

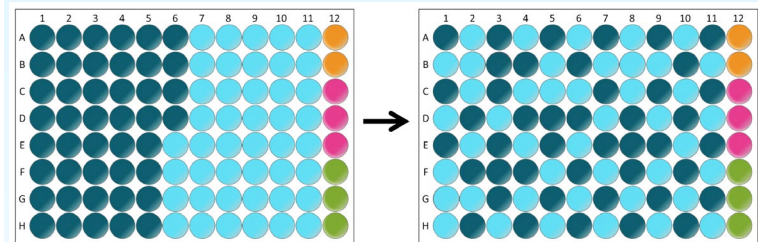


0

Before generating data

Facilitate plate randomization and bridge sample selection

`olink_plate_randomizer(manifest)` generates a scheme for how to plate samples with an option to keep subjects on the same plate for longitudinal randomization using `SubjectColumn` argument



Visualize plate randomization:

- `olink_displayPlateDistributions(data, fill.color)`
- `olink_displayPlateLayout(data, fill.color)`

Note about Randomization:

- When a subject has more than one sample, these should be kept together on the same plate
- Samples should still be randomized within a plate
- Groups of interest should be randomized across plates as evenly as possible

`olink_bridgeselector(df, sampleMissingFreq, n)` selects `n` number of bridge samples based on input data. The samples are chosen based on passing QC, covering a large range of the data, and having less than `sampleMissingFreq` data below LOD per sample.



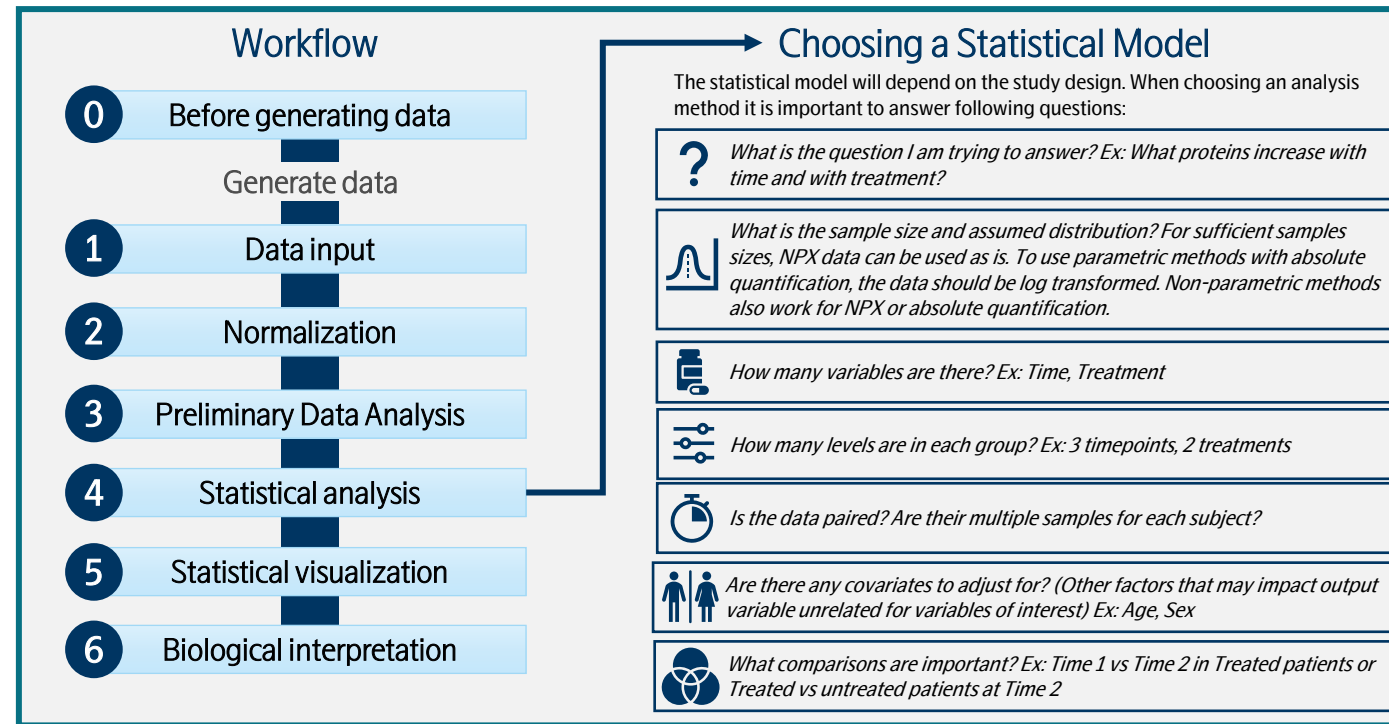
Contact us:

biostat@olink.com statistical services and general stats questions

biostattools@olink.com Olink Analyze and Shiny app support

support@olink.com Olink lab product and technical support

info@olink.com for more information



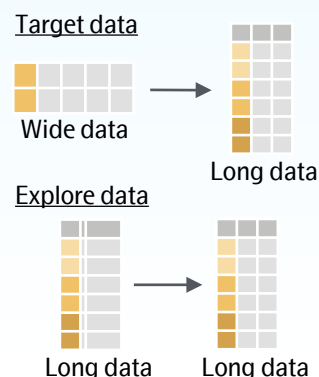
1

Data input

Reading in Olink data

`read_NPX(filename)` imports an NPX[™] or Quant file exported from Olink NPX Signature, Olink NPX Explore, or MyData. The file should not be altered prior to import and will be imported in long format

Column name	Description
SampleID	Sample ID name
Index	Unique number for each SampleID
OlinkID	Unique OlinkID for each protein
UniProt	UniProt ID
Assay	Protein symbol
MissingFreq	Proportion of samples below LOD
Panel_Version	Panel version
PlateID	Plate ID
QC_Warning	QC status
LOD	Limit of detection
NPX	Normalized Protein Expression
Normalization	Normalization method used



2

Normalization

Since NPX is a relative measurement, normalization is necessary to combine data across plates or across projects. An adjustment factor is calculated for each assay (protein) which is added to the NPX values from the second plate or project.

Normalizing 2 projects
`olink_normalization_bridge()` - bridges 2 projects using overlapping samples using the **median of paired differences**
`olink_normalization_subset()` - normalizes 2 projects using the **difference of group medians**. The group can be a subset or all samples. No overlapping samples required.
`olink_normalization()` - can be used for bridging, subset, or reference median normalization.

Normalizing more than 2 projects
`olink_normalization_n()` - normalize multiple projects based on provided normalization schema

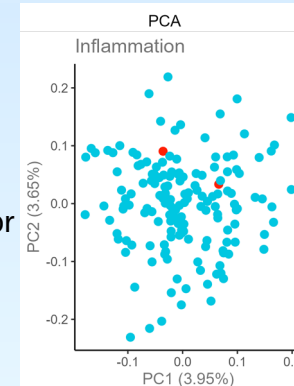
For more on bridging and normalization, check out our white paper and vignette.

3

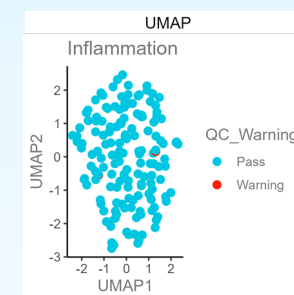
Preliminary Data Analysis and Quality Control

Data overview, outlier detection, and identifying global trends

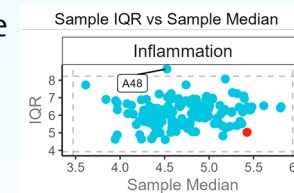
`olink_pca_plot(df)` generates a principal component analysis (PCA) projection of all samples from NPX data along two principal components. Color can be set by `color_g` argument.



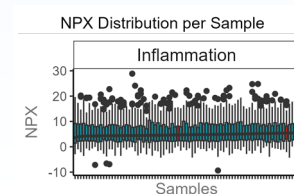
`olink_umap_plot(df)` generates a uniform manifold approximation and projection (UMAP) for dimension reduction of all samples from NPX data along 2 specified components. Color can be set by `color_g` argument.



`olink_qc_plot(df)` generates a plot of interquartile range (IQR) vs median for all samples for each panel. Outliers can be tagged using `IQR_outlierDef` and `median_outlierDef`. Color can be set using `color_g` argument.



`olink_dist_plot(df)` generates boxplots of NPX vs SampleID. Color can be set by `color_g` argument.



4

Statistical analysis

Number of variables	Number of levels per variable	Support for paired data?	Support for covariates?	Parametric Statistics <i>Statistical analysis of normally distributed data</i>	Non-parametric Statistics <i>Statistical analysis of non-normally distributed data</i>
1	2	☑	☒	<p>olink_ttest(df, variable, pair_id) performs a t-test for a given grouping variable as defined by variable argument. Paired t-test can be performed by defining pair_id argument. Variable must have 2 groups with more than 2 samples per group.</p>	<p>olink_wilcox(df, variable, pair_id) performs Mann-Whitney U Test per protein for a given grouping variable as defined by variable argument. Wilcoxon signed-rank test can be performed by defining pair_id argument. Variable must have 2 groups with more than 2 samples per group.</p>
1	2+	☒	☒	<p>olink_anova(df, variable) performs an ANOVA F-test for each protein based on variables and covariates defined by variable and covariate arguments or by formula defined by model_formula argument.</p> <p>olink_anova_posthoc(df, variable, olinkid_list, effect) performs an ANOVA posthoc test per protein listed in olinkid_list based on variables and covariates defined by variable and covariate arguments or by formula defined by model_formula argument.</p>	<p>olink_one_non_parametric(df, variable, dependence) performs a one-way Kruskal-Wallis Test for each protein. When dependence = TRUE, a one-way Friedman Test will be performed for each protein. Model is defined by variable argument.</p> <p>olink_one_non_parametric_posthoc(df, olinkid_list, variable) performs a Wilcox posthoc test per protein listed in olinkid_list based on the variable defined by the variable argument.</p>
1+	2+	☒	☑	<p>olink_ordinalRegression(df, variable, covariates) performs a 2-way ordinal ANOVA for each protein based on variables and covariates defined by variable and covariate arguments.</p> <p>olink_ordinalRegression_posthoc(df, olinkid_list, variable, covariates, effect) performs a 2-way ordinal ANOVA posthoc test per protein listed in olinkid_list based on variables and covariates defined by variable and covariate arguments.</p>	<p>olink_ordinalRegression(df, variable, covariates) performs a 2-way ordinal ANOVA for each protein based on variables and covariates defined by variable and covariate arguments.</p> <p>olink_ordinalRegression_posthoc(df, olinkid_list, variable, covariates, effect) performs a 2-way ordinal ANOVA posthoc test per protein listed in olinkid_list based on variables and covariates defined by variable and covariate arguments.</p>
1+	2+	☑	☑	<p>olink_lmer(df, variable) performs a linear mixed model per protein based on variables and covariates defined by variable and covariate arguments or by formula defined by model_formula argument</p> <p>olink_lmer_posthoc(df, variable, olinkid_list, effect, random) performs a linear mixed model posthoc per protein listed in olinkid_list based on variables and covariates defined by variable and covariate arguments or by formula defined by model_formula argument.</p>	<p>olink_lmer(df, variable) performs a linear mixed model per protein based on variables and covariates defined by variable and covariate arguments or by formula defined by model_formula argument</p> <p>olink_lmer_posthoc(df, variable, olinkid_list, effect, random) performs a linear mixed model posthoc per protein listed in olinkid_list based on variables and covariates defined by variable and covariate arguments or by formula defined by model_formula argument.</p>

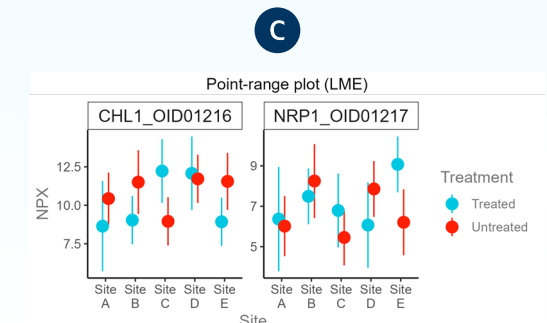
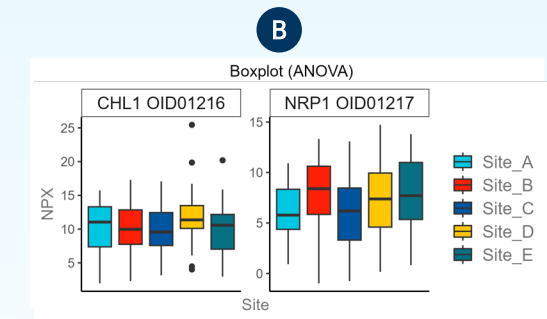
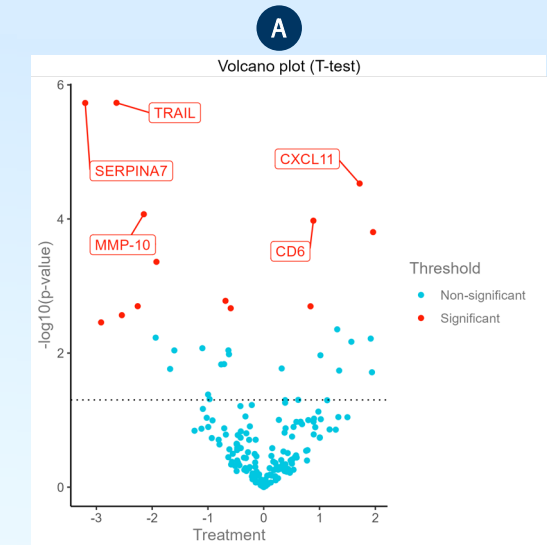
5

Statistical visualization

olink_volcano_plot(p.val_tbl) generates a volcano plot using the results from **olink_ttest()**

olink_boxplot(df, variable, olinkid_list, posthoc_results, ttest_results) generates boxplots of **variable** for each protein in **olinkid_list** for **posthoc_results** or **ttest_results**

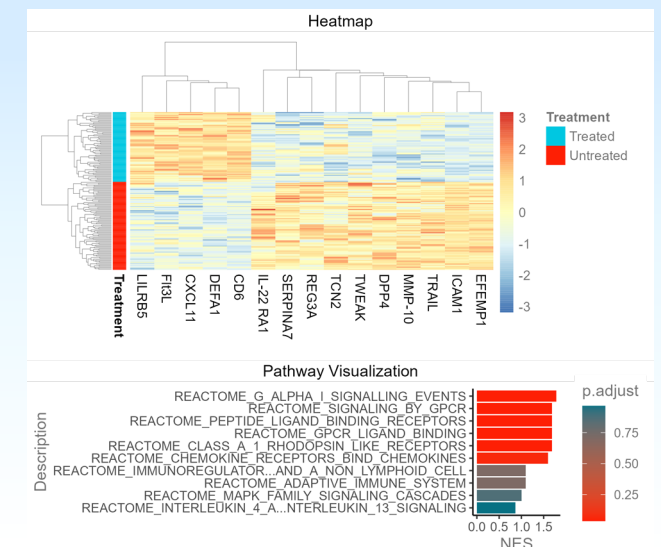
olink_lmer_plot(df, variable, random, olinkid_list, covariates, x_axis_variable, col_variable) generates point-range plots of using **x_axis_variables** (**x**-axis) and **col_variable** (**color**) for each protein in **olinkid_list**.



6

Biological interpretation

Pathway Enrichment and Heatmap



olink_pathway_enrichment(data, test_results) performs pathway enrichment using over-representation analysis or gene-set enrichment analysis (based on **method** argument)

olink_pathway_heatmap(enrich_results, test_results) generates a heatmap of proteins related to pathways using enrichment results from **olink_pathway_enrichment()**

olink_pathway_visualization(enrich_results) generates bargraph of top/selected enrichment terms based on enrichment results from **olink_pathway_enrichment()**

olink_heatmap_plot(df) generates a heatmap of all samples from NPX data