Using DIGGIT, a package for Infering Genetic Variants Driving Cellular Phenotypes

James Chen, Mariano J. Alvarez, Andrea Califano
Department of Systems Biology, Columbia University, New York, USA

October 24, 2025

1 Overview of DIGGIT

Identification of somatic mutations and germline variants that are determinants of cancer and other complex human diseases/traits (driver mutations) is mostly performed on a statistical basis, using models of genomic evolution [1] or mutational bias [2], to increase the significance of individual events. Achieving appropriate statistical power, however, requires large effect sizes or large cohorts due to multiple hypothesis testing correction [3]. In addition, these approaches are not designed to provide mechanistic insight. As a result, many disease risk determinants, such as apolipoprotein E, were discovered long before they were mechanistically elucidated [4].

Network-based analyses have recently emerged as a highly effective framework for the discovery of Master Regulator (MR) genes that are functional disease drivers [5, 6, 7, 8]. Here, we present the R implementation of DIGGIT (Driver-gene Inference by Genetical-Genomic Information Theory), an algorithm to identify genetic determinants of disease by systematically exploring regulatory/signaling networks upstream of MR genes. This collapses the number of testable hypotheses and provides regulatory clues to help elucidate associated mechanisms. We have applied DIGGIT to identify causal genetic determinants of the mesenchymal subtype of human glioma [9].

2 Citation

Chen JC, Alvarez MJ, Talos F, Dhruv H, Rieckhof GE, Iyer A, Diefes KL, Aldape K, Berens M, Shen MM, Califano A. Identification of Causal Genetic Drivers of Human Disease through Systems-Level Analysis of Regulatory Networks, Cell (2014) 159(2):402-14. `http://dx.doi.org/10.1016/j.cell.2014.09.021`.

3 Installation of *diggit* package

In order to install the *diggit* package, the user must first install R (`http://www.r-project.org`). After that, *diggit* and the required data package for the examples and this vignette (*diggitdata*) can be installed with:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install(c("diggitdata", "diggit"))
```

4 Example session to analyze the data provided by the *diggitdata* package

4.1 Getting started

After installing *diggit* and *diggitdata* packages, diggit can be loaded by

```
> library(diggit)
```

4.1.1 The *diggitdata* data package

The data distributed in the *diggitdata* package is required to execute the code provided in this vignette. The package consists on several datasets and regulatory networks contained in five R-image files:

gbm.expression Normalized human glioma expression data for 245 samples from TCGA, including metadata indicating tumor subtype, in an object of class ExpressionSet (see Biobase package from Bioconductor for a description of the ExpressionSet class).

gbm.cnv Normalized copy number variation data for 230 samples from TCGA

gbm.cnv.normal Normalized copy number variation data for 33 normal (blood) samples from TCGA

gbm.aracne Transcriptional regulatory network assembled by the ARACNe [10] algorithm from glioma expression data

gbm.mindy Post-translational regulatory network assembled by the MINDy [11] algorithm, limited to three transcription factors: STAT3, CEBPB and CEBPD.

4.2 Loading the data and generating a diggit-class object

The data provided in the *diggitdata* package can be loaded into memory with:

```
> data(gbm.expression, package="diggitdata")
> data(gbm.cnv, package="diggitdata")
> data(gbm.aracne, package="diggitdata")
> data(gbm.mindy, package="diggitdata")
```

In order to reduce the computer time, we are going to consider only 1,000 genes for the CNV analysis.

```
> genes <- intersect(rownames(gbmExprs), rownames(gbmCNV))[1:1000]
> gbmCNV <- gbmCNV[match(genes, rownames(gbmCNV)), ]
```

The diggit-class objects are containers for both the input data and the output results from the diggit algorithm. This allows for the individual results of each step of the pipeline to be stored in an appropriate format. This object will be updated sequentially as each step of DIGGIT is completed, as shown below.

We can create an object of class "diggit" and store the required data with:

```
> dobj <- diggitClass(expset=gbmExprs, cnv=gbmCNV, regulon=gbmTFregulon, mindy=gbmMindy)
> dobj

An object of class diggit
Slot expset:
Expression data of 9215 features by 245 samples

Slot cnv:
CNV data of 1000 features by 230 samples
```

```
Slot regulon:
Object of class regulon with 835 regulators, 8365 targets and 183774 interactions

Slot mindy:
Object of class regulon with 157 regulators, 3 targets and 178 interactions

Slot fcnv:
Empty

Slot mr:
Empty

Slot viper:
Empty

Slot aqtl:
Empty

Slot conditional:
Empty
```

4.3 Inferring functional copy number variation (fCNV)

We consider a CNV as functional if it is significantly associated with the expression levels of the altered (amplified/deleted) gene. The `fCNV()` function computes such association and store the results in the *diggit* object. This is done by measuring the statistical association between gene expression and gene copy number, which can be done either by correlation analysis,

```
> dobj <- fCNV(dobj, method="spearman", verbose=FALSE)
> diggitFcnv(dobj)[1:5]

       KLHL9       CHI3L1        CEBPD        RPL39     B4GALNT1
6.421432e-38 7.791331e-03 1.052353e-02 9.098632e-01 3.571987e-06

> head(dobj, 5)$fcnv

      CDKN2A        KLHL9        RRAGA      SIGMAR1      TUBGCP3
3.083411e-39 6.421432e-38 1.246612e-28 5.266522e-28 2.640506e-22
```

or by mutual information (MI),

```
> RNGkind("L'Ecuyer-CMRG")
> set.seed(1)
> #if(tools:::.OStype() == "unix") {
> #    mc.reset.stream()
> #}
> dobj <- fCNV(dobj, method="mi", cores=cores, verbose=FALSE)
> diggitFcnv(dobj)[1:5]

       KLHL9       CHI3L1        CEBPD        RPL39     B4GALNT1
1.050777e-42 1.242175e-03 6.877448e-06 9.652313e-01 5.453770e-12

> head(dobj, 5)$fcnv
```

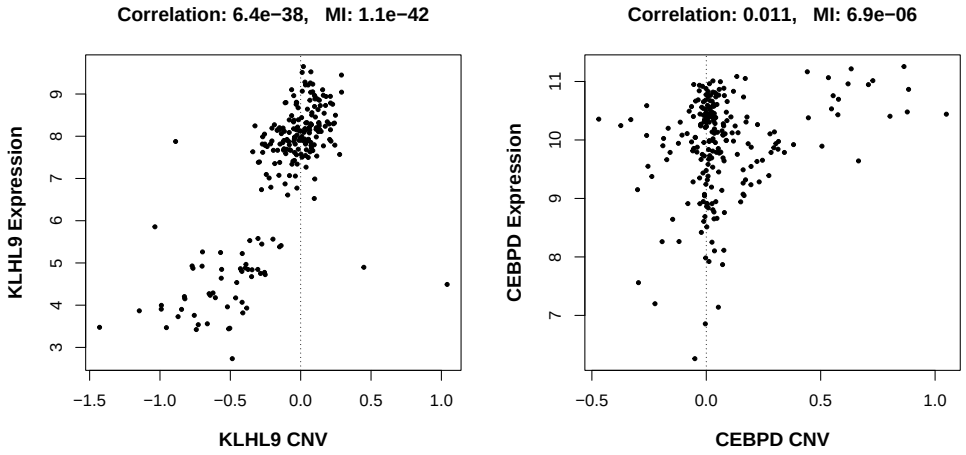

Figure 1: Scatter-plots showing the association between CNV and gene expression for KLHL9 and CEBPD. Correlation and MI p-values are shown over each plot.

```
       KLHL9        CDKN2A          OS9       SIGMAR1        RRAGA
1.050777e-42 1.735741e-39 9.985210e-32 3.343511e-31 1.353660e-30
```

The first three lines of code are meant to set the seed for random numbers generation to a constant. This is necessary to make the permutation process required for computing MI statistical significance reproducible. As seen in the code above, the fCNV slot from an object of class diggit can be retrieved by the function `diggitFcnv()`, while the top n most significant fCNVs can be reported using the function `head()`. In the example, we reported the top 5 most significant fCNVs. Figure 1 shows the association between CNV and expression for KLHL9 and CEBPD, with correlation and MI analysis p-values indicated on top of the figure.

4.4 Master Regulator Analysis

Master Regulators (MR) can be inferred with the `marina()` function. For this example, we infer the MR for the glioma mesenchymal subtype when compared with the proneural subtype. The subtype information is included as metadata in an AnnotatedDataFrame object for this example, which is contained in a Bioconductor ExpressionSet object, together with the expression profile data. Detailed documentation about the ExpressionSet objects can be obtained from the *Biobase* package (Bioconductor).

To correctly leverage this information, we must indicate the metadata column containing the tumor subtype information and the labels for the mesenchymal (MES) and proneural (PN) classes by the parameters `pheno`, `group1` and `group2` of the `marina()` function, as shown below:

```
> set.seed(1)
> library(parallel)
> #if(tools::.OStype() == "unix") {
> #    mc.reset.stream()
> #}
> dobj <- marina(dobj, pheno="subtype", group1="MES", group2="PN", cores=cores, verbose=FALSE)
> head(dobj, 5)$mr

 BHLHE40    CEBPD    LITAF    TEAD3    RCAN1
3.559992 3.424811 3.373444 3.348730 3.332088
```

4.5 Activity quantitative trait loci (aQTL)

fCNVs are then analyzed to identify those whose alteration is predictive of MR activity, similar to expression quantitative trait loci (eQTL) discovery [12]. Activity quantitative trait loci (aQTL) are inferred based on the statistical association between copy number and MR activity. First, single-sample MR activity is inferred using the Virtual Inference of Protein-activity by Enriched Regulon analysis (VIPER) algorithm (see *viper* package from Bioconductor for further details). Then, the association between the inferred protein activity and CNVs can be estimated by correlation or MI analysis. Both steps are implemented by the `aqtl()` function. In the example below, the aQTL for the synergistic regulators of the glioma mesenchymal subtype, CEBPD and STAT3 [6], is computed using MI.

```
> set.seed(1)
> #if(tools:::.OStype() == "unix") {
> #mc.reset.stream()
> #}
> dobj <- aqtl(dobj, mr=c("CEBPD", "STAT3"), method="mi", cores=cores, verbose=FALSE)
> head(dobj, 5)$aqtl

           KLHL9    CHI3L1      CEBPD  B4GALNT1
CEBPD 0.04251126 0.1327128 0.02554002 0.2300000
STAT3 0.01022815 0.5624479 0.04448877 0.2595135
```

4.6 Conditional association analysis

CNVs can span multiple genes, resulting in statistical dependencies equivalent to linkage disequilibrium in classical genetics. Conditional analysis helps assess whether association of a F-CNV (fCNV_i) with the phenotype may be an artifact resulting from its physical proximity to a bona fide driver F-CNV (fCNV_j), in which case conditional association of fCNV_i with the phenotype (i.e., using only fCNV_jWT samples) should not be statistically significant, thus removing such artifacts. Conditional analysis is performed by computing the association between samples harboring CNVs in gene a and sample groups (tumor subtypes in our example), after conditioning for the presence of CNVs in gene b. Because the association analysis is performed by Fisher's exact test (FET), the CNV continuous data should be discretized using an appropiate threshold. This threshold can be obtained from the analysis of normal (blood in this example) samples. We can estimate an appropiate threshold at $\alpha = 0.05$ as follows,

```
> data(gbm.cnv.normal, package="diggitdata")
> cnvthr <- quantile(as.vector(gbmCNVnormal), c(.025, .975), na.rm=TRUE)
```

The conditional analysis can be performed then with the `conditional()` function:

```
> dobj <- conditional(dobj, pheno="subtype", group1="MES", group2="PN", mr="STAT3",
+                     cnv=cnvthr, cores=cores, verbose=FALSE)
> dobj

An object of class diggit
Slot expset:
Expression data of 9215 features by 245 samples

Slot cnv:
CNV data of 1000 features by 230 samples

Slot regulon:
Object of class regulon with 835 regulators, 8365 targets and 183774 interactions
```

```
Slot mindy:
Object of class regulon with 157 regulators, 3 targets and 178 interactions

Slot fcnv:
F-CNV statistical significance for 1000 features

Slot mr:
Master regulator NES for 834 features

Slot viper:
VIPER protein activity matrix for 2 features by 245 samples

Slot aqtl:
aQTL matrix for 2 regulators by 510 genetic alterations

Slot conditional:
Conditional analysis results for 1 MRs and 5 modulators:

STAT3
3 modulators at p<0.05:
KDELR2, HMG20B, COPS6
```

The conditional analysis results can be displayed with the `plot()` function, as shown below and in figure 2, and summarized with the function `summary()`:

```
> plot(dobj, "STAT3", cluster="2")
> summary(dobj)

$STAT3
     KDELR2      HMG20B       COPS6      SEC23B       DEAF1
0.003334976 0.018017915 0.041976217 0.094385293 0.633802817
```

4.7 Limiting the analysis to upstream post-translational modulators

To increase the power of the test, the aQTL analysis can be restricted to evaluate only upstream post-translational modulators of the MRs, as inferred by the MINDy algorithm [11]. The post-translational modulators for STAT3, CEBPB and CEBPD in GBM have been inferred by the CINDy algorithm [13] from 3,540 candidate genes, including signaling pathway-associated citoplasmic proteins and membrane receptors, and distributed as part of the *diggitdata* package. We can use the information provided by this post-translational interactome to reduce the list of candidate potential upstream modulators, to the ones identified by the CINDy algorithm, by setting the *mindy* parameter of the *aqtl* function to TRUE:

```
> set.seed(1)
> #if(tools:::.OStype() == "unix") {
> #    mc.reset.stream()
> #}
> dobj <- aqtl(dobj, mr=c("CEBPD", "STAT3"), method="mi", mindy=TRUE, cores=cores, verbose=FALSE)
> dobj <- conditional(dobj, pheno="subtype", group1="MES", group2="PN", mr="STAT3",
+                     cnv=cnvthr, verbose=FALSE)
> summary(dobj)

$STAT3
       KLHL9
0.0004395437
```

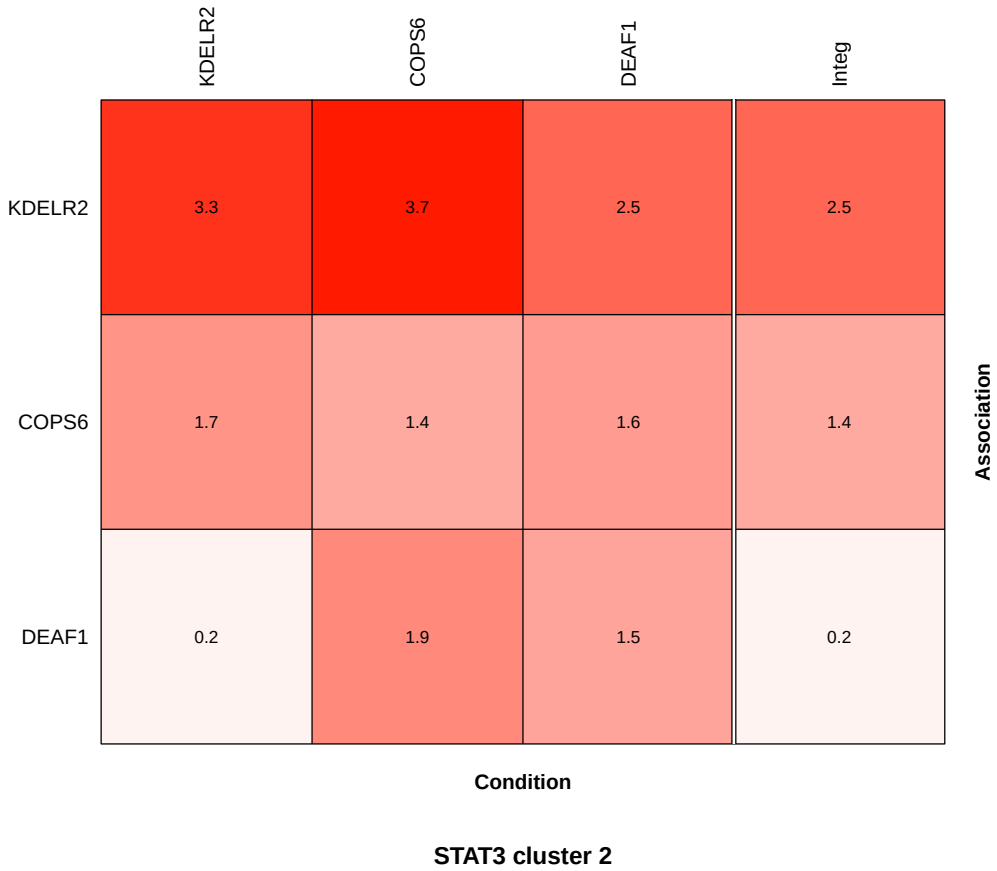

Figure 2: Conditional association analysis for a cluster of CNVs associated with STAT3 activity. The heatmap shows $-log_{10}(p)$ for the association between CNV and tumor subtype (rows) after conditioning on each gene (column).

5 Session information

```
> sessionInfo()

R Under development (unstable) (2025-10-20 r88955)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04.3 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.23-bioc/R/lib/libRblas.so
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0  LAPACK version 3.12.0

Random number generation:
 RNG:     L'Ecuyer-CMRG
 Normal:  Inversion
 Sample:  Rejection

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: America/New_York
tzcode source: system (glibc)

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] diggit_1.41.0       Biobase_2.69.1      BiocGenerics_0.55.4
[4] generics_0.1.4

loaded via a namespace (and not attached):
 [1] ks_1.15.1          plotly_4.11.0      tidyr_1.3.1        class_7.3-23
 [5] KernSmooth_2.23-26 lattice_0.22-7     pracma_2.4.6       digest_0.6.37
 [9] magrittr_2.0.4     grid_4.6.0         RColorBrewer_1.1-3 mvtnorm_1.3-3
[13] fastmap_1.2.0      jsonlite_2.0.0     Matrix_1.7-4       viper_1.43.0
[17] e1071_1.7-16       survival_3.8-3     mclust_6.1.1       httr_1.4.7
[21] kernlab_0.9-33     purrr_1.1.0        viridisLite_0.4.2  scales_1.4.0
[25] lazyeval_0.2.2     cli_3.6.5          rlang_1.1.6        splines_4.6.0
[29] tools_4.6.0        dplyr_1.1.4        ggplot2_4.0.0      vctrs_0.6.5
[33] R6_2.6.1           proxy_0.4-27       lifecycle_1.0.4    htmlwidgets_1.6.4
[37] segmented_2.1-4    MASS_7.3-65        pkgconfig_2.0.3    pillar_1.11.1
[41] gtable_0.3.6       data.table_1.17.8  glue_1.8.0         tibble_3.3.0
[45] tidyselect_1.2.1   mixtools_2.0.0.1   dichromat_2.0-0.1  farver_2.1.2
[49] nlme_3.1-168       htmltools_0.5.8.1  compiler_4.6.0     S7_0.2.0
```

6 References

References

[1] Frattini,V. et al. (2013) The integrated landscape of driver genomic alterations in glioblastoma. Nat. Genet., 45, 1141-9.

[2] Lawrence,M.S. et al. (2013) Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature, 499, 214-8.

[3] Califano, A., et al. (2012) Leveraging models of cell regulation and GWAS data in integrative network-based association studies. Nat. Genet. 44, 841-7.

[4] Liu, C.C., et al. (1023) Apolipoprotein E and Alzheimer disease: risk, mechanisms and therapy. Nat. rev. Neurol. 9, 106-18.

[5] Aytes,A. et al. (2014) Cross-Species Regulatory Network Analysis Identifies a Synergistic Interaction between FOXM1 and CENPF that Drives Prostate Cancer Malignancy. Cancer Cell, 25, 638-51.

[6] Carro,M.S. et al. (2010) The transcriptional network for mesenchymal transformation of brain tumours. Nature, 463, 318-25.

[7] Lefebvre,C. et al. (2010) A human B-cell interactome identifies MYB and FOXM1 as master regulators of proliferation in germinal centers. Mol. Syst. Biol., 6, 377.

[8] Piovan,E. et al. (2013) Direct reversal of glucocorticoid resistance by AKT inhibition in acute lymphoblastic leukemia. Cancer Cell, 24, 766-76.

[9] Chen JC, et al. (2014) Identification of causal genetic drivers of human disease through systems-level analysis of regulatory networks. Cell (In Press).

[10] Margolin,A.A. et al. (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics, 7 Suppl 1, S7.

[11] Wang,K. et al. (2009) Genome-wide identification of post-translational modulators of transcription factor activity in human B cells. Nat. Biotechnol., 27, 829-39.

[12] Yang,X. et al. (2009) Validation of candidate causal genes for obesity that affect shared metabolic pathways and networks. Nat. Genet., 41, 415-23.

[13] Giorgi FM, Lopez G, Woo JH, Bisikirska B, Califano A, Bansal M. Inferring protein modulation from gene expression data using conditional mutual information. PLoS One. 2014;9(10):e109569.