Integrative analysis intro

Mikhail Dozmorov

Spring 2018
Integrative analysis

Three main approaches

- **Concatenation-based integration** - combining data sets from different data types at the raw or processed data level before modelling and analysis.

- **Transformation-based integration** - performing mapping or data transformation of the underlying data sets before analysis, and the modelling approach is applied at the level of transformed matrices.

- **Model-based integration** - performing analysis on each data type independently, followed by integration of the resultant models to generate knowledge about the trait of interest.

Clustering for data integration

- **iCluster** - uses a Gaussian latent variable model to infer clusters
  - It assumes that there is a common set of latent cluster membership variables across all datasets
  - Differences in structure between different datasets are accounted for only via individual noise terms, which correspond to within-dataset variances
  - It uses the k-means algorithm to extract the actual cluster assignments given latent variable values


https://www.mskcc.org/departments/epidemiology-biostatistics/biostatistics/icluster
The concept is to formulate the tumor subtypes as the joint latent variable $Z$ that needs to be simultaneously estimated from multiple genomic data types measured on the same set of tumors.


https://www.mskcc.org/departments/epidemiology-biostatistics/biostatistics/icluster
Regression for data integration

- **remMap** — REgularized Multivariate regression for identifying MAster Predictors for fitting multivariate response regression models under the high-dimension–low-sample-size setting.
- Dependence between two datasets, e.g., RNA levels and DNA copy numbers, is modeled through a multivariate response linear regression model
  - RNA levels are responses
  - DNA copy numbers are predictors
Consider multivariate regression with $Q$ response variables $y_1, \ldots, y_Q$ and $P$ prediction variables $x_1, \ldots, x_P$

$$y_q = \sum_{p=1}^{P} X_p \beta_{pq} + \epsilon_q, \quad q = 1, \ldots, Q$$

- The goal is to identify nonzero entries in the $P \times Q$ coefficient matrix $B = (\beta_{pq})$

remMap - Regularized Multivariate Regression for Identifying Master Predictors, https://cran.r-project.org/web/packages/remMap/index.html

Proposed penalization

- **L1 norm penalty** to control the overall sparsity of the coefficient matrix $B$
- **L2 norm penalty** on regression coefficients for each predictor, i.e., the row vectors $C_p \cdot B_p$ - “group” sparse penalty inducing row sparsity of the product matrix $C \cdot B$ (some rows may be entirely zero)

\[
L(B, \lambda_1, \lambda_2) = \frac{1}{2} \| Y - \sum_{p=1}^{P} X_p B_p \|_F^2 + \lambda_1 \sum_{p=1}^{P} \| C_p \cdot B_p \|_1 + \lambda_2 \sum_{p=1}^{P} \| C_p \cdot B_p \|_2
\]

- $C_p$ is the $p$th row of $C = (c_{pq}) = (C_1^T : \ldots : C_P^T)^T$, which is a pre-specified $P \times Q$ 0-1 matrix indicating the coefficients on which penalization is imposed. Based on prior knowledge. Can be set to $c_{p,q} = 0$
- $B_p$ is the $p$th row of $B$
- $\| \cdot \|_F$ - Frobenius norm of matrices
- $\| \cdot \|_1$ and $\| \cdot \|_2$ are the $l_1$ and $l_2$ norms for vectors
- “$\cdot$” - Hadamard product (entry-wise multiplication)
Estimating coefficients

\[ L(B, \lambda_1, \lambda_2) = \frac{1}{2} \| Y - \sum_{p=1}^{P} X_p B_p \|_F^2 + \lambda_1 \sum_{p=1}^{P} \| C_p \cdot B_p \|_1 + \lambda_2 \sum_{p=1}^{P} \| C_p \cdot B_p \|_2 \]

- The \( C_p \) indicator matrix may be set to \( c_{pq} = 1 \)
- An iterative algorithm for solving a convex optimization problem when the two tuning parameters are non-zero. Estimate of the coefficient matrix \( B \) is

\[ \hat{B}(\lambda_1, \lambda_2) := \text{argmin}_B L(B; \lambda_1, \lambda_2) \]
Similarity Network Fusion for data integration

- SNF - Fusing correlation matrices for each data type into one network. Constructing sample similarity for each data type, then merging them into a single similarity network using a nonlinear combination method based on message passing theory.


http://compbio.cs.toronto.edu/SNF/SNF/Software.html
MultiAssayExperiment
The need for MultiAssayExperiment

Need a core data structure to:

- harmonize single-assay data structures
- relate multiple assays & clinical data
- handle missing and replicate observations
- accommodate ID-based and range-based data
- support on-disk representations of big data

https://github.com/waldronlab/MultiAssayExperiment
MultiAssayExperiment R package


http://cancerres.aacrjournals.org/content/77/21/e39
## MultiAssayExperiment R package

**Table 1. Summary of the MultiAssayExperiment API**

<table>
<thead>
<tr>
<th>Category and function</th>
<th>Description</th>
<th>Returned class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MultiAssayExperiment</td>
<td>Create a MultiAssayExperiment object</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>ExperimentList</td>
<td>Create an ExperimentList from a List or list</td>
<td>ExperimentList</td>
</tr>
<tr>
<td>Accessors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>colData</td>
<td>Get or set data that describe the samples</td>
<td>DataFrame</td>
</tr>
<tr>
<td>experiments</td>
<td>Get or set the list of experimental data objects as original classes</td>
<td>ExperimentList</td>
</tr>
<tr>
<td>assays</td>
<td>Get the list of experimental data numeric matrices</td>
<td>SimpleList</td>
</tr>
<tr>
<td>assay</td>
<td>Get the first experimental data numeric matrix</td>
<td>Matrix, matrix-like</td>
</tr>
<tr>
<td>sampleMap</td>
<td>Get or set the map relating observations to subjects</td>
<td>DataFrame</td>
</tr>
<tr>
<td>metadata</td>
<td>Get or set additional data descriptions</td>
<td>List</td>
</tr>
<tr>
<td>rownames</td>
<td>Get row names for all experiments</td>
<td>CharacterList</td>
</tr>
<tr>
<td>colnames</td>
<td>Get column names for all experiments</td>
<td>CharacterList</td>
</tr>
<tr>
<td>Subsetting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mae[i,j,k]</td>
<td>Get rows, columns, and/or experiments</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>mae[i,j]</td>
<td>GRanges, character, integer, logical, List, list</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>mae[,i]</td>
<td>Character, integer, logical, List, list</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>mae[,k]</td>
<td>Character, integer, logical</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>mae[[i]]</td>
<td>Get or set object of arbitrary class from experiments</td>
<td>(Varies)</td>
</tr>
<tr>
<td>mae[[i,k]]</td>
<td>Character, integer, logical</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>mae$scolumn</td>
<td>Get or set colData column</td>
<td>Vector (varies)</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>complete.cases</td>
<td>Identify subjects with complete data in all experiments</td>
<td>Vector (logical)</td>
</tr>
<tr>
<td>duplicated</td>
<td>Identify subjects with replicate observations per experiment</td>
<td>List of LogicalLists</td>
</tr>
<tr>
<td>mergeReplicates</td>
<td>Merge replicate observations within each experiment</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>intersectRows</td>
<td>Return features that are present for all experiments</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>intersectColumns</td>
<td>Return subjects with data available for all experiments</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>prepMultiAssay</td>
<td>Troubleshoot common problems when constructing main class</td>
<td>List</td>
</tr>
<tr>
<td>Reshaping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>longFormat</td>
<td>Return a long and tidy DataFrame with optional colData columns</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>wideFormat</td>
<td>Create a wide DataFrame</td>
<td>MultiAssayExperiment</td>
</tr>
<tr>
<td>Combining</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>Concatenate an experiment</td>
<td>MultiAssayExperiment</td>
</tr>
</tbody>
</table>

http://cancerres.aacrjournals.org/content/77/21/e39
> acc
A MultiAssayExperiment object of 9 listed experiments with user-defined names and respective classes.
Containing an ExperimentList class object of length 9:
[1] RNASeq2GeneNorm: ExpressionSet with 20501 rows and 79 columns
[2] miRNASeqGene: ExpressionSet with 1046 rows and 80 columns
[3] CNASNP: RaggedExperiment with 79861 rows and 180 columns
[4] CNVSNP: RaggedExperiment with 21052 rows and 180 columns
[6] RPPAArray: ExpressionSet with 192 rows and 46 columns
[7] Mutations: RaggedExperiment with 20166 rows and 90 columns
[8] gistica: SummarizedExperiment with 24776 rows and 90 columns
[9] gistict: SummarizedExperiment with 24776 rows and 90 columns
Features:
experiments() - obtain the ExperimentDataList instance
colData() - the primary/phenotype DataFrame
sampleMap() - the sample availability DataFrame
`$, `][` - extract colData columns, subset, or experiment
*Format() - convert into a long or wide DataFrame
assays() - convert ExperimentList to a SimpleList of matrices
MultiAssayExperiment

- MultiAssayExperiment, Bioconductor package for management of multi-assay data
- TCGA data
- How to get the data,

https://github.com/waldronlab/MultiAssayExperiment

https://docs.google.com/spreadsheets/d/1hl64DDS5mqDlYFzDyCY9HAUnxvI1b6hapKP_akFuNPY/edit#gid=0

https://github.com/waldronlab/curatedTCGAData