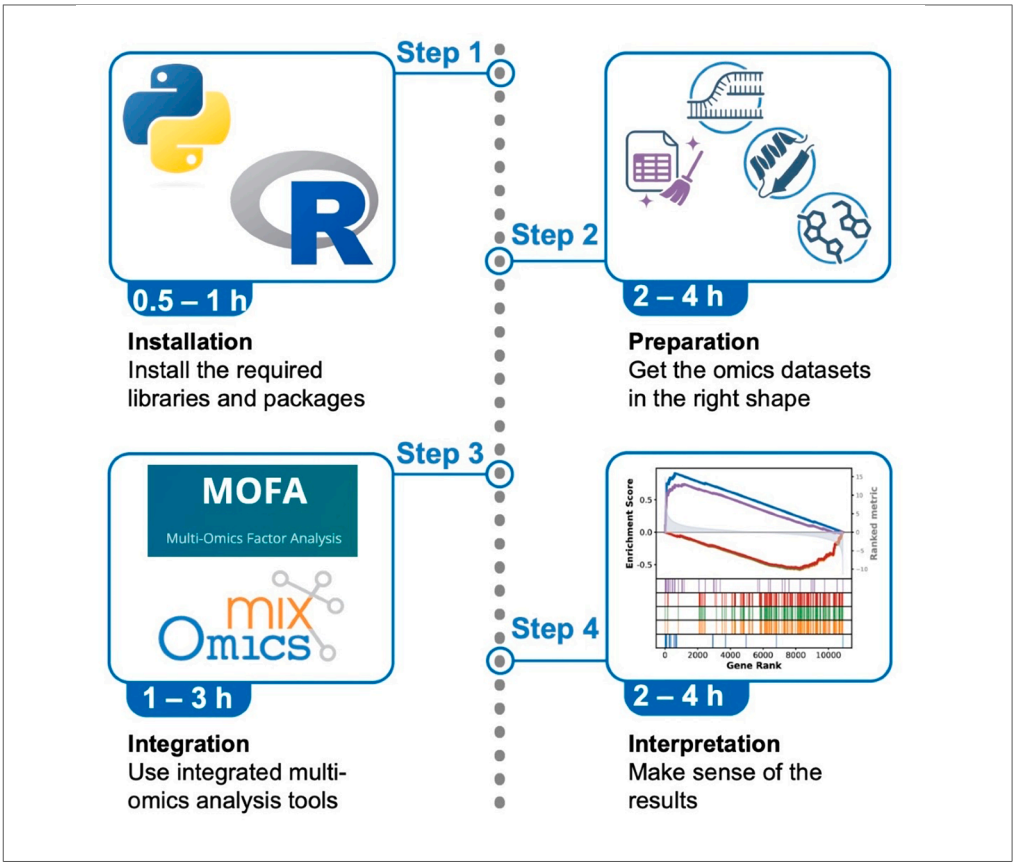
STAR Protocols

CellPress
OPEN ACCESS

Protocol

# Protocol for integrating and interpreting multi-omics data combining unsupervised and supervised data integrating approaches




Multi-omics integration combines data from transcriptomics, proteomics, and metabolomics to provide insights into biological systems. Here, we present a protocol for integrating and interpreting multi-omics data using unsupervised multi-omics factor analysis (MOFA), supervised projection-based integration (DIABLO), along with general single-omics analysis techniques and visualizations. We describe steps for data preparation, model construction, and biological interpretation of multi-omics datasets. These approaches identify coordinated molecular changes across biological layers and reveal regulatory mechanisms that drive biological processes.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Matthias Anagho-Mattanovich, Holda Awah Anagho-Mattanovich, Qian Gao, Thomas Moritz

matthias.mattanovich@sund.ku.dk (M.A.-M.)
thomas.moritz@sund.ku.dk (T.M.)

## Highlights

Instructions for integrating multi-omics datasets

Steps for applying MOFA and DIABLO multi-omics integrated analysis algorithms

Guidance on how to approach an integrated analysis

Protocol for a general integrated analysis and data interpretation across omics layers

Anagho-Mattanovich et al., STAR Protocols 7, 104534 June 19, 2026 © 2026 The Authors. Published by Elsevier Inc.
https://doi.org/10.1016/j.xpro.2026.104534

STAR Protocols

CellPress
OPEN ACCESS

Protocol

# Protocol for integrating and interpreting multi-omics data combining unsupervised and supervised data integrating approaches

Matthias Anagho-Mattanovich,[1,3,*] Holda Awah Anagho-Mattanovich,[2] Qian Gao,[1] and Thomas Moritz[1,4,*]

[1]Novo Nordisk Foundation Center for Basic Metabolic Research, University of Copenhagen, Blegdamsvej 3B, Copenhagen 2200, Denmark

[2]Novo Nordisk Foundation Center for Protein Research, University of Copenhagen, Blegdamsvej 3B, Copenhagen 2200, Denmark

[3]Technical contact

[4]Lead contact

*Correspondence: matthias.mattanovich@sund.ku.dk (M.A.-M.), thomas.moritz@sund.ku.dk (T.M.)
https://doi.org/10.1016/j.xpro.2026.104534

## SUMMARY

**Multi-omics integration combines data from transcriptomics, proteomics, and metabolomics to provide insights into biological systems. Here, we present a protocol for integrating and interpreting multi-omics data using unsupervised multi-omics factor analysis (MOFA), supervised projection-based integration (data integration analysis for biomarker discovery using latent components, DIABLO), along with general single-omics analysis techniques and visualizations. We describe steps for data preparation, model construction, and biological interpretation of multi-omics datasets. These approaches identify coordinated molecular changes across biological layers and reveal regulatory mechanisms that drive biological processes.**

**For complete details on the use and execution of this protocol, please refer to Anagho-Mattanovich et al.[1]**

## BEFORE YOU BEGIN

This protocol requires matched and pre-processed multi-omics datasets (transcriptomics, proteomics, metabolomics) from the same biological samples. Ensure data quality control has been performed on individual omics layers before integration.

### Innovation

The methodology presented here combines unsupervised (MOFA[2,3]) and supervised (DIABLO[4,5]) multi-omics data integration approaches within a single analytical pipeline, enabling both hypothesis-free discovery and targeted biomarker identification. This dual-strategy approach extracts biologically meaningful patterns from high-dimensional multi-omics data while linking statistical findings to mechanistic understanding through pathway enrichment and network analysis.

Current multi-omics studies typically apply either unsupervised or supervised methods in isolation. This protocol demonstrates the synergistic value of combining both: MOFA identifies coordinated variance patterns without prior assumptions, while DIABLO maximizes discrimination between experimental conditions. For more detailed information on the individual frameworks, see the documentations at biofam.github.io/MOFA2 and mixomics.org.

### Institutional permissions

Ensure that all necessary institutional permissions and ethical approvals are obtained for the use of biological samples and data.

### Install required software and packages

Timing: 30–60 min

This protocol requires R (version 4.0 or higher) and Python (version 3.8 or higher) with specific packages for multi-omics integration.

1. Install required R packages.
   a. Open R or RStudio.
   b. Install MOFA2 package from Bioconductor:
   ```
   if (!require("BiocManager", quietly = TRUE))
   install.packages("BiocManager") BiocManager::install("MOFA2")
   ```
   c. Install mixOmics package:
   ```
   BiocManager::install("mixOmics")
   ```
   d. Install additional required R packages.
   ```
   install.packages(c("tidyverse", "pheatmap", "ggplot2"))
   BiocManager::install(c("edgeR", "limma", "ComplexHeatmap"))
   ```
2. Install required python packages.

```
pip install gseapy pandas numpy scipy matplotlib seaborn
```

### Prepare your omics datasets

Timing: 1–2 h

Organize your multi-omics data into properly formatted matrices before beginning the integration analysis.

3. Prepare transcriptomics data.
   a. Obtain bulk RNA-seq read counts aligned to the reference genome using standard pipelines (e.g., nf-core rnaseq).
   b. Process raw counts using, e.g., edgeR to generate CPM (counts per million) values.
   c. Format data as a matrix with genes as rows and samples as columns.
4. Prepare proteomics data.
   a. Process raw mass spectrometry data using MaxQuant or similar software to quantify protein abundances.
   b. Extract LFQ (label-free quantification) intensity values.
   c. Format data as a matrix with proteins as rows and samples as columns.
5. Prepare metabolomics data.
   a. Integrate metabolomics data from multiple platforms (GC-MS, LC-MS) if applicable.
   b. Format data as a matrix with metabolites as rows and samples as columns.

*Note:* Ensure all three omics datasets use the same sample identifiers.

*Note:* Ensure all three omics datasets are obtained from the same level (e.g., bulk/sample level or single-cell level). In theory, the protocol can be applied to single-cell level data as long as all three omics datasets are obtained at the single-cell level. Single-cell RNA-seq data can also be converted into pseudo-bulk data by summing or averaging the counts.

*Note:* Gene/protein/metabolite annotations (ID types) are not critical for data integration but are important for pathway enrichment analysis. Enrichment analysis requires matching between the identifiers in your data and those used in the selected gene/protein/metabolite set database. Therefore, the ID types in your data should be aligned with the database you choose. Common ID types include:

*Note:* Genes/proteins: SYMBOL, ENTREZID, UNIPROT, ENSEMBL, REFSEQ Metabolites: HMDB, KEGG, CAS, DTXCID, DTXSID, SID, CID, ChEBI, DrugBank.

*Note:* Both targeted and untargeted metabolomics data can be used for data integration. However, only known metabolites with valid identifiers can be used for pathway enrichment analysis.

## KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Software and algorithms | | |
| edgeR | Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics *26*, 139–140. https://doi.org/10.1093/bioinformatics/btp616. | RRID:SCR_012802 |
| limma | Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. *43*, e47.https://doi.org/10.1093/nar/gkv007. | RRID:SCR_010943 |
| R 4.2.2 | R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/. | RRID:SCR_001905 |
| Python 3.9.7 | https://docs.python.org/3.9/reference/index.html | RRID:SCR_008394 |
| GSEApy | Fang, Z., Liu, X., and Peltz, G. (2023). GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. Bioinformatics *39*.https://doi.org/10.1093/bioinformatics/btac757. | RRID:SCR_025803 |
| mixOmics | Rohart, F., Gautier, B., Singh, A., and Lê Cao, K.-A. (2017). mixOmics: An R package for 'omics feature selection and multiple data integration. PLoS Comput. Biol. *13*, e1005752.https://doi.org/10.1371/journal.pcbi.1005752.<br>Tutorial: https://mixomics.org/mixdiablo/ | RRID:SCR_016889 |
| multiGSEA | Canzler, S., & Hackermüller, J. (2020). multiGSEA: a GSEA-based pathway enrichment analysis for multi-omics data. *BMC bioinformatics*, *21*(1), 561. | N/A |
| MOFA | Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J.C., Buettner, F., Huber, W., and Stegle, O. (2018). Multi-Omics Factor Analysis-a framework for unsupervised integration of multi-omics data sets. Mol. Syst. Biol. *14*, e8124.https://doi.org/10.15252/msb.20178124.<br>Tutorial: https://biofam.github.io/MOFA2/ | RRID:SCR_022992 |

## STEP-BY-STEP METHOD DETAILS

### Data preprocessing and normalization

Timing: 2–4 h

Preprocess and normalize each omics layer to ensure data compatibility for integration. This step standardizes data distributions and removes technical variation.

1. Load omics datasets into R.
   a. Load required libraries.

```
library(tidyverse) library(MOFA2) library(mixOmics) library(edgeR)
library(limma)
```

b. Read omics datasets (gene expression matrix, protein abundance matrix, metabolite abundance matrix).

2. Filter low-abundant features or features with many missing values from each omics layer.
   a. Filter transcripts by expression to remove genes with <0.2 CPM in more than 50% of the samples.
   b. Remove proteins with >50% missing values across samples.
   c. Remove metabolites with >50% missing values across samples.
3. Apply log2 transformation to all datasets.
   a. Transform the transcriptomics data: log2(CPM + 1).
   b. Transform the proteomics data: log2(LFQ intensity).
   c. Transform the metabolomics data: log2(peak intensity).
4. Perform z-score normalization.
   a. Calculate mean and standard deviation for each feature across all samples.
   b. Standardize each feature to mean=0 and SD=1:

```
Z_score = (value – mean) / SD
```

   c. Apply z-score transformation to all omics datasets.

*Note: Z-score normalization ensures all omics layers are on comparable scales for integration.*

*Note:* The decision to remove measurements based on a 50% missing value threshold is only a guideline. Filtering should consider the missingness pattern, such as whether data are missing at random or not. For example, if missing values are concentrated in specific biological groups, they may carry meaningful information and should not be removed. This consideration is particularly important for smaller datasets.

### Multi-omics factor analysis

Timing: 1–3 h

Perform unsupervised multi-omics integration using MOFA to identify latent factors that capture coordinated variation across omics layers.

5. Create a MOFA object.
   a. Prepare data in MOFA format as a list of matrices:

```
data_list <- list("Transcriptomics" = t(transcriptomics_data), "Proteomics" = t(proteomics_data), "Metabolomics" = t(metabolomics_data)
```

   b. Create MOFA object:

```
MOFAobject <- create_mofa(data_list)
```

6. Configure MOFA training parameters.
   a. Set data options:

```
data_opts <- get_default_data_options(MOFAobject)
```

   b. Set model options, start with 5 factors:

```
model_opts <- get_default_model_options(MOFAobject)
model_opts$num_factors <- 5
```

   c. Set training options:

```
train_opts <- get_default_training_options(MOFAobject)
train_opts$convergence_mode <- "fast" train_opts$seed <- 42
```

   d. Prepare the model:

```
MOFAobject <- prepare_mofa(object = MOFAobject, data_options = data_opts, model_options = model_opts, training_options = train_opts)
```

7. Train the MOFA model.
   a. Run model training:
      `MOFAobject \<- run_mofa(MOFAobject)`
   b. Wait for model convergence.

   *Note:* The model automatically prunes non-informative factors during training.

8. Evaluate MOFA model.
   a. Calculate variance explained by each factor:
      `r2 <- calculate_variance_explained(MOFAobject)`
   b. Plot variance explained:
      `plot_variance_explained(MOFAobject, plot_total = TRUE)`
   c. Visualize factor values for each factor:
      `plot_factors(MOFAobject, factors = 1:5)`
   d. Identify active factors explaining relevant amounts of variance in at least one omics layer (this is context specific).
9. Extract feature weights from active factors.
   a. Get feature weights from active factors.
      `weights <- get_weights(MOFAobject, views = "all", factors = "all", as.data.frame = TRUE)`
   b. Identify top contributing features (absolute weight > 0.9 or in the top 10).
   c. Generate weight distribution plots, e.g.,:
      `plot_weights(MOFAobject, view = "Transcriptomics", factor = 1, nfeatures = 10)`
10. Perform pathway enrichment analysis of MOFA factors.
   a. Extract top genes from each factor based on absolute weight values.
   b. Run gene set enrichment analysis using GSEApy in Python or multiGSEA in R.
   c. Test against KEGG and Reactome pathway databases.
   d. Apply Benjamini-Hochberg FDR correction (threshold < 0.05).
   e. Visualize enriched pathways for each factor using bar plots or dot plots.

### PLS-based integration using DIABLO

Timing: 2–4 h

Perform supervised multi-omics integration using DIABLO (mixOmics package) to identify features that discriminate between experimental conditions.

11. Prepare data for DIABLO analysis.
   a. Ensure sample groups/conditions are defined (e.g., time points, treatments).
   b. Create a list of data matrices (samples as rows, features as columns):
      `X <- list(Transcriptomics = transcriptomics_data, Proteomics = proteomics_data, Metabolomics = metabolomics_data)`
   c. Define outcome variable Y (sample groups).
12. Set up the DIABLO design matrix.
   a. Create design matrix specifying correlations between omics blocks.
      ```
      design <- matrix(0.1, ncol = 3, nrow = 3, dimnames = list(names(X), names(X)))
      diag(design) <- 0
      ```

   *Note:* Design matrix values of 0.1 indicate weak correlations whereas values close to 1 would indicate strong correlations.

13. Tune the number of components and features.
    a. Perform initial model training to determine optimal parameters:
    ```
    tune.diablo <- tune.block.splsda(X = X, Y = Y, ncomp = 5, design = design,
    validation = 'Mfold', folds = 3, nrepeat = 10)
    ```
    b. Select number of components based on lowest error rate and how much variance has been explained (typically 2–3 components).
    c. Choose number of features per omics layer (typically 10–50 features per component).
14. Build the final DIABLO model.
    a. Construct sparse PLS-DA model with selected parameters:
    ```
    diablo.model <- block.splsda(X = X, Y = Y, ncomp = 2, keepX =
    list(Transcriptomics = 10, Proteomics = 10, Metabolomics = 10), design =
    design)
    ```
    b. Assess model performance using classification error rates.
15. Visualize DIABLO results.
    a. Create sample plots showing separation of conditions:
    ```
    plotIndiv(diablo.model, ind.names = FALSE, legend = TRUE)
    ```
    b. Generate loadings plots showing feature contributions:
    ```
    plotLoadings(diablo.model, comp = 1, contrib = 'max')
    ```
    c. Create circos plots to visualize correlations between features across omics layers:
    ```
    circosPlot(diablo.model, cutoff = 0.7)
    ```
    d. Extract variable weights using the plotVar() function.
16. Perform pathway enrichment on DIABLO components.
    a. Extract selected features for each component and omics layer.
    b. Run gene set enrichment analysis using GSEApy in Python or multiGSEA in R.
    c. Test against KEGG and Reactome pathway databases.
    d. Apply Benjamini-Hochberg FDR correction (threshold < 0.05).
    e. Visualize enriched pathways for each factor using bar plots or dot plots.

## EXPECTED OUTCOMES

Multi-omics integration approaches provide a systematic framework to evaluate how different omics layers are perturbed together, revealing correlations and interconnected patterns that may be missed when layers are analyzed individually. This integrated view can uncover coordinated molecular responses across layers and provide insights into complex system-level interactions.

The expected outcomes differ between MOFA (unsupervised) and DIABLO (supervised) approaches, but both provide complementary perspectives on the data. An example dataset can be accessed at https://github.com/mmattano/Multi_example_data.

### MOFA integration outcomes

MOFA typically identifies a small number of active factors that collectively explain 30%–50% of total variance across all omics layers. Each factor represents a distinct subset of biochemical elements with contributions across omics layers. Variance explained plots reveal which omics layers contribute most to each factor. Weight distribution plots typically show linear patterns at extremes with many features having weights >0.9, indicating broad coordinated changes rather than sparse signals.

### DIABLO integration outcomes

DIABLO produces clear separation of experimental conditions in low-dimensional space. Loadings plots and contribution plots reveal which features drive component separation, with colors indicating time point associations and signs showing dimension orientation. Circos plots illustrate strong correlations between features across omics layers, revealing trans-omics relationships not apparent in single-layer analyses.

## LIMITATIONS

This protocol has several important limitations that should be considered when designing experiments and interpreting results. The protocol requires matched samples across all omics layers. Missing one or more omics measurements for specific samples reduces statistical power and may prevent identification of weak but biologically relevant factors. Ideally, obtain complete omics profiles for all biological replicates.

No minimal sample size can be universally recommended for this protocol. The required sample size is highly study-dependent and influenced by effect size, biological heterogeneity, technical variation, number of omics layers, feature dimensionality, missingness, and the number of experimental groups. MOFA can be applied for exploratory analysis with relatively small sample sizes, but the stability and reproducibility of inferred latent factors should be carefully evaluated. DIABLO typically requires a larger number of samples per group to ensure robust model training and reliable cross-validation performance. Robustness can be evaluated through repeated cross-validation and assessment of factor or feature stability. Simulation- or power-based approaches can be used when planning new experiments to ensure adequate sample size and study design.

Sample size affects model reliability. Smaller sample sizes may lead to overfitting factors that do not generalize independent datasets. Missing data handling differs between methods. DIABLO requires complete data matrices or imputation, which can introduce bias if missingness is not random. Evaluate missing data patterns before choosing the integration method.

The protocol identifies correlation and association patterns but cannot establish causality. Strong correlations between features across omics layers may reflect common regulatory mechanisms but do not prove direct regulatory relationships. Experimental validation is required to confirm causal relationships identified through integration.

Pathway enrichment analysis depends on knowledge base completeness. Less-studied organisms or biological processes may show few significant enrichments due to limited pathway annotations rather than lack of biological signal. Metabolite identification and pathway mapping remain incomplete for many metabolomics features.

The protocol focuses on linear relationships between features. Non-linear regulatory relationships or complex interaction effects may not be captured by correlation-based network construction or factor analysis. Consider complementary approaches for non-linear pattern detection if linear methods yield limited insights. Temporal or dynamic processes require appropriate experimental design.

The protocol works with time-series or multi-condition data but does not explicitly model temporal dynamics. For studying dynamic processes, consider time-series specific methods or differential equation-based modeling approaches.

Limited inclusion of categorical data. Phenotype or clinical data can be incorporated in multi-omics integration if they are numerical. MOFA and DIABLO allow numerical phenotype or clinical data to be analyzed as a separate block alongside omics layers. Categorical variables (nominal or ordinal) cannot be directly included, because the distances between categories are not comparable and numeric transformations may be misleading. For categorical data, alternative approaches such as partial correlation or statistical association tests can be applied to evaluate associations with omics features.

## TROUBLESHOOTING

### Problem 1

MOFA model does not converge, related to step 7.

### Potential solution

MOFA convergence issues typically arise from overly complex models or poor data quality.

First, reduce the initial number of factors. The model will prune uninformative factors automatically, but starting with too many can prevent convergence.

Second, increase the maximum number of iterations in training options to 5000 or 10000. Some datasets require more iterations to reach stable ELBO values.

Third, apply more stringent feature filtering by retaining only the top 2000–5000 most variable features per omics layer. This reduces noise and computational burden.

Fourth, verify that z-score normalization was applied correctly to all datasets. Extreme outliers or incorrectly scaled data can prevent convergence.

Finally, check for factors with very low variance explained (<5%) after a few hundred iterations - these indicate the model is trying to fit noise. Use the elbo_opts parameter to set convergence tolerance lower (e.g., 0.001 instead of 0.01) if near-convergence is occurring but not reaching the threshold.

### Problem 2

DIABLO model shows poor classification performance, related to steps 14 and 15.

### Potential solution

Poor DIABLO classification indicates either too high biological similarity between conditions or suboptimal model parameters.

First, increase the number of features selected per component (keepX parameter) from 10 to 20–50. More features may be needed to capture discriminative patterns.

Second, adjust the design matrix correlation values. Try stronger correlations (0.5 or 0.7) between omics blocks if the biological signal is expected to be consistent across layers, or weaker correlations (0.05) if layers show independent patterns.

Third, increase the number of components from 2 to 3–4 to capture more complex patterns. Run tuning with ncomp = 5 to systematically evaluate optimal component number.

Fourth, verify that sufficient biological replicates exist per condition (minimum 3, ideally 5+). With fewer replicates, the model may overfit training data.

Fifth, check for batch effects or technical variation that may obscure biological differences. Apply ComBat or similar batch correction before DIABLO if samples were processed in multiple batches.

Finally, consider whether the biological conditions actually show strong molecular differences. Not all phenotypic differences manifest as clear omics signatures, particularly in complex or heterogeneous systems.

### Problem 3

No significant pathway enrichments identified, steps 10 and 16.

### Potential solution

Lack of enrichment may indicate diffuse biological signals or mismatched pathway databases.

First, expand the feature list used for enrichment. Instead of using only top 1% by weight, include top 500 or 1000 features per factor. Enrichment requires sufficient overlap with pathway gene sets.

Second, test multiple pathway databases. Try adding, for example, GO Biological Processes or WikiPathways for broader coverage. KEGG works well for metabolism but poorly for signaling pathways. For metabolomics, KEGG metabolic pathways and HMDB metabolite classes provide complementary annotations.

Third, relax the FDR threshold from 0.05 to 0.1 for exploratory analysis. Many potentially true enrichments fall just below significance with small sample sizes.

Fourth, verify that correct gene identifiers are used. Gene symbols work for most databases, but some require Entrez IDs or Ensembl IDs. Mismatched identifiers cause failed enrichment.

Fifth, consider organism-specific databases. Human pathway databases are most complete; for other organisms, use organism-specific resources or ortholog mapping.

Finally, examine factor weight distributions - if weights are uniformly distributed rather than showing clear positive and negative extremes, the factor may not represent a coherent biological process and enrichment is less meaningful. Focus enrichment on factors with bimodal weight distributions indicating clear positive and negative regulation.

### Problem 4

High missing data rates prevent DIABLO analysis, related to steps 13 and 14.

### Potential solution

DIABLO requires complete data matrices unlike MOFA which handles missing values.

First, apply feature-wise filtering before sample-wise filtering. Remove features with > 50% missing values to reduce overall missingness.

Second, alternatively use k-nearest neighbors (KNN) imputation for remaining missing values. KNN imputation preserves local data structure better than simple mean imputation. The impute package in R or scikit-learn in Python provides robust implementations.

Third, evaluate imputation quality using cross-validation. Mask known values, impute, and calculate imputation errors. High errors indicate unreliable imputation and suggest using MOFA instead. Fourth, consider analyzing omics layers with acceptable missing rates (<20%) and excluding problematic layers from DIABLO. Perform separate integration on compatible subsets.

Fifth, if missing data is not random (e.g., metabolites below detection limit), use appropriate censored data methods rather than standard imputation. Left-censored missing values (below detection) can be replaced with small values (half minimum detected value).

Finally, if missing rates exceed 30% across multiple layers, use MOFA exclusively rather than forcing DIABLO analysis.

## RESOURCE AVAILABILITY

### Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Thomas Moritz (thomas.moritz@sund.ku.dk).

## Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Matthias Anagho-Mattanovich (matthias.mattanovich@sund.ku.dk).

## Materials availability

This protocol does not generate new unique materials.

## Data and code availability

This protocol describes the analysis of publicly available multi-omics datasets. All software packages used are freely available through CRAN, Bioconductor, or PyPI. An example dataset can be found at https://github.com/mmattano/Multi_example_data.

## ACKNOWLEDGMENTS

M.A.-M. and H.A.A.-M. were funded by the Novo Nordisk Foundation PhD fellowship (NNF19CC0035454 and NNF19SA0035440). Novo Nordisk Foundation (grant nos. NNF18CC0034900 and NNF23SA0084103) provided unconditional donation to Novo Nordisk Foundation Center for Basic Metabolic Research. Some of the figures were generated using BioRender.

## AUTHOR CONTRIBUTIONS

M.A.-M. conceptualized and wrote this manuscript. All authors contributed to setting up this protocol and reviewed the manuscript.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

## REFERENCES

1. Anagho-Mattanovich, M., Anagho-Mattanovich, H.A., Argemi-Muntadas, L., Nielsen, M.L., Gao, Q., and Moritz, T. (2025). Multi-omics analysis of thermogenic lipolysis in brown adipocytes. iScience *28*, 113382. https://doi.org/10.1016/J.ISCI.2025.113382.

2. Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J.C., Buettner, F., Huber, W., and Stegle, O. (2018). Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. Mol. Syst. Biol. *14*, 8124. https://doi.org/10.15252/msb.20178124.

3. Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J.C., and Stegle, O. (2020). MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. *21*, 111. https://doi.org/10.1186/S13059-020-02015-1/FIGURES/4.

4. Rohart, F., Gautier, B., Singh, A., and Lê Cao, K.A. (2017). mixOmics: An R package for 'omics feature selection and multiple data integration. PLoS Comput. Biol. *13*, e1005752. https://doi.org/10.1371/JOURNAL.PCBI.1005752.

5. Singh, A., Shannon, C.P., Gautier, B., Rohart, F., Vacher, M., Tebbutt, S.J., and Lê Cao, K.A. (2019). DIABLO: an integrative approach for identifying key molecular drivers from multi-omics assays. Bioinformatics *35*, 3055–3062. https://doi.org/10.1093/BIOINFORMATICS/BTY1054.