ReadMe

PloiDEx - Ploidy-Dependent Expression. Supplementary plotting application.

The supplementary plotting software is a suave.io-based, standalone console application, which utilizes **.NET Framework 4.6.1.**

The pre-compiled application can be started with the **PloiDEx.exe** in the Release folder. It will by default open http://localhost:8083/YeastPloidy/ in the systems default browser. Testing has been performed in Windows 10 (v1909) with Mozilla Firefox (ESR 78.5.0) in a 1080p display setting.

The application is written in F# (4.5) and built in Microsoft Visual Studio 2019 (16.8.2).

The following scripts are included in the compilation and open to freely modify and adapt:

Types.fs contains the type declarations.

Fun.fsx contains the filtering and plotting functions.

Suave.fsx contains the application structure.

Used libraries and packages:

FSharp.Core.4.5.2	https://www.nuget.org/packages/FSharp.Core/4.5.2
FSharp.Data.2.4.6	https://www.nuget.org/packages/FSharp.Data/2.4.6
Newtonsoft.Json.11.0.2	https://www.nuget.org/packages/newtonsoft.json/11.0.2
Suave.2.4.3	https://www.nuget.org/packages/Suave/2.4.3
Jquery.js (cdn_latest)	https://jquery.com/download/
Plotly.js (cdn_latest)	https://plotly.com/javascript/
Bootstrap.min.js (3.3)	https://getbootstrap.com/docs/3.3/

All packages are open-source and use the MIT License with the exception of suave.io and FSharp.Data, which use the apache 2.0 license. Special thanks to the respective and all contributing authors.

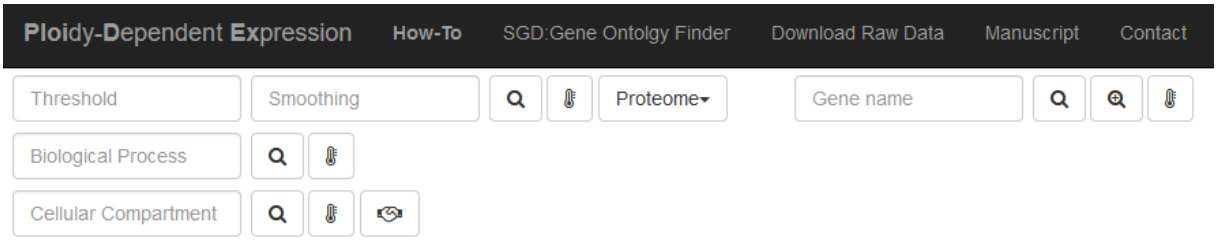
Contact: See related manuscript.

Functionality:

The application allows a straightforward, explorative visualization of the proteomics and transcriptomics dataset presented in the related manuscript. To do so, it follows a simple, browser-based structure and listens to requests for as long as the console application is running in the background. Closing it will terminate the process.

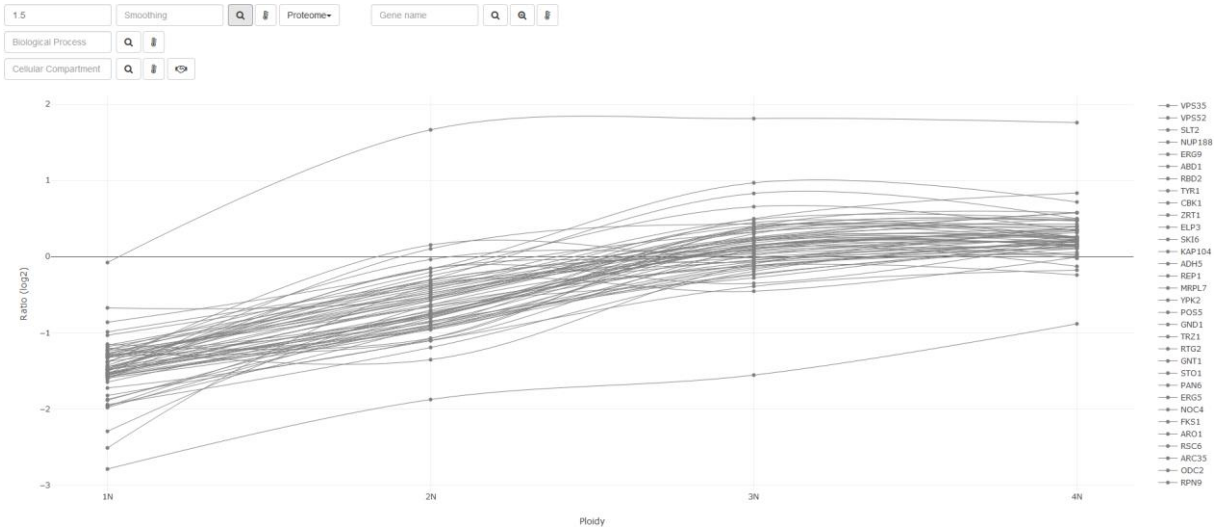
All plots are visualized by the plotting library **plotly.js**, which comes with a set of interactive built-in functions. Such as showing the label of the trace by hovering over it, zooming or re-scaling and exporting the plot as .png.

How-To opens a module explaining each individual function in more detail.

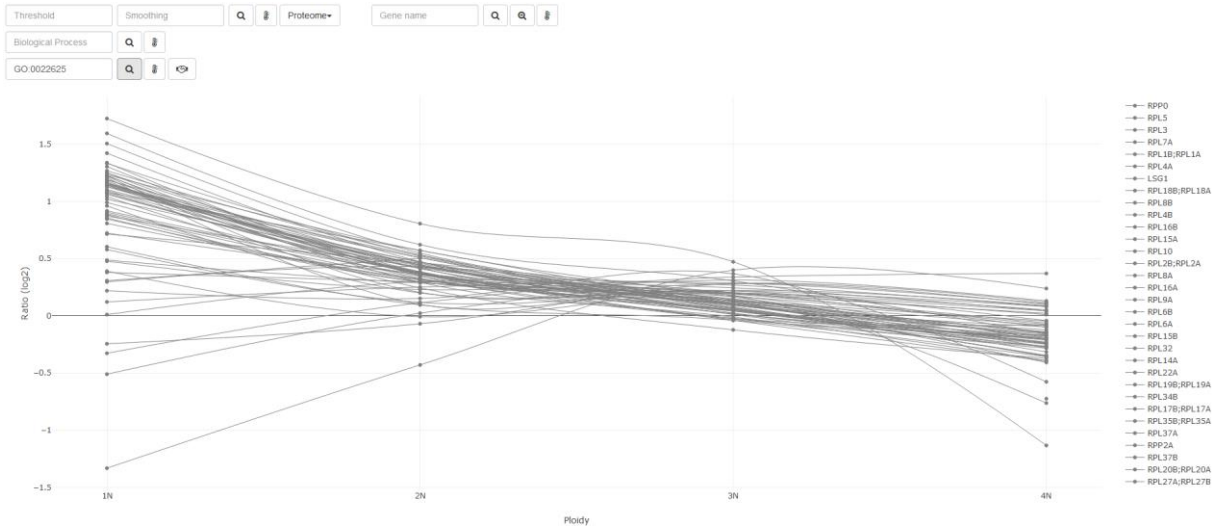


Profile plots can be plotted by specifying log2 FC cutoffs as filtering **threshold** for either the proteome or transcriptome dataset, normalized to the SILAC standard or s. pombe spike. The dataset can be chosen in the dropdown menu. For further information about the data see Materials and Methods.

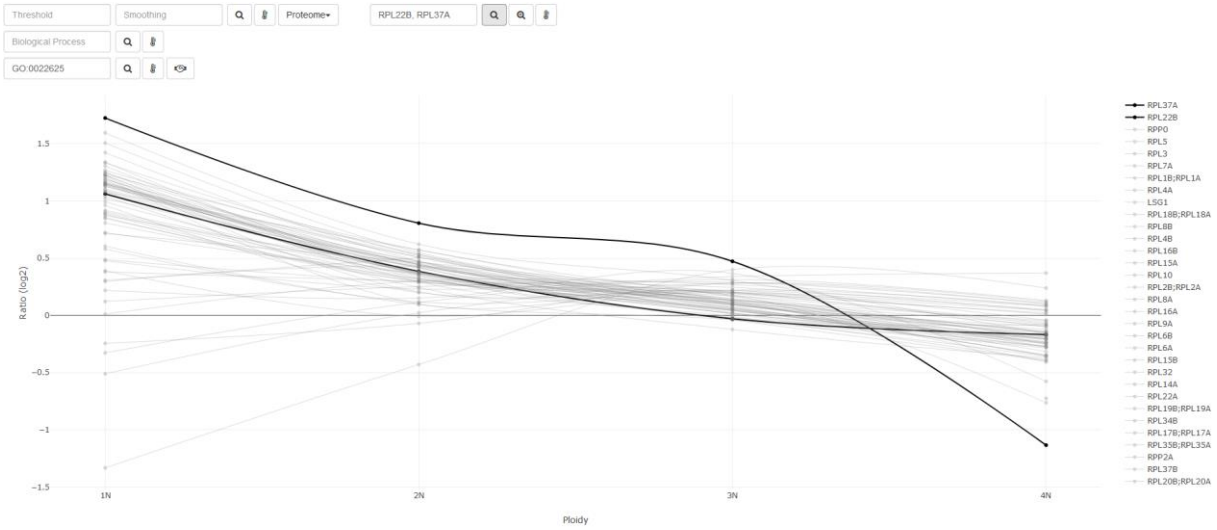
The program will calculate the total difference between ploidies and filter for bigger or equals positive, or smaller or equals negative thresholds. The **smoothing** function optionally allows the user to add a successive filtering step, which calculates the individual differences between ploidies and filters out spikes that cross the absolute, specified smoothing cutoff.



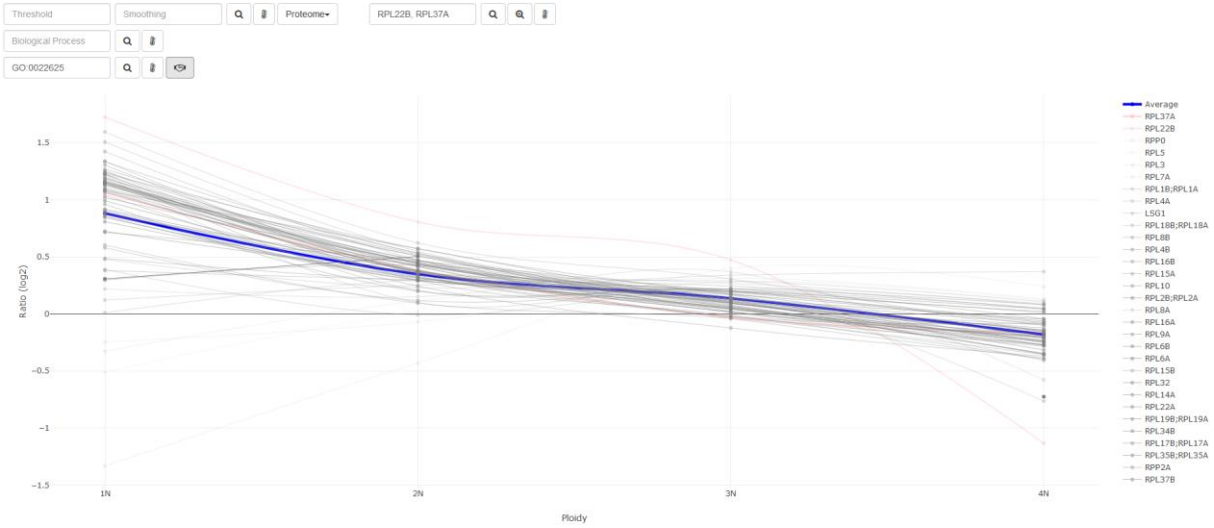
Alternatively, a profile plot can be initialized by searching for a GO **biological process** or GO **cellular compartment** Term ID, such as GO:0022625 for cytosolic large ribosomal subunit. For ease of use the saccharomyces genome database has been linked in the navbar, to search and copy the Term IDs:



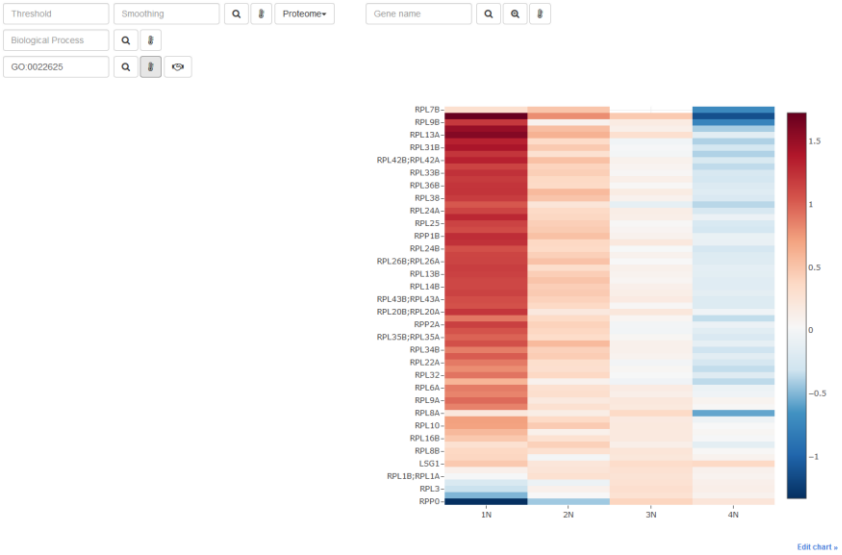
Individual genes can be initialized by the **gene name** function, which will be plotted in front of the current GO Term if specified, which in turn optionally gets a lower opacity. The gene names can be added individually or in form of a comma or semicolon separated list. The application will try to autocomplete, e.g. “mcm” would plot mcm2-7:



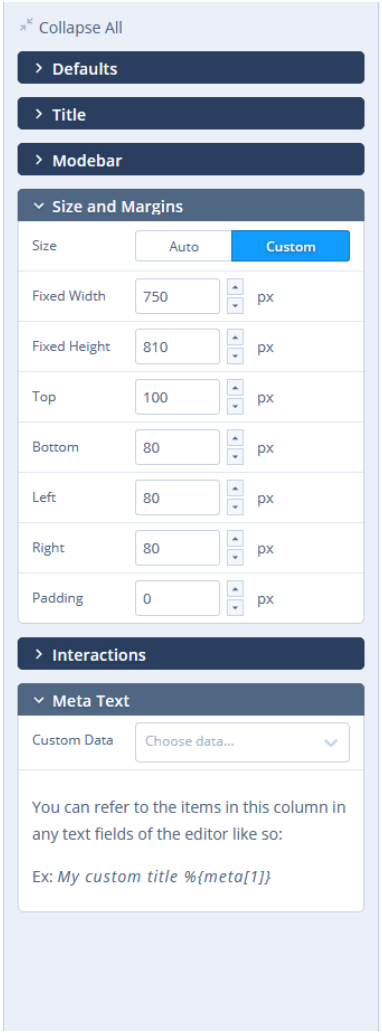
The application can calculate the **average** of the given GO term and add it as a separate blue trace. Specified gene names will be plotted in red. All traces have a reduced opacity based on their distance to the average.



The application can plot the specified input as heatmap, sorted by difference. The color, width and height of the heatmap are set manually. The application can be freely modified in the underlying Plot.js, which can be found in Release\public\Plot.



Alternatively, the “edit chart>>” button in the bottom right corner transfers the given plot and underlying data to the **Plotly chart studio**, to either export the data or further customize the plot:



	x	y	z[0]	z[1]	z[2]	z[3]
1	1N	RPP0	-1.33232831954956	-0.429654747247696	0.398629665374756	0.23880036175251
2	2N	RPL5	-0.510514736175537	0.0228984728455544	0.281592100858688	0.0990520268678665
3	3N	RPL3	-0.327569723129272	0.113289780914783	0.312312304973602	0.130052074790001
4	4N	RPL7A	-0.244571104645729	-0.0696856901049614	0.281059116125107	0.118567675352097
5		RPL1B;RPL1A	0.01073978841304...	0.298663467168808	0.268057942390442	0.0808972492814064