Semi-Supervised Recursively Partitioned Mixture Models (SS-RPMM) Tutorial

Devin C. Koestler Ph.D.
Post-doctoral research fellow
Quantitative Biosciences program

October 2 2012
Getting Started

The following files are posted at:

Software Devin Koestler

(1) **SSRPMM Functions.r** – R source code for the SS-RPMM functions

(2) **SSRPMM Tutorial.r** – R code for this tutorial and for future reference
SS-RPMM in a nutshell

- Method for identifying profiles of methylation that are associated with a clinical variable of interest (Koestler et al. Bioinformatics 2010)

- Has been used for identifying methylation profiles/classes that are associated with:
  - Survival time in mesothelioma subjects (Koestler et al. Bioinformatics 2010)
  - Birth weight based on placental patterns of DNA methylation (Banister et al. Epigenetics 2011)
  - Bladder cancer case/control status based on peripheral blood DNA methylation signatures (Marsit et al. JCO 2011)

Devin.C.Koestler@dartmouth.edu
Characteristics of SS-RPMM

- 2-stage procedure for identifying phenotypically important profiles of methylation
- Dimension reduction method
- Utilizes a novel model-based clustering framework, the Recursively partitioned mixture model (RPMM) (Houseman et al. BMC Bioinformatics 2008)
- Estimates the number of methylation classes
Origins and history of SS-RPMM

- Clustering samples via RPMM with all CpGs (GoldenGate data)
  - May introduce noise
  - Computationally intensive when J is large

- Top $M$ most variable CpGs
  - Doesn’t include use of clinical data
  - Can miss informative loci
  - $M$ is arbitrarily chosen
  - Dimension Reduction

- SS-RPMM
  - Motivated by the methods of Bair & Tibshirani (2004)
  - Uses clinical data for informing selection of CpGs for clustering
  - The number of CpGs for clustering is determined by a Nested Cross validation
Necessary R-packages

- Note, the following R-packages will need to be installed prior to implementing the SS-RPMM functions described in the succeeding slides:
  1. RPMM
  2. survival
  3. nlme

- The following packages can be downloaded from http://cran.r-project.org/
Step 1

Full Methylation Dataset

Randomly Select

Training Dataset

Testing Dataset

Step 2

Appropriate univariate model controlling for relevant covariates

Identify CpG loci most associated with clinical variable of interest

Step 3

Cross Validation to choose number of CpG loci

Step 4

Fit RPMM using the selected loci

Empirical Bayes class prediction

Predicted Classes Test Association
Step 1

Full Methylation Dataset

Randomly Select

Training Dataset

Testing Dataset

Appropriate univariate model controlling for relevant covariates

Identify CpG loci most associated with clinical variable of interest

Cross Validation to choose number of CpG loci

Fit RPMM using the selected loci

Empirical Bayes class prediction

Predicted Classes Test Association
Step 1 of SS-RPMM

Randomly split the full data into Training and Testing sets.
- May want to do a stratified random split.
  *Example: case/control status

Function: `TrainTestSplit`

Arguments:
(1) `Y`: Data.frame of beta-values for the full data set (n x J)
(2) `Covariates`: Data.frame of covariates for the full data set (n x p)
(3) `Strat`: Stratification variable, if any (i.e., gender, case/con status)
(4) `seed`: will select randomly if not specified
(5) `propTrain`: Proportion of samples allocated to be in training set

Returns: A list whose first item is the training data and whose second item is the testing data
Open up the SSRPMM Tutorial.R file

- **Betas_HNSCC**: data.frame of beta values for the HNSCC data (184 x 26486)
- **Covariates_HNSCC**: data.frame of covariate information for the HNSCC data (184 x 12)
- **Betas_Mesothelioma**: data.frame of beta values for the Mesothelioma data (158 x 1413)
- **Covariates_Mesothelioma**: data.frame of covariate information for the Mesothelioma data (158 x 29)
Full Methylation Dataset

Randomly Select

Training Dataset

Testing Dataset

Step 2

Appropriate univariate model controlling for relevant covariates

Identify CpG loci most associated with clinical variable of interest

Cross Validation to choose number of CpG loci

Fit RPMM using the selected loci

Empirical Bayes class prediction

Predicted Classes Test Association

Devin.C.Koestler@dartmouth.edu
Step 2 of SS-RPMM

Identifying the CpG loci most associated with the clinical variable of interest (COI).

- Two different functions, depending on what type of variable your COI is.

For instances where the COI is not a time-to-event:

Function: `MostImpCpGs`

Arguments:

1. `y`: Data.frame of beta values for the training data set
2. `covariates`: Data.frame of covariates for the training data set
3. `clinvar`: Name of the clinical variable of interest (i.e. “case”)
4. `terms`: Name(s) of the terms to be controlled for in the model (i.e. confounders)
5. `factors`: Name(s) of the terms that should be treated as factors in the model
6. `randomEffect`: Name of a variable you want to be treated as a random effect (i.e. Plate/BeadChip)
7. `is.factor`: is the clinical variable of interest categorial? - TRUE if yes.

Returns: A ranked list of CpG loci and there corresponding absolute (T-score). The larger the absolute T-Score, the stronger the association with COI.
For instances where the COI is time-to-event:

**Function:** MostImpCpGsSurvival  
**Arguments:**
1. \(Y\): Data.frame of beta values for the training data set  
2. \(covariates\): Data.frame of covariates for the training data set  
3. \(times\): Event times (i.e. survival time / time to censoring)  
4. \(censor\): Censoring indicator  
5. \(terms\): Name(s) of the term(s) to be controlled for in the model (i.e. confounders)  
6. \(factors\): Name(s) of the term(s) that should be treated as factors in the model  
7. \(strat\): Name of a variable you want to be used to fit a stratified Cox-Model  

**Returns:** A ranked list of CpG loci and their corresponding absolute (Cox-Score). The larger the absolute Cox-Score, the stronger the association with time-to-event.
A couple remarks about Step 2:

1. Fits univariate models for each of the CpG loci to assess the association between methylation and the COI.

2. High ranking CpGs may be correlated.

3. Takes ~1-3 minutes for GoldenGate data and up to 20 minutes for Infinium data.
   - Admittedly, there are probably ways to improve the existing code so this time can be reduced.

4. Doesn’t tell us how many of the high-ranking CpGs should be used to fit RPMM.
Step 3 of SS-RPMM

Identifies the number of CpGs ($M^*$) with the strongest association with the COI that should be used to fit RPMM

Nested Cross-Validation procedure:
1. Randomly split Training data into a Learning and Validation set
2. Fit RPMM to the Learning data using the $M$ CpGs with the largest T-Score or Cox-Score
3. Predict methylation class for the observations in the Validation set
4. Test the association between the predicted methylation classes in the Validation set and the COI (record p-value from this test)
5. Repeat steps 2-4 with different selections of $M$ (i.e. $M = (5-50)$)
6. Repeat steps 1-5 for different random splits of the Training data into Learning and Validation sets.

Devin.C.Koestler@dartmouth.edu
Step 3 of SS-RPMM

- Two different functions, depending on what type of variable your COI is.

For instances where the COI is not a time-to-event:

**Function:** NestedXValidation

**Arguments:**
(1) $\mathbf{y}$: Data.frame of beta values for the training data set
(2) covariates: Data.frame of covariates for the training data set
(3) Tscores: T-score object from Step 2
(4) clinvar: Clinical variable of interest (i.e. “case”) given as a vector
(5) vartype: Type of variable for the COI (i.e. binary, categorical, or continuous)
(6) mrange: Range of “M” (i.e. minimum and maximum possible “M”)
(7) method: Fits a guassian- or beta-distributed RPMM.
(8) $L$: Number of nested splits of the training data, “L”
(9) seeds: Vector of seeds (length must be equal to “L”)

**Returns:** Median p-values for each specification of “M”. Select the “M” that yields the lowest median p-value or do a smoothing procedure on the median p-values for selection of “M”.

Step 3 of SS-RPMM!
Step 3 of SS-RPMM

For instances where the COI is a time-to-event:

**Function:** NestedXValidationSurvival

**Arguments:**
1. \(Y\): Data.frame of beta values for the training data set
2. \(covariates\): Data.frame of covariates for the training data set
3. \(CoxScores\): Cox-score object from Step 2
4. \(times\): Event times (i.e. survival time / time to censoring)
5. \(censor\): Censoring indicator
6. \(mrange\): Range of “M” (i.e. minimum and maximum possible “M”)  
7. \(method\): Fits a guassian- or beta-distributed RPMM.
8. \(L\): Number of nested splits of the training data, “L”
9. \(seeds\): Vector of seeds (length must be equal to “L”)

**Returns:** Median p-values for each specification of “M”. Select the “M” that yields the lowest median p-value or smoothing on the median p-values can be done for suitable selection of “M”
Step 4

Full Methylation Dataset

Randomly Select

Training Dataset

Appropriate univariate model controlling for relevant covariates

Identify CpG loci most associated with clinical variable of interest

Cross Validation to choose number of CpG loci

Fit RPMM using the selected loci

Empirical Bayes class prediction

Predicted Classes Test Association
Fit an RPMM to the Training data using the $M^*$ CpG loci with the largest T-Scores/Cox-Scores – Based on this solution, predict methylation class membership for the observations in the Test data.

Function: PredMethClasses

Arguments:
1. $Y_{\text{train}}$: Data.frame of beta values for the training data set
2. $Y_{\text{test}}$: Data.frame of beta values for the testing data set
3. Score: T-Score/Cox-Score object from Step 2
4. $M$: Number of high-ranking CpGs to be used to fit RPMM – $M^*$ determined from Step 3
5. method: Fits a gaussian- or beta-distributed RPMM

Returns: Factor vector of methylation class assignments for the observations in the Testing data
### Recap on functions used based on what type of COI we’re interested in

#### Clinical Outcome of Interest (COI)

<table>
<thead>
<tr>
<th>Steps of SS-RPMM</th>
<th>Continuous, Binary, Categorical (i.e. birth weight, case/control status, smoking status)</th>
<th>Time to Event (i.e. survival, recurrence, time to metastasis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>TrainTestSplit</td>
<td>TrainTestSplit</td>
</tr>
<tr>
<td>Step 2</td>
<td>MostImpCpGs</td>
<td>MostImpCpGsSurvival</td>
</tr>
<tr>
<td>Step 3</td>
<td>NestedXValidation</td>
<td>NestedXValidationSurvival</td>
</tr>
<tr>
<td>Step 4</td>
<td>PredMethClasses</td>
<td>PredMethClasses</td>
</tr>
</tbody>
</table>

*See “SSRPMM Functions.r” for further details regarding function arguments.*
Post SS-RPMM Analyses

- Test the association between the predicted methylation classes in the testing data and the COI
  - Permutation Chi-Square or Permutation Kruskal Wallis test
  - Log-rank test if COI is a time-to-event variable
  - GLM or Linear regression to control for confounders
  - Cox-Proportional Hazards model to control for confounders if the COI is time-to-event

- Visualization of the SS-RPMM results
  - Heatmap of the testing data by predicted methylation class
  - Barplot – i.e percent case/control by predicted methylation class
  - Boxplot – i.e. birthweight by predicted methylation class
  - Kaplan Meier survival curves by predicted methylation class
  - ROC curves to assess the predictive performance of the predicted methylation classes
Questions and suggestions

For questions and suggestions regarding the SS-RPMM code or implementation, please do not hesitate to contact me via email at:

Devin.Koestler@dartmouth.edu