Hierarchical Clustering and Dynamic Branch Cutting

Peter Langfelder

Dept. of Human Genetics, UCLA
Outline

• What is clustering?
• Brief overview of various clustering methods
• Hierarchical clustering
• Cluster identification in hierarchical clustering trees
  – Fixed-height tree cut
  – Adaptive-height tree cut (Dynamic Tree Cut)
What is clustering?

- Input: pair-wise dissimilarities between objects

<table>
<thead>
<tr>
<th>Gene.9</th>
<th>Gene.8</th>
<th>Gene.7</th>
<th>Gene.5</th>
<th>Gene.6</th>
<th>Gene.10</th>
<th>Gene.4</th>
<th>Gene.3</th>
<th>Gene.1</th>
<th>Gene.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.63</td>
<td>0.55</td>
<td>0.69</td>
<td>1</td>
<td>0.94</td>
<td>0.82</td>
<td>0.92</td>
<td>0.98</td>
</tr>
<tr>
<td>1</td>
<td>0.63</td>
<td>0</td>
<td>0.49</td>
<td>0.56</td>
<td>1</td>
<td>1</td>
<td>0.76</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.55</td>
<td>0.49</td>
<td>0</td>
<td>0.44</td>
<td>1</td>
<td>0.94</td>
<td>0.88</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.69</td>
<td>0.56</td>
<td>0.44</td>
<td>0</td>
<td>0.95</td>
<td>1</td>
<td>0.9</td>
<td>1</td>
<td>0.91</td>
</tr>
<tr>
<td>0.9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.95</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
<td>0.82</td>
</tr>
<tr>
<td>0.9</td>
<td>0.94</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.76</td>
<td>0.82</td>
</tr>
<tr>
<td>0.98</td>
<td>0.82</td>
<td>0.76</td>
<td>0.88</td>
<td>0.9</td>
<td>1</td>
<td>0.76</td>
<td>0</td>
<td>0.67</td>
<td>0.56</td>
</tr>
<tr>
<td>1</td>
<td>0.92</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
<td>0.82</td>
<td>0.67</td>
<td>0</td>
<td>0.38</td>
</tr>
<tr>
<td>1</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
<td>0.91</td>
<td>0.82</td>
<td>0.65</td>
<td>0.56</td>
<td>0.38</td>
<td>0</td>
</tr>
</tbody>
</table>
What is clustering?

- Input: pair-wise dissimilarities between objects
- Aim: find clusters (groups) of objects that are closely related according to the given (dis-)similarity measure
What is clustering?

- **Input:** pair-wise dissimilarities between objects
- **Aim:** find clusters (groups) of objects that are closely related according to the (dis-)similarity measure
- **Output:** a cluster label for each object
Clustering example for non-experts

• Example:
  – **Objects** = people living in California
  – **Dissimilarity** = geographical distance of their homes

• Resulting clusters = groups of people who live close to one another: cities, towns, neighborhoods

• Question: how to assign people who live outside of towns and cities?
How to deal with objects that are far from clusters?

• Three possible answers:
  • Create a separate cluster for each outlying object
  • Assign them to the nearest cluster
  • Leave them “unassigned”

• Most clustering methods produce a partition in which every object is assigned to a cluster

• Sometimes this is desirable: for example, assigning people to the nearest town is good for the mail delivery service

• In biomedical applications it is often a bad idea
Applications of clustering in biomedical research

• Clustering of patients
  – Discovery of subtypes of heterogeneous diseases such as cancer, neurological diseases, etc

• Clustering of high-throughput molecular phenotypes (measurements) such as gene expression, methylation, proteomic, metabolomic etc.
  – Part of network analysis techniques (WGCNA)
  – Discovery of transcriptional, methylation organization of genome, protein complexes, etc.
There are many clustering methods!

- K-means, Partitioning Around Medoids (PAM), Self-Organizing Maps (SOM), Model-based clustering approaches, multitude of other methods

- Hierarchical clustering: good method for exploratory data analysis because it works well with high-dimensional data, provides visualization, does not require specifying number of clusters beforehand
Cluster identification using hierarchical clustering

Two-step process:

- Construct a hierarchical clustering tree (dendrogram) that provides information on how objects are iteratively merged together.
Cluster identification using hierarchical clustering

- Two-step process:
  - Construct a hierarchical clustering tree (dendrogram) that provides information on how objects are iteratively merged together
  - Identify branches that correspond to clusters
  - Label branches by numbers or colors
Hierarchical clustering cartoon

- Clustering of 10 simulated objects (say gene expression profiles)
  - Start with (dis-)similarity matrix
  - White: distant (dissimilar) objects
  - Red: close (similar) objects
Hierarchical clustering cartoon

- Find the two closest objects (here Gene.1 and Gene.2)...
Hierarchical clustering cartoon

- Find the two closest objects (here Gene.1 and Gene.2)...

- ...and merge them
Hierarchical clustering cartoon

- Find the next two closest objects...
Hierarchical clustering cartoon

- Find the next two closest objects... ...and merge them
Hierarchical clustering cartoon

- Two closest objects again: Gene.7 and the 5-6 cluster
Hierarchical clustering cartoon

- Merge Gene.7 with the 5-6 branch
Hierarchical clustering cartoon

- Next: Gene.3 and 1-2 cluster
Hierarchical clustering cartoon

- Next: Gene.3 and 1-2 cluster
Hierarchical clustering cartoon

- Closest: Gene.8 and 5-6-7 cluster
Hierarchical clustering cartoon

- Gene.8 and 5-6-7 cluster
Hierarchical clustering cartoon

- Closest: Gene.4 and 1-2-3 cluster
Hierarchical clustering cartoon

- Merge Gene.4 and 1-2-3 cluster
Hierarchical clustering cartoon

- Closest: Gene.10 and 1-2-3-4 cluster
Hierarchical clustering cartoon

- Merge Gene.10 and 1-2-3-4 cluster
Hierarchical clustering cartoon

- Closest: Clusters 1-2-3-4-10 and 5-6-7-8
Hierarchical clustering cartoon

- Merge Clusters 1-2-3-4-10 and 5-6-7-8
Hierarchical clustering cartoon

- Closest: Gene.9 and the large cluster (1-2-3-4-5-6-7-8-10)
Hierarchical clustering cartoon

- Merge Gene.9 and the large cluster (1-2-3-4-5-6-7-8-10)
- The clustering ends, we have a complete tree
Final hierarchical clustering tree
a.k.a. dendrogram
Multiple versions of hierarchical clustering

- Different versions of hierarchical clustering differ in how they measure dissimilarity between an object and a cluster

  - Average linkage: average the dissimilarities between all objects
  - Single linkage: take the minimum dissimilarity
  - Complete linkage: take the maximum dissimilarity
  - Other choices are available
Hierarchical clustering in R

- Function `hclust` in (standard) package stats

- Two important arguments:
  - `d`: distance structure representing dissimilarities between objects
  - `method`: hierarchical clustering version. We usually use "average".

- Result: a hierarchical clustering tree that can be displayed using `plot(...)` or used as input to other functions such as tree cutting functions

- Alternative for very large data sets: `hclust` from package fastcluster
How to identify clusters in hierarchical clustering trees?

"Tree cutting", "Branch pruning" of hierarchical clustering trees
Identifying clusters in dendrograms

- Visual impression: there are two clusters (branches of the hierarchical tree)
Simple solution for simple trees

- Pick a suitable constant height
Simple solution for simple trees

- Pick a suitable constant height (here 0.97)
Simple solution for simple trees

- Pick a suitable constant height (here 0.97)
- Cut branches at the height
Simple solution for simple trees

• Pick a suitable constant height (here 0.97)
• Cut branches at the height
• Each individual branch is a cluster
Simple solution for simple trees

- Pick a suitable constant height (here 0.97)
- Cut branches at the height
- Each individual branch is a cluster
- Enforce a minimum cluster size to avoid very small clusters
Cut height must be chosen carefully!

- Different cut heights lead to very different results
- Proper cut height setting requires an intelligent operator
- In general, each application will require a different cut height
- This is a major disadvantage of constant-height tree cut
Example genomic application

- Human brain expression data (Oldham et al 2006)
- Modules group together genes expressed in specific brain regions
WGCNA functions `cutreeStatic` and `cutreeStaticColor`, based on function `cutree`

- `cutreeStatic` returns numeric labels (1,2,3,...; unassigned label is 0)
- `cutreeStaticColor` returns color labels (turquoise, blue, ...; unassigned color is grey)

Both function take as input a hierarchical cluster tree, cut height and minimum cluster size

Use `help("cutreeStatic")` to see more details
The simple solution does not always work for complicated trees

• Clustering of mouse adipose expression data: no single cut height captures the prominent branches
Solution: make the cut height adaptive

Dynamic Tree Cut

Dynamic Tree Cut

- Branches are followed bottom to top
- When two branches merge, they are evaluated using shape criteria such as minimum number of objects (genes), their core scatter and the gap between the branches
- If the branches meet criteria for being a module, they are called separate modules, otherwise they are merged
Start a new branch
Start a second branch
Add object to branch 2
Add object to a branch 1
Add objects to branches 1 and 2
Two branches merge
Dynamic Tree Cut

- Branches are followed bottom to top
- When two branches merge, they are evaluated using shape criteria such as minimum number of objects (genes), their core scatter and the gap between the branches
- If the branches meet criteria for being a cluster, they are called separate clusters, otherwise they are merged into a single cluster
The good, the bad, and the flexible

- Bad news: shape criteria are heuristic and somewhat arbitrary

- Good news for general users: they often work well in finding meaningful clusters

- Good news for power users who would like to use their own criteria: cutreeDynamic is flexible!

- “plug-in” system allows users to use their own branch similarity criteria whenever 2 branches merge
Examples of external branch similarity criteria

- For genomic data: one often wants to merge modules whose expression profiles are very similar (“correlation of eigengenes is too high”): this is easily accomplished with an external criterion.

- One may want to merge branches whose split disappears when one perturbs the data (e.g., in a resampling study).
Optional assignment of outlying objects to nearest branch

- Optionally the method can include a Partitioning Around Medoids (PAM)-like step to assign outlying objects
  - Clustering tree is not sufficient to assign those objects

Cluster dendrogram and module colors
Optionally the method can include a Partitioning Around Medoids (PAM)-like step to assign outlying objects

- Assign those objects to nearest cluster
Function cutreeDynamic in the R package dynamicTreeCut

```r
library(dynamicTreeCut)
help("cutreeDynamic")
```

Input:
- clustering tree
- dissimilarity matrix that was used to produce the tree
- multiple options to fine-tune cluster criteria and PAM stage

Most important options:
- `DeepSplit` (0-4): controls how finely clusters will be split
- `pamStage` (FALSE or TRUE): turns PAM stage off/on
Effect of \texttt{deepSplit}

- \texttt{deepSplit} controls how finely the branches should be split
- Higher values give more smaller modules, low values (0) give fewer larger modules
PAM stage: assigning more distant objects to clusters

- Optional, by default enabled, PAM stage allows the user to assign more outlying objects to clusters
- Without PAM stage, sometimes there are many "grey" genes
- With PAM stage the dendrogram is sometimes more difficult to interpret

```
<table>
<thead>
<tr>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.3</td>
</tr>
</tbody>
</table>
```

depthSplit=2, no PAM

depthSplit=2, PAM
Hierarchical clustering is a useful method for finding groups of similar objects. It produces a hierarchical clustering tree that can be visualized. Clusters correspond to branches of the tree; cluster identification is also known as tree cutting or branch pruning. Simple methods for cluster identification are not always suitable, especially in complicated clustering trees. Dynamic Tree Cut is capable of identifying clusters in complicated clustering trees. Most important arguments are `deepSplit` and `pamStage`. A single setting works well and produces comparable results in many applications: Dynamic Tree Cut is suitable for automation.
Limitations

- Hierarchical clustering is heuristic - does not optimize a cost (penalty) function
- Hierarchical clustering is not "stable": relatively small changes in data can produce different trees
  - This can be remedied using a resampling or other perturbation study
- Visualization is imperfect (all visualizations of high-dimensional data are imperfect); users should not rely too much on the dendrogram
  - This applies especially when PAM stage is used
- Dynamic Tree Cut uses heuristic criteria for deciding whether a branch is a cluster; the criteria are by no means unique
Usage in R

- Hierarchical clustering: function `hclust`
- Constant height tree cut: `cutreeStatic`, `cutreeStaticColor` in package WGCNA
- Dynamic Tree Cut: `cutreeDynamic` in package dynamicTreeCut

Further reading

Langfelder P, Zhang B, Horvath S
Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R.
Bioinformatics 2008 24(5):719-720
http://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/BranchCutting/