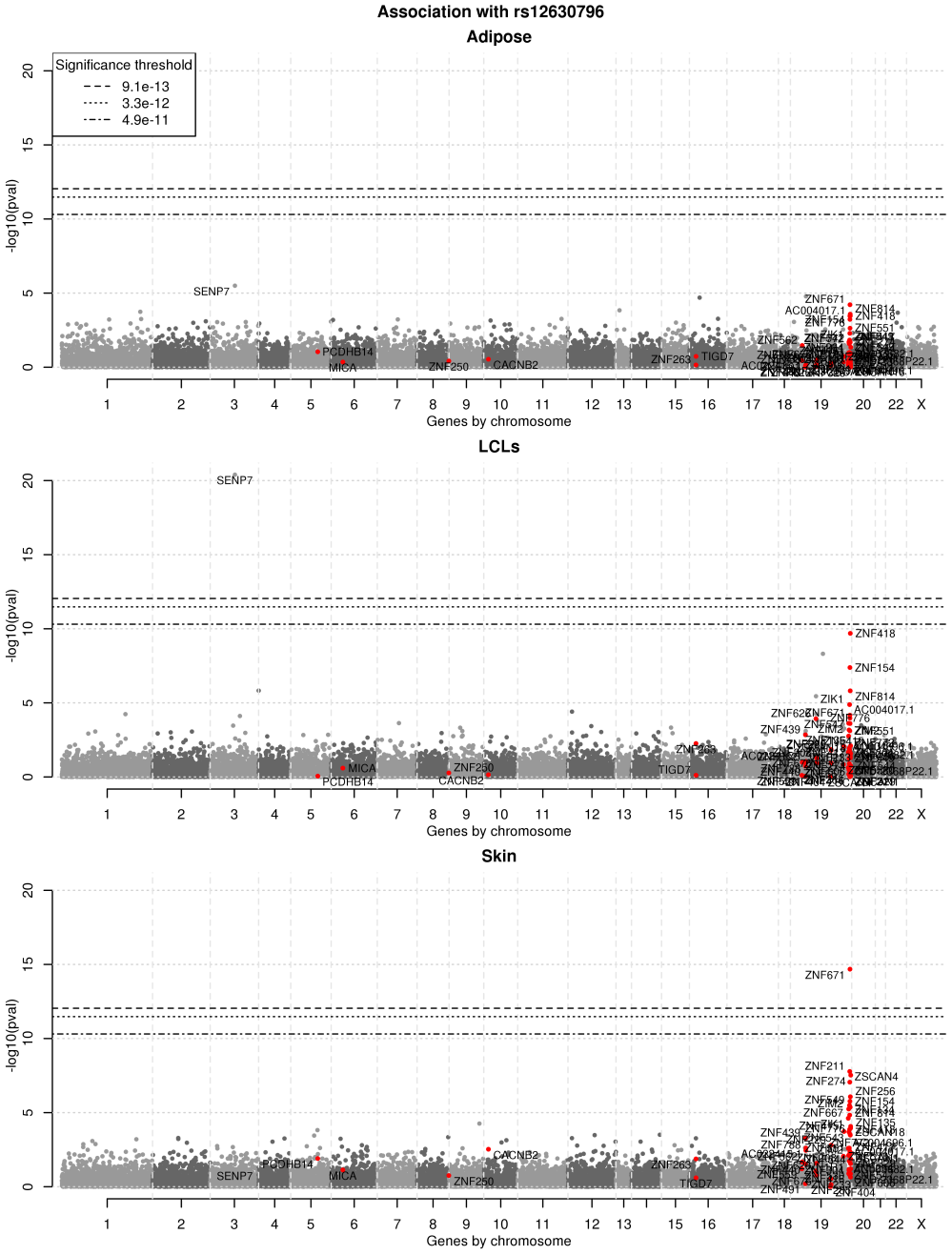


Figure 11: Marginal associations of Zinc finger component lead SNP on chromosome 3 with genes in that component. Genes identified in the component are shown in red.

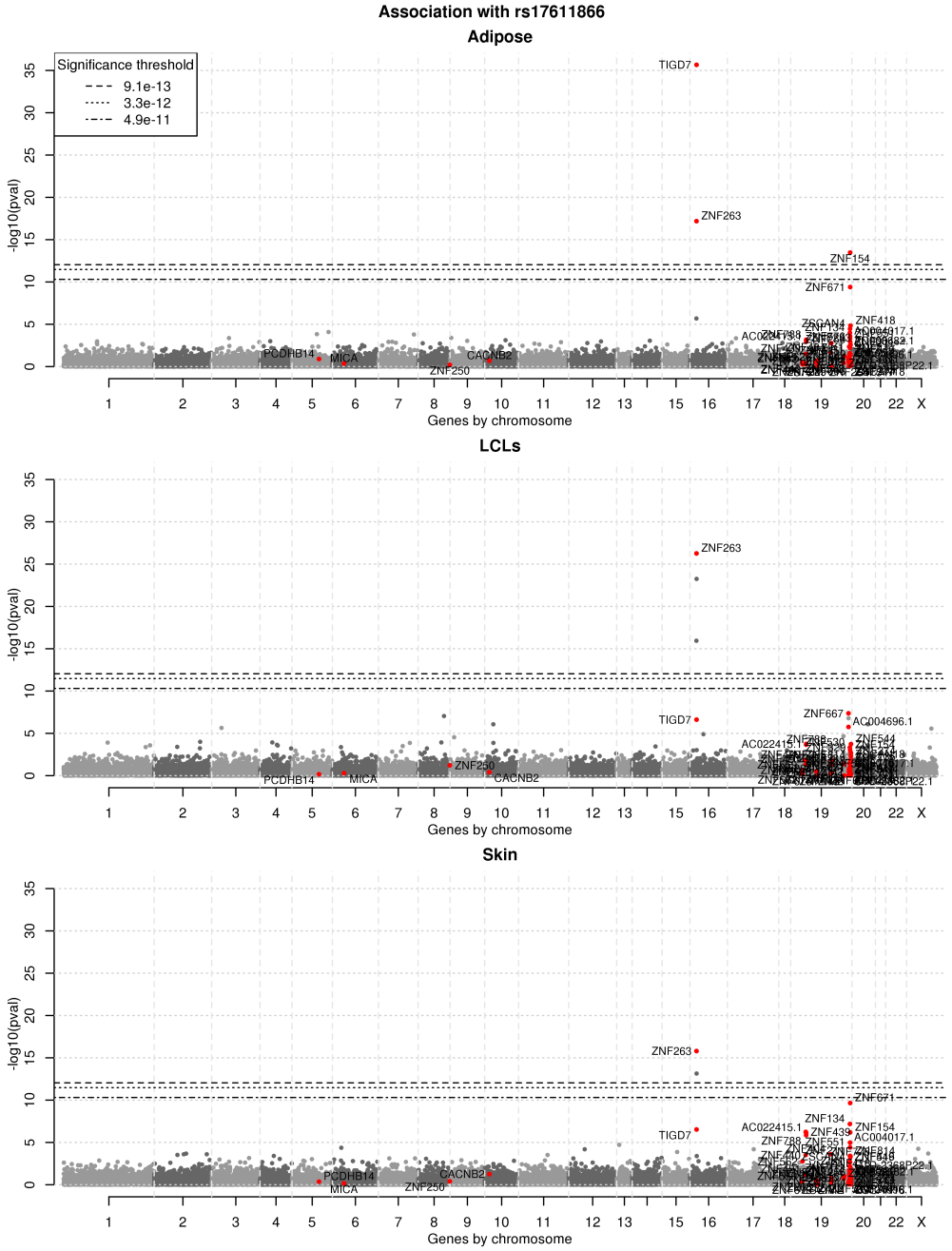


Figure 12: Marginal associations of Zinc finger component lead SNP on chromosome 16 with genes in that component. Genes identified in the component are shown in red.

2.3 Gene ontology analysis

We performed a gene ontology (GO) analysis for genes identified in the components (Figure 13). Many components contained genes that were enriched for GO terms, although we caution against over interpretation of these results. For example, some dense components may show evidence of enrichment for a GO term, but this does not mean that every gene in that component is involved in the same network or biological process. Figure 13 summarizes the number of components associated with batch variables, phenotypes and enriched for a GO term.

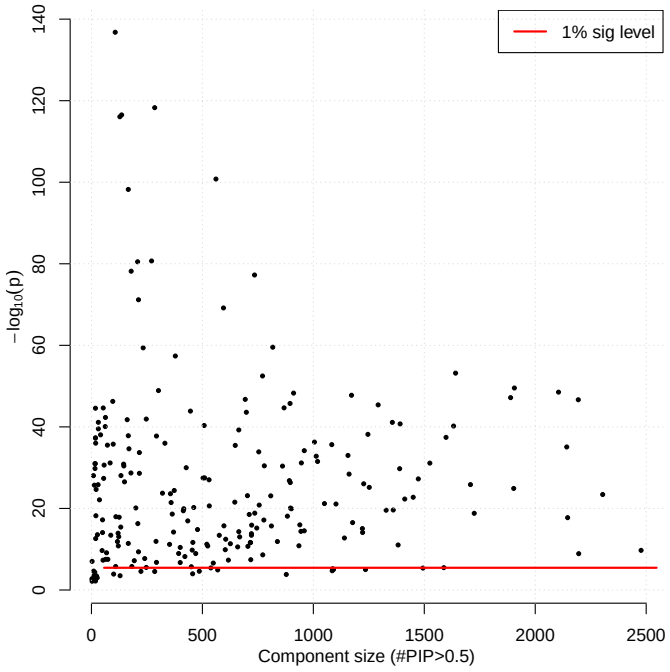


Figure 13: Gene ontology p-values for 236 robustly clustered components. x-axis shows the component size (number of PIPs>0.5) and the y-axis shows the $-\log_{10}$(p-value) for the most strongly associated GO term. The red line indicates a 1% significance threshold.

2.4 Investigation of PEER factors

When searching for *cis* eQTLs is has become common for researchers to use PEER Stegle et al. (2010) to estimate and remove confounding variables. SDA does not use PEER as a pre-processing step. However, we do use PEER when examining the marginal associations between the SNPs identified by SDA and the genes in the associated components. A reviewer raised a concern that using PEER may result in loss of power to detect *trans* eQTLs. In fact, we found that by using PEER we recovered more significant associations (data not shown) compared to an approach which did not use PEER.

We further examined the PEER factors for each tissue we estimated and found them all to be dense in terms of their gene weights (or loadings) (data not shown). (The *trans* eQTL networks that we report in this paper (see Figures 2-6 of the main paper) are sparse.) We also carried out GWAS for the individual score vectors for each PEER component (data not shown). Using a strict threshold of $5 \times 10^{-8}/15 = 9.33 \times 10^{-9}$, we only find one significant SNP. A peer factor (from a decomposition of data from Skin tissue) was associated with rs146412791 (chromosome 9, p-value $=9.43 \times 10^{-10}$, maf=0.017). rs146412791 lies within the *SLC35D2* gene and was the only SNP in the region that had a small p-value. Our conclusion is that the PEER components do not remove *trans* effects.

2.5 Application of ICA and PCA to the TwinsUK dataset

Based on a comment by a reviewer, we applied Independent Component Analysis (ICA) Hyvärinen, 1999 to the TwinsUK dataset in addition to SDA. This method decomposes a matrix of data into components such the gene loadings are uncorrelated and show departures from Gaussianity.

We used the R package fastICA to run ICA on a single matrix of data consisting of the concatenated normalized expression data from all 3 tissues. Only 618 out of 845 individuals had expression data on all 3 tissues, so this matrix had 618 rows and 3×18409 columns. We fit the maximum number of components possible (618). We selected the 200 components for which the gene loadings had a kurtosis of greater than 3.5. For each of these components we ran a GWAS against all SNPs using the component individual scores vectors as a phenotype.

We found 26 had a SNP with a significant association ($p < 1 \times 10^{-10}$). We applied the method suggested in Rotival et al. (2011) (that uses the R package fdrtool to identify which genes from the gene loadings are significant at an FDR of 1×10^{-3}. The majority of the components contained a significant peak of association at genes near the associated SNP (suggesting a *cis* eQTL) but the fdrtool method identified genes throughout the genome (in our opinion spuriously). To further examine this we produced plots of the marginal association of the lead SNP with all 18,409 genes for a subset of the components that looked most like plausible *trans* effects. In almost all cases there was no significant evidence that the genes identified were associated marginally with the SNP. The exception was a component which identified the SNP underlying the *KLF14 trans* eQTL network. None of the components identified using this ICA approach overlapped with the components we find and show in Figure 2-6 of the main paper.

We repeated the analysis with data in a single tissue (LCLs) with qualitatively similar results. Due to the considerable amount of material already reported in the Supplementary material we have not included these results.

In addition, we found several papers that advise against simple concatenation of datasets like this (Groves et al., 2011; Lian et al., 2015; Shen et al., 2013; Virtanen et al., 2012).

We also applied PCA to the concatenated data and the LCLs data only (also at the suggestion of a reviewer). The gene loadings of all the components we looked at were not sparse and none of the components were associated with any SNPs at $p < 1 \times 10^{-10}$ (data not shown).

2.6 Run with the highest negeative free energy run

In addition to clustering results of SDA from across 10 runs, we also investigated the run that produced the highest value for the negative free energy. This run produced 944 components (after removing components that shrink to zero) (Supplementary Data Set), of which 51 showed significant genetic associations (p-value$< 1 \times 10^{-10}$); 39 of these showed clear *cis* effects (see Table 2 for a summary of these components). Although we identify components in many of the 10 runs associated with a SNP in the *KLF14* gene, these components do not cluster well and this signal is not represented in our set of 236 robust components. Dense factors such as these can suffer from non-identifiability due to rotational invariance (Fokoue (2004)) this may also explain why the signal is split across several components within one run; however this does not matter if our aim is to identify possible *trans* SNPs for further investigation.

	Tissue activation pattern							Row totals
	A	L	S	AL	AS	LS	ALS	
# Components	188	273	203	24	140	15	101	944
SNP (1×10^{-10}) *cis*	0	1	0	1	1	0	36	39
SNP (1×10^{-10}) *trans*	3	2	0	0	0	1	6	12
Phenotype (1×10^{-6})	52	0	21	0	17	0	1	91
Sequencing (1×10^{-6})	88	121	103	6	45	5	16	384
GO term (1×10^{-6})	145	219	144	11	91	9	36	655

Table 2: Summary of 944 components from the model run with the largest negative free energy. Components are categorized according to which set of tissues they are active in (A : Adipose L : LCLs, S : Skin) using a threshold of 0.02 on the tissue scores matrix. The first row of data gives the number of components with each activation pattern; subsequent rows summarize the number of component associated with SNPs, phenotypes, batch variables and enriched for GO terms (with significance levels given in brackets).

3 Trans eQTL simulations

This section describes details of a simulation study that evaluates whether our method has the power to find *trans* effects in gene expression data. The simulated data consisted of genotypes and gene expression data in several tissues, containing a variety of signals including *trans* effects, *cis* effects and confounding factors. The tensor decomposition version of SDA was compared to individual matrix decompositions which analysed data for each tissue independently. In addition, to test robustness to missing data, the tensor decomposition was run with a subset of the samples hidden.

3.1 Data simulation

Simulated data consisted of genotype and gene expression data for $N = 700$ related individuals (150 monozygotic twin pairs, 150 dizygotic twin pairs and 100 singletons). Gene expression data was simulated at $L = 2,500$ genes in $T = 3$ tissues and contained both non-genetic signals (noise and confounding factors) and genetic signals (*cis* and *trans* effects). It was assumed that each gene contained only one SNP; this simplified case is equivalent to assuming that there is at most one *cis* eQTL for each gene.

3.1.1 Genotypes

Of the $L = 2,500$ SNPs simulated, $C_{\text{cis}} = 500$ (20%) were randomly selected to be *cis* eQTLs. A *cis* eQTL partially determined the expression of its nearby gene. A subset of the *cis* eQTLs were additionally assumed to be *trans* eQTLs. *Trans* eQTLs were not only associated with a nearby gene (via a *cis* effect) but also multiple other genes, creating a *trans* network.

Let $G \in \mathbb{R}^{N \times L}$ be the matrix of simulated genotype data. Genotypes were simulated under the Hardy-Weinberg equilibrium with a minor allele frequency (MAF) drawn uniformly from $[0.05, 0.5]$ (unless the SNP was also a *trans* eQTL in which case MAF $= 0.3$). Monozygotic twins shared all their genetic material and dizygotic twins shared half of their genetic material. All SNPs were sampled independently.

3.1.2 Gene expression

The simulated gene expression data ($\mathcal{Y}^{N \times L \times T}$) consisted of noise, confounding factors, *cis* and *trans* effects. The vector of expression levels for gene l in tissue t is denoted $\mathbf{y}_{lt}$. The data was simulated in stages, noise, confounding factors and *cis* effects were generated initially. These three signals were then combined, and *trans* effects incorporated.

Noise: A heteroscedastic model was used to simulate noise,

$$\mathbf{y}_{lt}^{\text{noise}} \sim \mathcal{N}(0, \lambda_{lt}^{-1}), \ \sqrt{\lambda_{lt}^{-1}} \sim \mathcal{G}(100, 0.01). \tag{38}$$

The standard deviation of the noise $\sqrt{\lambda_{lt}^{-1}}$ was drawn from a Gamma distribution with mean 1 and low variance (note that λ_{lt} is the noise precision for gene l in tissue t).

Confounding factors: $C_{\text{cf}} = 10$ independent confounding factors were simulated as follows,

$$\mathbf{y}_{lt}^{\text{cf}} = \sum_{c=1}^{C_{\text{cf}}=10} \mathbf{a}_c b_{tc} x_{cl} \quad \text{for} \quad l \in \{1, \ldots, L\},\ t \in \{1, 2, 3\}$$
$$\mathbf{a}_c \sim \mathcal{N}_N(\mathbf{a}_c|0, I_N),\ |b_{tc}| \sim \mathcal{G}(100, 0.01),\ x_{cl} \sim 0.5\mathcal{N}(x_{cl}|0, 0.1) + 0.5\delta_0(x_{cl}) \tag{39}$$

where the sign of b_{tc} was randomly selected. Note that the confounding factors were simulated under the PARAFAC model with a sparsity level of 50%.

***Cis* effects:** Let $\mathbf{g}_l \in \mathbb{R}^N$ be the vector of simulated genotypes for SNP l. Supposing that SNP l was a *cis* eQTL, then its contribution to the expression of gene l was given by

$$\mathbf{y}_{lt}^{\text{cis}} = \hat{\alpha}_l \tilde{\alpha}_t \mathbf{g}_l, \tag{40}$$

where

$$\begin{cases} |\hat{\alpha}_l| = \phi_l^{cis} \\ \tilde{\alpha}_t = 1 \end{cases} \quad \text{if SNP } l \text{ also acted as a } \textit{trans} \text{ eQTL}$$
$$\begin{cases} |\hat{\alpha}_l| \sim \mathcal{G}(4, 0.1) \\ \tilde{\alpha}_t \sim \mathcal{B}\text{ernoulli}(0.5) \end{cases} \quad \text{otherwise.} \tag{41}$$

$\hat{\alpha}_l$ can be thought of as an effect size (the effect direction is random) and $\tilde{\alpha}_t$ as a binary value indicating whether the *cis* effect was active in tissue t. A different effect size was used depending on whether the eQTL was also a *trans* eQTL or not; this is discussed more later. A *cis* effect was active in a particular tissue with probability 0.5, (unless the eQTL was also a *trans* eQTL, in which case it was active in every tissue, although it did not necessarily target downstream genes in every tissue). $\mathbf{y}_{lt}^{\text{cis}}$ was set to zero if SNP l was not a *cis* eQTL.

Combining noise, confounding factors and *c*is effects: Simulated noise, confounding factors and *cis* effects were combined additively to get a temporary set of expression levels for a gene l in tissue t,

$$\mathbf{y}_{lt}^{\text{tmp}} = \mathbf{y}_{lt}^{\text{noise}} + \mathbf{y}_{lt}^{\text{cf}} + \mathbf{y}_{lt}^{\text{cis}}. \tag{42}$$

***T*rans effects:** Finally, *trans* effects were simulated. *Trans* associations between SNPs and distant genes were created as follows; the *trans* SNP regulated a nearby gene (via a *cis* effect). It was further assumed that this gene was a transcription factor (abbreviated as TF) and regulated multiple downstream genes (called target genes). Importantly, the *trans* eQTL was only indirectly associated with the target genes.

Data sets were simulated to contain 20 *trans* effects. The number of target genes in the *trans* networks (M_{trans}) varied, and target genes were selected at random. (For simplicity, a gene acting as a TF in a *trans* effect could not additionally be involved in another *trans* effect, however any of the other genes – including those regulated by a regular *cis* eQTL – could be regulated by any number of TFs.) *Trans* effects were active in one, two or three tissues. If the *trans* effect was active in several tissues, the same network of genes was regulated in each tissue.

Let l be a gene, and S_l be the set of TFs that regulate it. *Trans* effects were simulated as follows,

$$\mathbf{y}_{lt}^{trans} = \sum_{j \in S_l} \hat{\beta}_{lj} \tilde{\beta}_{tj} \mathbf{y}_{jt}^{\text{tmp}}$$
$$|\hat{\beta}_{lj}| \sim \mathcal{G}(\psi^{trans}, 0.02) \tag{43}$$

where $\hat{\beta}_l$ was the relative effect of TF j on gene l (with a random effect direction). $\hat{\beta}_{tj}$ was equal to 1 if the TF j was active in tissue t and 0 otherwise. In order to investigate a variety of scenarios, of the 20 *trans* effects simulated, 12 were active in just one tissue, 4 were active in 2 tissues and the remaining 4 were active in all tissues. If l was not a target gene for any of the 20 TFs (i.e. S_l was the empty set) then $\mathbf{y}_{lt}^{\text{trans}} = 0$.

Selecting parameters for the *trans* effects: Three parameters determine the strength of a simulated *trans* effect: the effect of the *cis* eQTL on the TF (ϕ^{cis}); the effect of the TF on the target genes (determined by ψ^{trans}) and the number of target genes (M_{trans}). In order to make the data as realistic as possible, these parameters were selected so that signal strengths were similar to those seen in real data sets. The real *trans* signal used as a reference was the *KLF14 trans* signal (Small et al., 2011).

KLF14 is a gene on chromosome 7 that encodes for a transcription factor. There is a group of highly correlated SNPs just upstream of the *KLF14* gene that are associated with its expression levels. Small et al., 2011 regressed gene expression levels across the whole genome against one of these SNPs (rs4737102) and found an enrichment of low p-values suggesting that KLF14 regulates multiple genes across the genome.

For half of the *trans* effects, ϕ^{trans} and ψ^{trans} were selected to match the effect sizes seen in the *KLF14 trans* signal. The remaining 10 *trans* effects were weaker, for these signals, values of the parameters were halved. *Trans* effects had either 150 or 75 target genes (6% or 3% of all genes). Supplementary Table **??** gives a summary of the *trans* effects simulated.

Index	ϕ^{cis}	ϕ^{trans}	M_{trans}
1-5*	0.6	20	150
6-10**	0.3	10	150
11-15*	0.6	20	75
16-20**	0.3	10	75

Table 3: Summary of the parameters used to simulate *trans* effects. *Trans* effects were grouped into sets of 5 according to their signal strength. $*$ indicates signal strength match the *KLF14 trans* signal and $**$ indicate weaker signals. Within the groups they were further split according to their activity in different tissues; 3 were active in just one tissue, 1 was active in 2 tissues and 1 was active in all three tissues.

Combining all of the data: Finally, the contribution from the *trans* effects was incorporated to create a final set of simulated expression levels,

$$\mathbf{y}_{nl}^{\text{final}} = \mathbf{y}_{nl}^{\text{tmp}} + \mathbf{y}_{nl}^{\text{trans}}. \tag{44}$$

A summary of the simulation parameters are given in Supplementary Table 4. A total of 50 data sets were simulated.

Parameter	Value	Description
T	3	Number of tissues
N	700	Number of individuals
L	2,500	Number of genes
C_{cf}	10	Number of confounding factors
C_{cis}	500	Number of *cis* effects
C_{trans}	20	Number of *trans* effects

Table 4: Simulation parameters.

3.2 Details of methods compared

Several different versions of SDA were run, see summary in Supplementary Table 5. The tensor decomposition was run with a Gaussian prior on the individual scores matrix (denoted T_G) and also kinship-informed prior on the individual scores matrix (T_K) (see details in section 1.11). In addition, individual matrix decompositions of each tissues were performed, using SDA with $T = 1$.

Performance in the presence of missing samples was also tested. Up to 2 samples were removed for each individual at random such that only 75% of the data remained. Two versions of SDA were run in this scenario (i) the extension to deal with missing samples given in section 1.10, which essentially ignores the missing data, and (ii) a naive approach that removes any individual with missing samples, then runs a tensor decomposition on the remaining data. On average, only 350 individuals had complete data. For both of these methods, a Gaussian prior was used for the individual scores matrix. These two approaches, (i) and (ii), are denoted T_G^i and T_G^r respectively.

All methods were run with the initial number of components set to 100; this was a sufficient number for all the models to recover the *trans* effects and confounding factors. It was not expected that the methods pick up all the *cis* eQTLs. Hyperparameters (see section 1.5) were chosen to be uninformative. All methods were run 10 times with different initialisations and for 1,000 iterations.

3.3 Post-processing and metrics

In these simulations, the aim was to investigate recovery of the underlying signals in the data. Several metrics were utilised to evaluate recovery of the confounding factors and *trans* effects. For the *trans* effects, both the recovery of the causal SNP and also the set of target genes were evaluated.

Although variational Bayes is a deterministic algorithm, there is no guarantee that different initialisations will result in the same set of component estimates. The negative free energy can be used to select the 'best' run. An alternative approach detailed in the online methods combines component estimates from across multiple runs. The idea is to average similar components from across multiple runs of the method to get a set of 'robust' components. For these simulations, a correlation threshold of 0.5 was used to terminate the clustering algorithm. Only clusters containing 5 or more components were used in further analysis.

The following performance metrics were applied to the component estimates from the

	Method	Description
Complete data	T_G	Tensor decomposition with a Gaussian prior on the individual scores matrix.
	T_K	Tensor decomposition with a mixture of the Kinship matrix and identity as the prior on the individual scores matrix (see section 1.11).
	M_G	Matrix decompositions on data for each tissue separately. Gaussian prior on the individual scores matrices.
Missing data	T_G^i	Missing samples are ignored in the model likelihood (see section 1.10.1).
	T_G^r	Individuals with any missing data were removed to get a set of individuals with complete data.

Table 5: Different versions of SDA run on simulated data.

'best' run based on negative free energy values and the averaged component estimates from the clustering approach.

3.3.1 Confounding factors

To assess whether the models recovered the confounding factors, a search was performed to find the set of estimated components that best explain the true confounding. This was performed by maximising the absolute correlation between the truth and estimated individual component scores (via a greedy algorithm), resulting in a set of 10 estimated components that looked most like the true confounding factors. Recovery was then evaluated by calculating the average absolute correlation between estimated individual scores vectors and the truth.

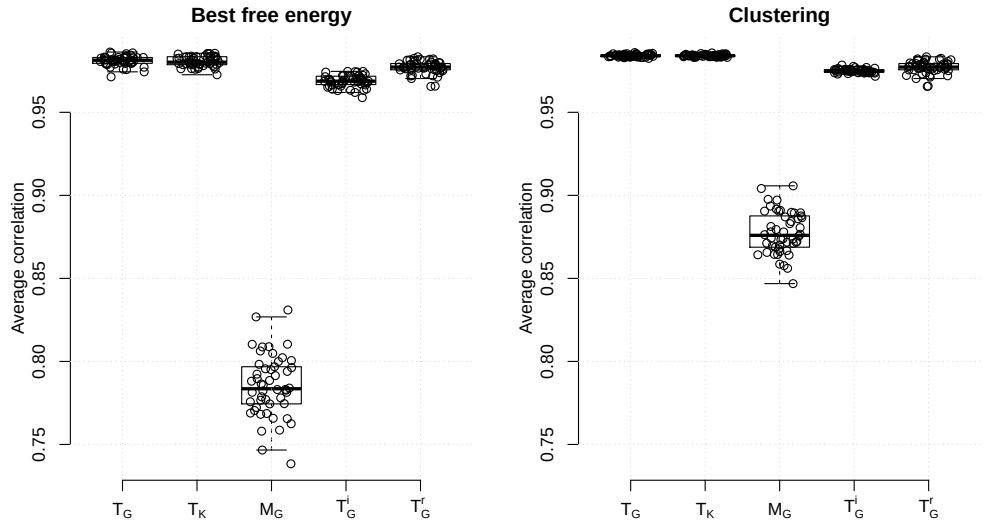
3.3.2 GWAS

As with the real data, each individual scores vector was treated as a phenotype and a genome-wide scan for association was performed. If a SNP simulated was significant;y associated with a component *trans* (using a p-value threshold of 2×10^{-7}), then the *trans* signal was said to be 'recovered'.

The fraction of each type of *trans* effect recovered for each type (across 50 simulated data sets). Results for the matrix model are averaged across the three tissues, taking into account the fact that the signal may not exist in all three data types.

3.3.3 Power to detect regulated genes

Once a component has been identified as describing a *trans* effect (based on a GWAS signal at a *trans* eQTL), the component was further investigated to evaluate whether the correct set of target genes had been recovered. Power and false positive rates (FPR) were used to compare the genes in the component, those with PIPs>0.5, with the true set of target genes.



Supplementary Figure 40: Average correlation between components describing confounding factors and the truth. Boxplots summarise results across 50 simulated data sets for each method.

For a *trans* effect with vector of PIPs given by $\hat{\mathbf{s}}$, power and FPR are defined as

$$\text{Power} = \frac{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 1, \hat{\mathbf{s}}_l > 0.5)}{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 1)} \tag{45}$$

$$\text{FPR} = \frac{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 0, \hat{\mathbf{s}}_l > 0.5)}{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 0)} \tag{46}$$

where $\tilde{\mathbf{s}}$ is a binary vector of length L such that $\tilde{\mathbf{s}}_l = 1$ if gene l is a target gene and 0 otherwise. $\mathbb{I}$ is an indicator function. Results for the matrix decompositions were averaged, taking into account the number of tissues each *trans* effect was active in.

3.3.4 Combining factors

In some situations, a single *trans* effect, active in multiple tissues, was modelled by several components. This occurs because – although the simulated *trans* effects act on the same set of target genes in each tissue – the contribution to the expression of the target genes varies. It is easy to see why this happens by considering the expression of the TF in each tissue. Expression of the TF depends on the genotype (an effect which is shared across tissues), and two tissue independent effects; confounding factors and noise. When these latter two effects are large, the expression of the TF is largely uncorrelated across tissues and its effect on the target genes across tissues will differ. If this is the case, the model treats *trans* effect as a different signal in each tissue resulting in them being picked up by several components. When this happens, these components were combined by averaging the scores and loadings.

3.3.5 Results

Supplementary Figure 40 shows the average correlation between the true and estimates component scores recovering confounding factors. The boxplots summarise results from

M_{trans}	#tiss	Best free energy			Clustering		
		T_G	T_K	M_G	T_G	T_K	M_G
150	1	0.40	0.39	0.37	0.39	0.39	0.37
	2	0.82	0.82	0.36	0.82	0.84	0.39
	3	0.92	0.94	0.35	0.96	0.96	0.34
75	1	0.34	0.31	0.41	0.43	0.44	0.42
	2	0.70	0.78	0.39	0.80	0.76	0.39
	3	0.90	0.94	0.41	0.98	0.94	0.43

Table 6: Fraction of *trans* effects recovered. Only results for *trans* signals of strength half that of the *KLF14* *trans* signal shown. T_G performs a tensor decomposition with a Gaussian prior on the scores matrix, T_K performs a tensor decomposition with kinship-informed prior on the scores matrix and M_G performs matrix decompositions on data from each tissue with Gaussian priors on the scores matrices. The best result in each row is highlighted in red. Results averaged across 50 data sets with no missing samples.

across 50 data sets. As expected, the joint analysis via a tensor decomposition (T_G and T_K) outperforms an analysis of each tissue separately (M_G). Even when data is missing (T_G^i and T_G^r), confounding factor recovery is good.

Supplementary Table 6 summarises results of *trans* effect recovery for *trans* signals with signal strength half that of the *KLF14* signal. The recovery of *trans* effects with signal strengths similar to the *KLF14* signal is almost perfect (data not shown). Supplementary Table 6 gives the fraction of each type of *trans* effect recovered (over 50 data sets). *Trans* effects in single tissues were harder to recover than *trans* effects in multiple tissues, and for these signals, performance of the tensor and matrix approaches were comparable. For the *trans* effects active in two or three tissues, the tensor approaches performed considerably better than the matrix decomposition. The is likely a result of the tensor decomposition pooling information from across multiple tissues, and also better explaining confounding.

A comparison of the tensor decomposition with different priors on the individual scores matrix (T_G and T_K) show no obvious difference. With high levels of noise and confounding in the data, in is unclear how heritable the *trans* signal actually are. In these simulations, the mixture prior recovers the underlying signals in the data only slightly better than the Gaussian prior, showing the Gaussian prior is surprisingly flexible.

The clustering approach to combine results across multiple runs slightly outperformed the results for the highest free energy run.

Conditional on *trans* effects being recovered by the models, the power to find the target genes involved is given in Supplementary Table 7. Again, only results for signals with half the signal strength of the *KLF14* *trans* signal are presented. Power to recover target genes is consistently high, and appears fairly independent of the number of target genes (M_{trans}) and activity of the *trans* effect across tissues. The tensor decomposition results in a higher power to find *trans* effects compared to the matrix decompositions. Clustering does not appear to have any benefits over the highest free energy run in terms of power. False positive rates were consistently below 0.5% for all methods (data not shown).

The performance of SDA when the data contained missing samples was also investigated. T_G^i incorporates information from individuals with incomplete data whereas T_G^r removes individuals with any missing data before running a tensor decomposition. On average, after

M_{trans}	#tiss	Best free energy			Clustering		
		T_G	T_K	M_G	T_G	T_K	M_G
150	1	0.84(0.04)	0.85(0.04)	0.68(0.18)	0.85(0.03)	0.85(0.03)	0.66(0.15)
	2	0.84(0.03)	0.83(0.07)	0.71(0.15)	0.84(0.04)	0.84(0.04)	0.59(0.22)
	3	0.83(0.06)	0.85(0.04)	0.64(0.19)	0.85(0.04)	0.85(0.04)	0.62(0.20)
75	1	0.78(0.10)	0.77(0.11)	0.71(0.13)	0.77(0.06)	0.77(0.10)	0.70(0.12)
	2	0.80(0.10)	0.80(0.09)	0.67(0.16)	0.80(0.06)	0.80(0.07)	0.68(0.17)
	3	0.82(0.05)	0.82(0.06)	0.71(0.13)	0.83(0.05)	0.83(0.05)	0.69(0.13)

Table 7: Power to find target genes in *trans* effects, conditional on the *trans* eQTL being recovered. Only results for *trans* signals of strength half that of the *KLF14 trans* signal shown. T_G performs a tensor decomposition with a Gaussian prior on the scores matrix, T_K performs a tensor decomposition with kinship-informed prior on the scores matrix and M_G performs matrix decompositions on data from each tissue with Gaussian priors on the scores matrices. Results averaged across 50 data sets with no missing samples. The best result in each row is highlighted in red.

M_{trans}	#tiss	Clustering	
		T_G^i	T_G^r
150	1	0.21	0.05
	2	0.58	0.28
	3	0.78	0.54
75	1	0.20	0.01
	2	0.56	0.14
	3	0.80	0.38

Table 8: Fraction of *trans* effects recovered. Results averaged across 50 data sets in which 25% of samples were missing. Two approaches are compared in this table, T_G^i ignores the missing samples using the approach from section 1.10.1 and T_G^r only uses data for individuals with no missing samples. Only results for *trans* signals of strength half that of the *KLF14 trans* signal shown.

removing individuals with any missingness, only 350 individuals remained. Not surprisingly, T_G^i consistently outperformed T_G^r, recovering more *trans* effects, with a higher power to find the target genes (see Supplementary Tables 8 and 9). In fact, the power to find target genes using T_G^i is comparable to the matrix decomposition approach M_G, on the complete data set. These results show the benefits of a method that can deal with missing samples. Again, false positive rates for these methods were very low ($< 0.5\%$).

M_{trans}	#tiss	Clustering	
		T_G^i	T_G^r
150	1	0.78(0.04)	0.56(0.05)
	2	0.69(0.13)	0.60(0.07)
	3	0.71(0.07)	0.61(0.05)
75	1	0.68(0.09)	0.35(0.05)
	2	0.67(0.11)	0.38(0.10)
	3	0.68(0.11)	0.47(0.08)

Table 9: Power to find target genes in *trans* effects, conditional on the *trans* eQTL being recovered. Only results for *trans* signals of strength half that of the *KLF14 trans* signal shown. Two approaches are compared in this table, T_G^i ignores the missing samples using the approach from section 1.10.1 and T_G^r only uses data for individuals with no missing samples. Results averaged across 50 data sets in which 25% of samples were missing.

4 Method comparisons

4.1 Comparison of tensor decompositions

We have compared our method to the Bayesian Matrix Tensor Factorisation (BMTF) approach, which is a linked decomposition of an arbitrary number of matrices and tensors (Khan and Kaski, 2014; Khan et al., 2014). For the case of decomposing a single 3D array BMTF performs a PARAFAC decomposition (see Eqn 1), with a prior encouraging sparsity on the loadings matrix given by,

$$\begin{aligned} x_{cl} &\sim h_c \delta_0 + (1 - h_c)\mathcal{N}(x_{cl}|0, \alpha_{cl}^{-1}), \\ h_c &\sim \mathcal{B}\text{ernoulli}(\pi_c). \end{aligned} \tag{47}$$

The spike and slab distribution in equation 47 has a mixing parameter, $h_c \in \{0, 1\}$, which determines the activity of each component. If $h_c = 0$ then the whole component is shrunk to zero; if $h_c = 1$, then the prior on the loadings vector reduces to an element-wise ARD prior. (In comparison, SDA uses a spike and slab distribution to obtain element-wise sparsity.) BMTF places a Beta hyperprior on π_c and a Gamma prior on the variance parameters α_{cl}. Standard normal priors are specified for the individual and tissue scores matrices. A homogeneous Gaussian noise model is assumed such that $\epsilon_{nlt} \sim \mathcal{N}(0, \sigma^2)$. Inference is performed using Gibbs sampling. Code for BMTF written in `R` can be downloaded from `http://research.cs.aalto.fi/pml/software/bmtf/`.

4.1.1 Data simulation

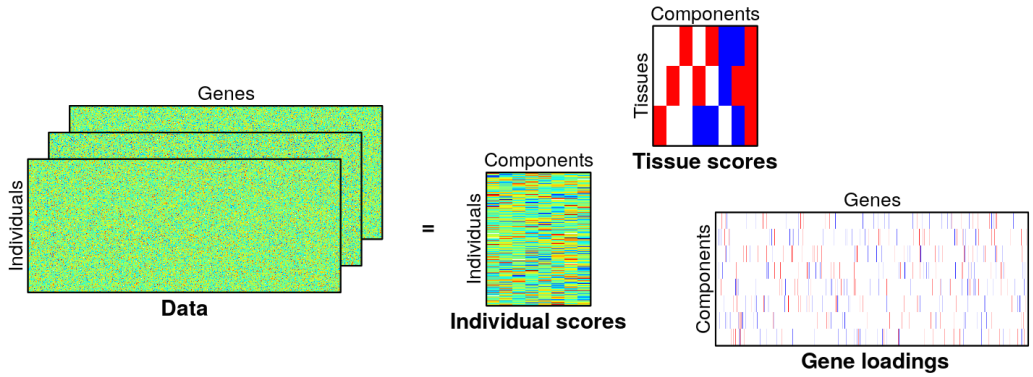
Data was simulated under the PARAFAC model,

$$y_{nlt} = \sum_{c=1}^{C=8} a_{nc} b_{tc} x_{cl} + \epsilon_{nlt} \tag{48}$$

with $C = 8$ components and dimensions $N = 200$ individuals, $L = 500$ genes and $T = 3$ tissues. Three components were simulated to be active in a single tissue, a further three were active in 2 tissues and the remaining two components were active in all tissues. If a component c was active in tissue t then b_{tc} was randomly sampled from $\{-1, 1\}$, otherwise, b_{tc} was set to zero. An example data set showing the pattern of zeros in B is given in Figure 41.

The loadings vectors (rows of X) were simulated to be sparse, with an element set to zero (with probability $1-p$) or drawn from $\mathcal{N}(0, 1)$ (with probability p). The parameter p determines the fraction of non-zero elements in X. This choice of distribution for the loading vectors does favour SDA, as this is exactly the spike and slab that SDA fits. However, it is also an obvious choice for simulating sparse vectors (e.g. Zhao et al., 2014). The individual scores matrix A was drawn from $\mathcal{N}(0, 1)$. Finally, a homogeneous noise model was used, with ϵ_{nlt} drawn from $\mathcal{N}(0, 10)$. SDA uses a more general non-homogeneous noise model, so these simulations are closer to the assumptions of the BMTF approach. Nevertheless, SDA still outperforms BMTF (see below).

Data sets with increasing levels of sparsity were simulated with 50 data sets generated for each value of $p \in \{0.5, 0.4, 0.3, 0.2, 0.1\}$.



Supplementary Figure 41: Example of a data set simulated under the PARAFAC model with $p = 0.1$ (noise not shown).

4.1.2 Post-processing and metrics

For $method \in \{\text{SDA}, \text{BMTF}\}$, denote the estimated individual scores matrix, tissue scores matrix and loadings matrix by A^{method}, B^{method} and X^{method} respectively. The true set of scores and loadings matrices are given by A^{truth}, B^{truth} and X^{truth}.

Number of estimated components: Both methods can automatically shrink components to zero to estimate the true number of underlying components, i.e. perform model selection. Both methods were initialised with 16 components and the number of estimated components recorded to compare performance.

A set of 8 estimated components is required for the following post-processing steps. If more than 8 components were estimated, extra components were removed to leave the set most correlated with the true individual scores. If fewer than 8 components were estimated, then additional components consisting of all zeros were used to make up the difference.

Permutation indeterminacy: Both models have a scaling and permutation indeterminacy. This means that the estimated components will not necessarily be in the same order as the true components, and a direct comparison can not be made. An exhaustive search was performed to find the permutation of the estimated components which best matched the truth. The optimal permutation was selected to maximise the average (absolute) correlation between the true and estimated individual scores vectors. Once the optimal permutation was recovered, the signs of the estimated components were flipped (if necessary) so that correlations were positive. Correlations involving zero components were taken to be 0.

Root mean squared error (RMSE): Root mean squared error (RMSE) was used to evaluate similarity between the true and estimated individual scores vectors. In addition to the optimal permutation, RMSE requires that vectors be on the same scale. Scaling was performed so that the estimated and true scores vectors both had unit variance. RMSE between the true and estimated scores matrices (after a permutation and scaling) was defined as

$$\text{RMSE} = \sqrt{\text{mean}\big((A^{\text{truth}} - A^{\text{method}})^2\big)}. \tag{49}$$

Sparse stability index (SSI): In some cases, if the set of estimated components is very poor, it can be hard to find the best permutation. The sparse stability index (SSI) is invariant to scaling and permutation (Gao et al., 2013).

Let $\Sigma \in \mathbb{R}^{C\times C}$ be a matrix such that Σ_{ck} is the absolute correlation between the cth true individual scores vector and the kth estimated individual scores vector. Additionally let $\mathbf{s}^r \in \mathbb{R}^C$ and $\mathbf{s}^c \in \mathbb{R}^C$ be row and column means of Σ respectively. The sparse stability index is defined as

$$\begin{aligned}\text{SSI} = \frac{1}{2C}\sum_{i=1}^{C}\left[\max\big(\Sigma_{i\cdot}\big) - \frac{\sum_{j=1}^{C}\mathbb{1}\big(\Sigma_{i,j} > \mathbf{s}_i^r\big)\Sigma_{ij}}{C-1}\right] \\ + \frac{1}{2C}\sum_{j=1}^{C}\left[\max\big(\Sigma_{\cdot j}\big) - \frac{\sum_{i=1}^{C}\mathbb{1}\big(\Sigma_{i,j} > \mathbf{s}_i^c\big)\Sigma_{ij}}{C-1}\right],\end{aligned} \tag{50}$$

where $\mathbb{1}(\cdot)$ is an indicator function; it takes a value of 1 if the condition in the brackets is true and a value of zero otherwise.

Equation 50 penalises cases in which there is more than one large number in each row or column of Σ. This occurs if one of the true components is split across two components in the estimated set. The metric also penalises the case where there are no large numbers in a row or column. This arises if the estimated set of components misses one of the true components, or, if the estimated set contains a component that does not exist in the true components. A higher SSI implies a better correspondence between the two input matrices.

Receiver operating characteristic (ROC curve) Finally, the two methods were compared by evaluating recovery of the correct set of non-zero elements in the loadings matrix. Neither SDA not BMTF produce exact sparsity. For SDA, posterior inclusion probabilities (PIPs) were used to threshold X^{SDA} to create a set of genes with non-zero loadings. PIPs tended to be at the extremes of the set $[0, 1]$, i.e. either close to 0 or close to 1. A threshold of 0.5 was used, but any threshold in (0.1, 0.9) gave very similar results. It is less clear how to threshold the estimates for BMTF as different thresholds on X^{BMTF} give rise to very different levels of sparsity. Rather then selecting an arbitrary threshold, a wide range of thresholds were tried, varying between the extreme cases of all zeros to no zeros in X^{BMTF}. For each threshold, power and false positive rates (FPR) were calculated, and plotted to create an ROC curve. Power and FPR point estimates for SDA were also evaluated.

Power and false positive rates (FPR) were calculated as follows,

$$\text{Power} = \frac{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(\hat{X}_{cl}^{method} \neq 0 \text{ and } \hat{X}_{cl}^{truth} \neq 0\big)}{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(\hat{X}_{cl}^{truth} \neq 0\big),} \tag{51}$$

$$\text{FPR} = \frac{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(\hat{X}_{cl}^{method} \neq 0 \text{ and } \hat{X}_{cl}^{truth} = 0\big)}{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(\hat{X}_{cl}^{truth} \neq 0\big).} \tag{52}$$

A summary of the metrics used is give in Table 10.

4.1.3 Run settings

Both methods were initialised with 16 components, double the true number of components. BMTF was run using the default settings and the posterior mean used as a point estimate. SDA was run using settings given in 1.8. SDA was run 10 times and the set of estimates with the highest negative free energy selected.

Metric	Description
Number of components estimated	Evaluates model selection.
Root mean squared error (RMSE) for individual scores matrices	Measures the absolute difference between true and estimated matrices. Requires permutation and scaling of estimated components.
Sparse stability index (SSI) for individual scores matrices	Measure of how similar two component sets are. Invariant to permutations and scalings.
ROC curve (power and false positive rate)	Evaluates recovery of sparsity in loadings matrices. Requires a permutation of the estimated components.

Table 10: Summary of metrics used for method comparison.

4.1.4 Results

	p	Number of estimated components										
		1	2	3	4	5	6	7	8	9	10	11
SDA	0.5								21	19	8	2
	0.4								38	9	3	
	0.3								47	3		
	0.2								46	4		
	0.1					5	28	13	4			
BMTF	0.5								50			
	0.4			1	1	2	1	3	42			
	0.3			9	10	4	6	6	15			
	0.2		1	36	9	3	1					
	0.1	3	8	32	7							

Table 11: Frequency table showing the number of component recovered by SDA and BMTF across 50 data sets; the true number of components is 8. p determines the sparsity of the true components; sparsity increases as p decreases.

Table 11 is a frequency table showing the number of components estimated by each method across 50 data sets. The behaviour of the two methods changes as the sparsity levels increase. When the simulated data set contains dense components ($p = 0.5$), SDA often overestimates the number of components. This is presumably a unwanted effect of adding an extra level of hierarchy into the spike and slab, making it harder to remove components. The performance of SDA improves as the sparsity increases however, with the correct number of component being estimated more often. For high sparsity ($p = 0.1$), the performance starts to decrease again with too few components being estimated. This is unsurprising as the signal-to-noise ratios are decreasing with increasing sparsity. BMTF accurately estimates the number of components 100% of the time when the components are dense ($p = 0.5$). However, for sparser components, this approach tends to underestimate the number of components. This may be a result of the component-level spike and slab removing components too aggressively.

Figure 14 shows the RMSE for estimated sets of individual scores. Each boxplot summarises results from across 50 data sets. As the sparsity increases, the performance of both models decay, but SDA performs substantially better at low levels of sparsity. The multi-

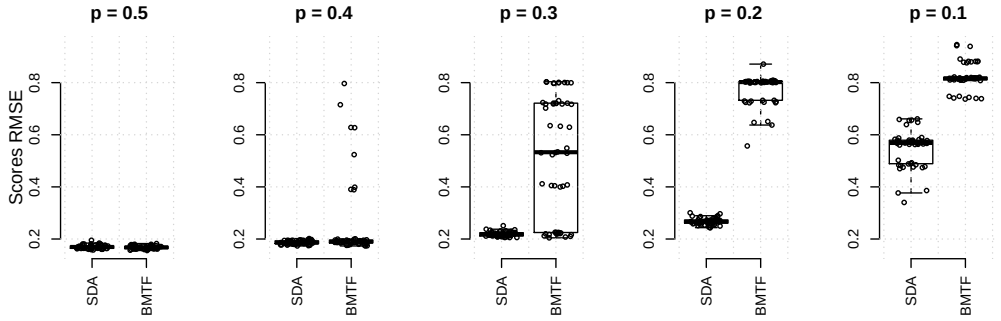


Figure 14: Root mean squared error for individual scores matrices. Sparsity increases from left to right. Boxplots summarise results for 50 data sets.

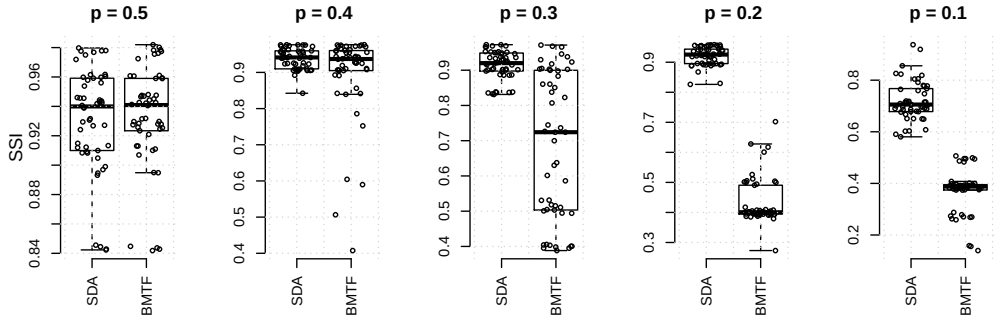


Figure 15: Sparse stability index (SSI) for individual scores matrices. Sparsity increases from left to right. Boxplots summarise results for 50 data sets.

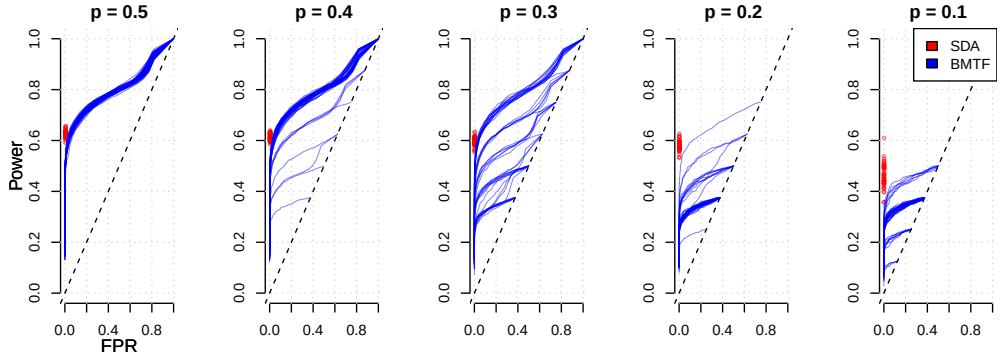


Figure 16: ROC curve for recovery of non-zero elements in the loadings matrices. Sparsity increases from left to right. Blue lines show power and false positive rates for different thresholds on the estimated loadings from BMTF.

modal distribution appearing in some of the boxplots is caused by underestimation of the number of components. BMTF performs less well at higher sparsity levels because of this. SSI shows a similar pattern of behaviour to RMSE (see Figure 15).

The power and FPRs for recovery of non-zero elements in the estimated loadings matrices are shown in Figure 16. The spike and slab distribution used in SDA appears to perform better at feature selection than the ARD prior used in BMTF.

4.2 Comparison of group decompositions

As described in section 1.12 our more general method can carry out group factor analysis. We compared this version of SDA with three other methods from the literature, BGFA (Zhao et al., 2014), CCAGFA (Klami et al., 2014a), and iClusterPlus (Mo et al., 2013). All four methods are based on group factor analysis,

$$y_{nl}^{(d)} = \sum_{c=1}^{C} a_{nc} x_{cl}^{(d)} + \epsilon_{nl}^{(d)} \tag{53}$$

for data $Y^{(d)} \in \mathbb{R}^{N \times L_d}$. Equation (53) decomposes several linked matrices to uncover individual and shared structure. The methods differ in the assumptions they make on the loadings matrices, noise and the inference scheme.

4.2.1 Method descriptions

BGFA: Bayesian Group Factor Analysis (BGFA) (Zhao et al., 2014) employs a three-parameter beta prior to encourage sparsity in the loadings matrices (Armagan et al., 2011; Gao et al., 2013). Briefly, this distribution extends the Beta distribution, adding another parameter to allow it to model a wider range of densities. The BGFA formulation explicitly shrinks globally, at a factor level and at an element level in the loadings matrix,

$$\begin{aligned} x_{cl}^{(d)} &\sim \mathcal{N}(0, \theta_{cl}^{(d)}), \\ \theta_{cl}^{(d)} &\sim \pi^{(d)} \mathcal{G}(g, \delta_{cl}^{(d)}) + (1 - \pi^{(d)}) \delta(\theta_{cl}), \\ \pi^{(d)} &\sim \mathcal{B}\text{eta}(1, 1). \end{aligned} \tag{54}$$

$\pi^{(d)}$ defines global shrinkage, if $\theta_{cl}^{(d)}$ takes a very small value, then $x_{cl}^{(d)}$ will also be small. Heteroscedastic noise is assumed in this model.

Inference for this model has two steps. First, a Gibbs sampler is run to find a good set of initial estimates. These are then used as input for a variational expectation maximization algorithm which finds maximum a posteriori estimates. The code is available for download from `http://beehive.cs.princeton.edu/software/`.

CCAGFA: CCAGFA is an R package[1] that implements several different models (canonical correlation analysis and group factor analysis) described in Klami et al. (2014a,b) and Virtanen et al. (2011, 2012). The method used here is group factor analysis from (Klami et al., 2014a), referred to as CCAGFA from now on.

CCAGFA places an ARD prior on the loadings matrices to encourage sparsity,

$$x_{cl}^{(d)} \sim \mathcal{N}(0, (\alpha_c^{(d)})^{-1}) \tag{55}$$

[1] `http://cran.r-project.org/web/packages/CCAGFA/`

A noise model, $E^{(d)} \sim \mathcal{N}(e^{(d)}|0, (\tau^{(d)})^{-1})$, with a different precision variable for each data type is employed. Inference is performed using variational Bayes. Note that CCAGFA is similar to BMFT with the use of an ARD prior, although CCAGFA learns one precision variable per component, not per element. Also, CCAGFA does not use a spike and slab prior to switch off components, however components can be shrunk to zero if the component precision grows very large.

iClusterPlus: iClusterPlus is a method for joint clustering of multidimensional data (Mo et al., 2013; Shen et al., 2009, 2013). The approach performs several generalised linear regressions, with latent variables $\mathbf{z}_n$ common to all regressions. The framework explicitly allows for binary, categorical and continuous input data types by modelling the data with Binomial, multinomial, and Gaussian random variables respectively. When the input data is continuous, as in these simulations, the model is,

$$y_{nl}^{(d)} \sim \mathcal{N}(y_{nl}^{(d)}|\mathbf{z}_n\boldsymbol{\beta}_l^{(d)}, (\sigma_l^{(d)})^2), \tag{56}$$

$$\mathbf{z}_n \sim \mathcal{N}(0, 1), \tag{57}$$

assuming the data has zero mean. A lasso penalty is placed on the loadings vectors, with tuning parameters γ, (γ can also be data type specific), to give the following penalised likelihood,

$$\max_{\boldsymbol{\beta}_l^{(d)}} \; l(y_{nlt}, \mathbf{z}_n; \boldsymbol{\beta}_l) - \sum_d \gamma||\boldsymbol{\beta}_l^{(d)}||_1. \tag{58}$$

Penalised likelihood estimation is performed using a Monte-Carlo Newton-Raphson algorithm. iClusterPlus can be thought of as a non-Bayesian version of group factor analysis. Regression coefficients $\boldsymbol{\beta}_l$ and latent variables $\mathbf{z}_n$, have a similar interpretation to the gene loadings and individual scores in equation 53 respectively. iClusterPlus is available as an R package[2].

Table 12 summarises the differences between these methods.

Model	**Sparsity**	**Inference**
SDA	Spike and slab prior	Variational Bayes
CCAGFA	ARD prior	Variational Bayes
BGFA	Three parameter Beta prior	Gibbs sampling then variational expectation maximisation
iClusterPlus	Lasso penalty	Monte-Carlo Newton-Raphson

Table 12: Summary of the main features of the methods being compared.

4.2.2 Data simulation

Data was simulated for $D = 3$ data types and $N = 200$ individuals; each data type consisted of L=500 variables. Data was generated under the group factor analysis model as a linear combination of $C = 8$ underlying components and additive noise as in equation (53).

[2] `https://www.bioconductor.org/packages/release/bioc/html/iClusterPlus.html`

Of the 8 components simulated, three were active in just one data type, a further three were active in two of the data types and the remaining two were active in all three data types. (A component is said to be 'active' in a data type if it contributes to the variance in that data matrix.) Figure 17 summarises the component activity patterns. If a component was not active in a particular data type, then the relevant row of the loadings matrix was set to zero. For components which were active, their loadings vectors were sparse, with 90% of the elements equal to 0. The non-zero elements were drawn from $\mathcal{N}(0,1)$.

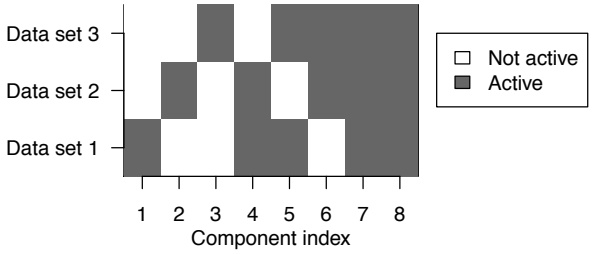


Figure 17: Patten of component activity across data types.

Elements in the individual scores matrix A were also drawn from $\mathcal{N}(0,1)$. Finally, a homoscedastic noise model with precision λ was used, i.e. $\epsilon_{nl}^{(d)} \sim \mathcal{N}(0, \lambda^{-1})$ for $d \in \{1,2,3\}$. In order to evaluate the performance of the methods at different signal-to-noise levels, data was simulated with three different values of λ, (0.1, 0.2 and 0.3).

For each of the three noise levels, 50 data sets were simulated. Signal-to-noise ratios for each data set were calculated as follows,

$$SNR = \frac{\sum_d \text{trace}\big((AX^{(d)})(AX^{(d)})^t\big)}{\sum_d \text{trace}(E^{(d)}E^{(d)t})}. \tag{59}$$

Values of $\lambda = 0.3, 0.2$ and 0.1 corresponded to signal-to-noise ratios (averaged over 50 data sets) of 0.15, 0.1 and 0.05 respectively.

4.2.3 Run settings

All methods can estimate the number of underlying components in the data, however for simplicity, the models were initialised using the true number of components. If fewer than 8 components were estimated, then components consisting of zeros were added to make up the difference.

Default parameter settings for BGFA, CCAGFA and iClusterPlus were used. SDA was run using parameter settings as described in previous sections. SDA, BGFA and CCAGFA were all run 10 times and the set of estimates that resulted in the largest value of the negative free energy selected. Posterior means were used as point estimates for SDA and CCAGFA. BGFA evaluates a maximum a posteriori estimate.

The iClusterPlus algorithm requires selection of a tuning parameter (γ) which determines the sparsity of the resulting components. Rather than selecting a single value for γ, iClusterPlus was run 20 times with different values for γ in $[0,1]$. This range covered the extreme cases of complete sparsity to very dense. As this procedure was slow, results for iClusterPlus were only performed for one noise level, $\lambda = 0.1$.

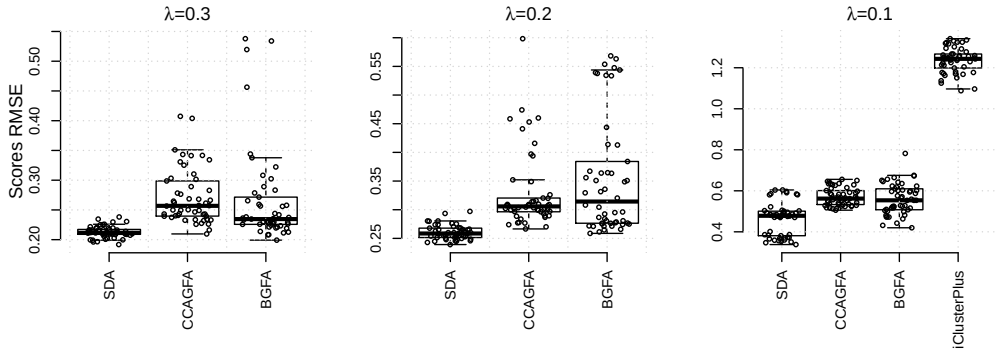


Figure 18: Root mean squared error (RMSE) for recovered individual scores matrices. Boxplots summarise results from across 50 data sets. Noise levels in the simulated data sets increase from left to right.

4.2.4 Post-processing and metrics

Estimated and true loadings matrices were concatenated to create matrices of dimensions C by $3L$, then the post-processing steps described in section 4.1.2 were used. Sparse estimates for SDA were obtained by thresholding PIPs at 0.5. BGFA generated sparse loadings so no thresholding was required. CCAGFA does not give exact sparsity so a sequence of thresholds was used to create ROC curves (as for BMTF in the previous simulations). iClusterPlus gives sparse estimates; power and FPRs were calculated for each value of γ and plotted to create an ROC curve. For RMSE and SSI statistics, the value of γ that resulted in a FPR of 0.01 was used.

4.2.5 Results

Figures 18 and 19 show RMSE and SSI for the estimated individual scores matrices at 3 different noise levels. The multi-modal distribution seen in some of these boxplots is again due to an incorrect number of components being recovered. RMSE and SSI degrade as the noise levels increase, with SDA outperforming BGFA and CCAGFA. iClusterPlus does not recover the individual scores matrix as well as the other approaches. However, it should be noted that the tuning parameter was selected to optimise metrics on the loadings matrices rather then individual scores matrix.

Power and FPRs for the recovery of non-zero elements in the loadings matrices are shown in the ROC curves in Figure 20. SDA, CCAGFA and BGFA perform similarly although at low levels of sparsity ($\lambda = 0.1$) it appears that SDA slightly outperforms the other methods. BGFA appears to shrink aggressively, resulting in a very low FPRs at the expense of power. SDA on the other hand shrinks slightly less strictly resulting in more power but higher FPRs compared to BGFA. Performance of CCAGFA depends heavily on the threshold, but performs well at some thresholds. iClusterPlus does not recover sparsity in the loading matrices as well as the other methods.

References

A. Armagan, D. B. Dunson, and M. Clyde (2011). "Generalized Beta Mixture of Gaussians". *Advances in Neural Information Processing Systems* 24, pp. 523–531.

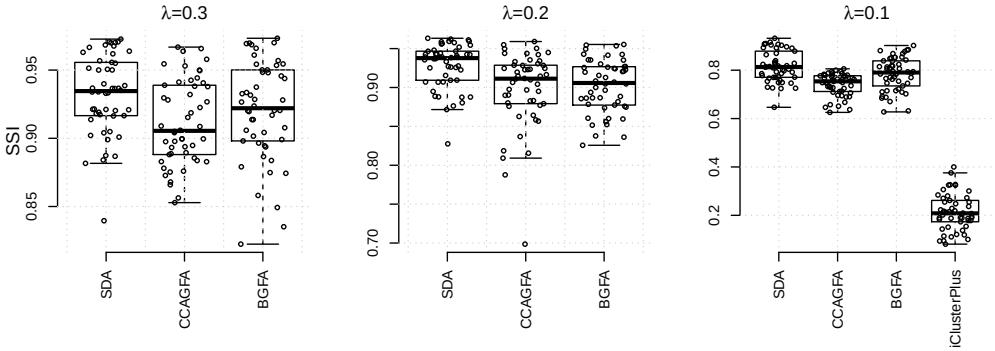


Figure 19: Sparse stability index (SS1) for recovered individual scores matrices. Boxplots summarise results from across 50 data sets. Noise levels in the simulated data sets increase from left to right.

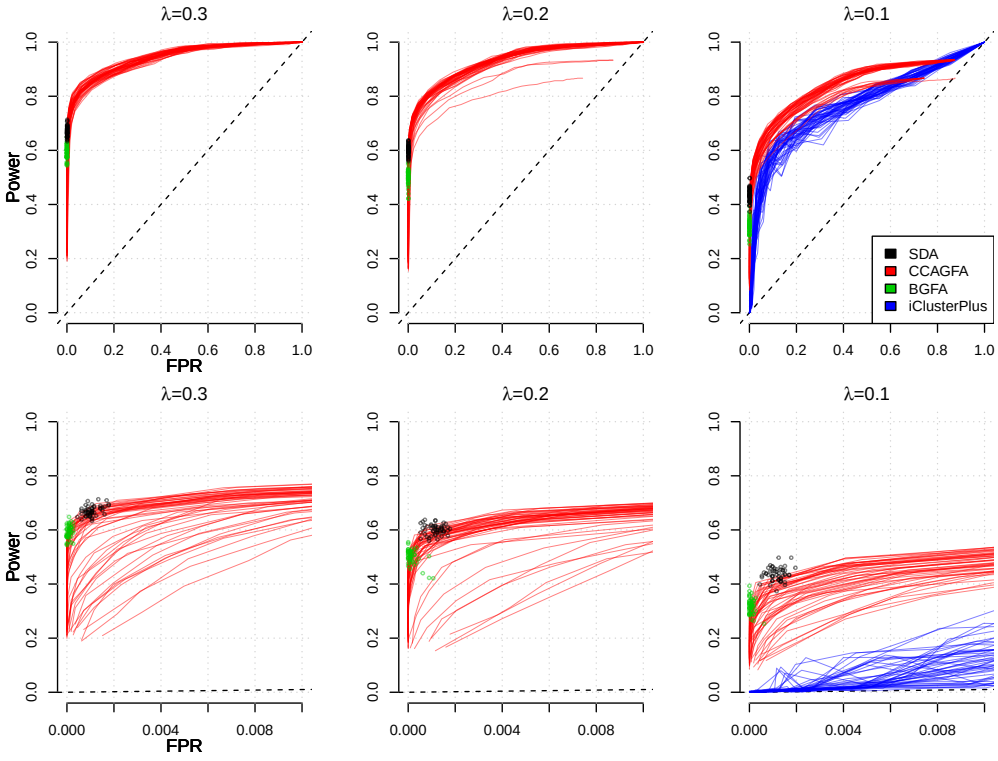


Figure 20: ROC curves for recovery of non-zero elements in the loadings matrices. Point estimates for SDA obtained by thresholding the PIPs at 0.5. No thresholding was required for BGFA as the raw estimates have exact sparsity. Results for CCAGFA were generated using a range of thresholds on the same estimate set, resulting in an ROC curve. The ROC curve for iClusterPlus was generated by running the method multiple times with different tuning parameters. Noise levels in the simulated data sets increase from left to right. The top and bottom rows of plots only differ in their x-axis range.

J. D. Carroll and J.-J. Chang (1970). "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition". *Psychometrika* 35.3, pp. 283–319.

E Fokoue (2004). "Stochastic determination of the intrinsic structure in Bayesian factor analysis". *Statistical and Applied Mathematical Sciences Institute*.

C. Gao, C. D. Brown, and B. E. Engelhardt (2013). "A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects". *arXiv:1310.4792v1*, pp. 1–28.

A. R. Groves, C. F. Beckmann, S. M. Smith, and M. W. Woolrich (2011). "NeuroImage Linked independent component analysis for multimodal data fusion". *NeuroImage* 54.3, pp. 2198–2217.

R. A. Harshman and M. E. Lundy (1994). "PARAFAC: Parallel factor analysis". *Computational Statistics & Data Analysis* 18.1, pp. 39–72.

A. Hyvärinen (1999). "Fast and robust fixed-point algorithms for independent component analysis." *IEEE transactions on neural networks* 10.3, pp. 626–34.

S. A. Khan and S. Kaski (2014). "Bayesian Multi-View Tensor Factorization". *Machine Learning and Knowledge Discovery in Databases, ECML PKDD*, pp. 656–671.

S. A. Khan, E. Leppäaho, and S. Kaski (2014). "Multi-tensor factorization". *arXiv:1412.4679v1*, p. 22.

A. Klami, S. Virtanen, E. Leppäaho, and S. Kaski (2014a). "Group Factor Analysis". *Neural Networks and Learning Systems, IEEE Transactions* 26.9.

A. Klami, G. Bouchard, and A. Tripathi (2014b). "Group-sparse Embeddings in Collective Matrix Factorization". *arXiv:1312.5921v2*, p. 2.

W. Lian, P. Rai, E. Salazar, and L. Carin (2015). "Integrating Features and Similarities : Flexible Models for Heterogeneous Multiview Data". *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. Nevins, et al. (2006). "Sparse statistical modelling in gene expression genomics". *Bayesian Inference for Gene Expression and Proteomics* 1.

T. Mitchell and J. Beauchamp (1988). "Bayesian Variable Selection in Linear Regression". *Journal of the American Statistical Association* 83.404, pp. 1023–1032.

Q. Mo, S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, et al. (2013). "Pattern discovery and cancer gene identification in integrated cancer genomic data." *Proceedings of the National Academy of Sciences of the United States of America* 110.11, pp. 4245–50.

M. Rotival, T. Zeller, P. S. Wild, S. Maouche, S. Szymczak, et al. (2011). "Integrating Genome-Wide Genetic Variations and Monocyte Expression Data Reveals Trans-Regulated Gene Modules in Humans". *PLoS Genetics* 7.12, e1002367.

R. Shen, A. B. Olshen, and M. Ladanyi (2009). "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis." *Bioinformatics* 25.22, pp. 2906–12.

R. Shen, S. Wang, and Q. Mo (2013). "Sparse integrative clustering of multiple omics data sets". *The Annals of Applied Statistics* 7.1, pp. 269–294.

K. S. Small, A. K. Hedman, E. Grundberg, A. C. Nica, G. Thorleifsson, et al. (2011). "Identification of an imprinted master trans regulator at the KLF14 locus related to multiple metabolic phenotypes." *Nature genetics* 43.6, pp. 561–564.

O. Stegle, L. Parts, R. Durbin, and J. Winn (2010). "A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies." *PLoS computational biology* 6.5, e1000770.

M. Titsias and M. Lázaro-Gredilla (2011). "Spike and slab variational inference for multi-task and multiple kernel learning". *Neural Information Processing Systems*, pp. 1–9.

S. Virtanen, A. Klami, and S. Kaski (2011). "Bayesian CCA via group sparsity". *Proceedings of the 28th International Conference on Machine Learning*, pp. 2339–2347.

S. Virtanen, A. Klami, S. A. Khan, and S. Kaski (2012). "Bayesian group factor analysis". *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pp. 1269–1277.

S. Zhao, C. Gao, S. Mukherjee, and B. E. Engelhardt (2014). "Bayesian group latent factor analysis with structured sparse priors". *arXiv:1441.2698v1*.