RCC Data Evaluation and Analysis App

Purpose:

This is a demo project in progress to get more familiar with MVP pattern for UI development as my previous tools were more like prototypes (functional for multiple teams to use but still prototypes and not ideal foreasy re-use of code). The ostensible purpose of the tool is evaluation and automated analysis of the output of all nCounter platforms, including gene expression (single and cross-panel/cross-RLF), miRNA, CNV, PlexSet (sample multiplexed), or nCounter readout from GeoMx protein experiments.

The first goal is quick data ingestion and data eval with no persistence as this functionality alone is important and useful to core labs and those running the instruments and the small size of the files makes re-loading trivial. Normalization and analysis (DE and dimension reduction both ontology-driven and data-driven) with auditable intermediates will come later, with the purpose there (aside from use to any hypothetical users) being to demonstrate algorithm programming, possibly using unmanaged code for speed.

Functionality:

The tool can load the gene count output of nCounter platforms, RCCs, as well as the RLF and/or PKC files that link nCounter molecular barcodes to their targets. Files can either be loaded directly or from zips. For zips, the tool will recursively extract all files/directories and pull out any of the three file types found. RLF definition as an object in the project is flexible enough to only require loading of RLFs if necessary (Gene target and accession info is included in the RCC but RLFs include additional columns of info, including ProbeID which is a primary key necessary for functionality like cross-RLF experiments).

The main form is where data loading and initiating any analyses will start. Barebones at the moment with any later added controls to provide redundant function for funneling the user into the workflow. After RCCs are loaded a table will display user selected columns for RCC properties (defaults will be available with option to select from RCC properties by context menu displayed when right clicking column headers). Left click of column headers will sort RCCs by the data in that column and holding down control while clicking will allow secondary and tertiary sorting. The table can be saved as a CSV via right click context menu option.

Raw counts and RCC header info can be collected in a raw count table for QC review by clicking an option in the Review menu. This will be the menu to go to for all QC evaluation functionality. A "plate view" format will be available shortly for multiplex experiments to show QC info in context of how samples were hybridized, for troubleshooting purposes. Dynamic range, linearity/correlation scatterplots, correlation heatmap, and PCA analyses will be available shortly to aid in evaluating flagged data for impacts on dynamic range and linearity. Classes for this already exist from previous projects but dialogs need to be converted to MVP and some refactoring for maintainability needs to be done.

A main focus for this project is on streamlined workflow and inherent flexibility, though further flexibility is continuously being added through dialogs available from the preferences menu.

Long run the tool will include GeNorm for annotated HK evaluation and an implementation of RefSeq, particularly for miRNA data which, particularly for biofluid data, often presents normalization challenges. In addition a robust though streamlined differential expression analysis (neg binomial multiple regression model) will be included, and I'm looking at different dimension reduction methods to include. Ontology is generally already curated for Nanostring panels, given the biology-specific bias in the content, so no ontology enrichment will be included.

Design:

This is a WinForms app (i.e. early twenty teens) using .NET framework as I had a number of classes already built for the eval and data analysis functions in framework 4.6.2.

All windows will include at least a View and presenter and most will have a model. The models use business objects found in the NCounterCore directory which will eventually be its own DLL once stabilized. Data transfer is handled either directly by the presenter or via custom EventArgs. Communication between presenters is via TinyMessenger through custom generic messages.