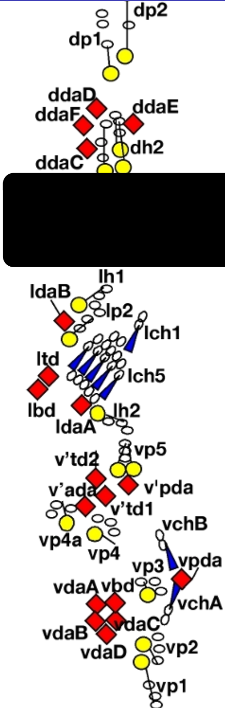
Clustering in Bioinformatics
Bio2, Lecture8

Dr. Ian Simpson

Centre for Integrative Physiology
University of Edinburgh

March 10th 2010



Clustering

Introduction

- Clustering is the classification of data into subsets so that members of each subset are similar (and ideally more similar to each other than to members of other subsets)
- There are literally hundreds of different methods that can be used to cluster data
- Clustering finds application in a huge number of areas such as Biology, Medicine, Geology, Chemistry, Market Research, Commerce, Social Networking...
- We are interested in using clustering to both categorise and prioritise biological data

Clustering

Introduction

- Clustering is the classification of data into subsets so that members of each subset are similar (and ideally more similar to each other than to members of other subsets)
- There are literally hundreds of different methods that can be used to cluster data
- Clustering finds application in a huge number of areas such as Biology, Medicine, Geology, Chemistry, Market Research, Commerce, Social Networking...
- We are interested in using clustering to both categorise and prioritise biological data

Clustering

Introduction

- Clustering is the classification of data into subsets so that members of each subset are similar (and ideally more similar to each other than to members of other subsets)
- There are literally hundreds of different methods that can be used to cluster data
- Clustering finds application in a huge number of areas such as Biology, Medicine, Geology, Chemistry, Market Research, Commerce, Social Networking...
- We are interested in using clustering to both categorise and prioritise biological data

Clustering

Introduction

- Clustering is the classification of data into subsets so that members of each subset are similar (and ideally more similar to each other than to members of other subsets)
- There are literally hundreds of different methods that can be used to cluster data
- Clustering finds application in a huge number of areas such as Biology, Medicine, Geology, Chemistry, Market Research, Commerce, Social Networking...
- We are interested in using clustering to both categorise and prioritise biological data

Features of unsupervised clustering

Advantages

- We make no assumptions about the structure of the data and by not introducing priors (or a supervised scheme) we don't add bias
- consistent results, i.e. initialising with the same conditions produces the same results

Disadvantages

- Produces clusters even when the data has no structure
- Not clear which method is best or which parameters to set
- Rarely produce any indication of the robustness of the clusters themselves or the members of the clusters (so not good for prioritisation within a cluster)
- The noise inherent in biological data sets is not particularly well suited to unsupervised clustering

Features of unsupervised clustering

Advantages

- We make no assumptions about the structure of the data and by not introducing priors (or a supervised scheme) we don't add bias
- consistent results, i.e. initialising with the same conditions produces the same results

Disadvantages

- Produces clusters even when the data has no structure
- Not clear which method is best or which parameters to set
- Rarely produce any indication of the robustness of the clusters themselves or the members of the clusters (so not good for prioritisation within a cluster)
- The noise inherent in biological data sets is not particularly well suited to unsupervised clustering

Features of unsupervised clustering

Advantages

- We make no assumptions about the structure of the data and by not introducing priors (or a supervised scheme) we don't add bias
- consistent results, i.e. initialising with the same conditions produces the same results

Disadvantages

- Produces clusters even when the data has no structure
- Not clear which method is best or which parameters to set
- Rarely produce any indication of the robustness of the clusters themselves or the members of the clusters (so not good for prioritisation within a cluster)
- The noise inherent in biological data sets is not particularly well suited to unsupervised clustering

Features of unsupervised clustering

Advantages

- We make no assumptions about the structure of the data and by not introducing priors (or a supervised scheme) we don’t add bias
- consistent results, i.e. initialising with the same conditions produces the same results

Disadvantages

- Produces clusters even when the data has no structure
- Not clear which method is best or which parameters to set
- Rarely produce any indication of the robustness of the clusters themselves or the members of the clusters (so not good for prioritisation within a cluster)
- The noise inherent in biological data sets is not particularly well suited to unsupervised clustering

Features of unsupervised clustering

Advantages

- We make no assumptions about the structure of the data and by not introducing priors (or a supervised scheme) we don’t add bias
- consistent results, i.e. initialising with the same conditions produces the same results

Disadvantages

- Produces clusters even when the data has no structure
- Not clear which method is best or which parameters to set
- Rarely produce any indication of the robustness of the clusters themselves or the members of the clusters (so not good for prioritisation within a cluster)
- The noise inherent in biological data sets is not particularly well suited to unsupervised clustering

Features of unsupervised clustering

Advantages

- We make no assumptions about the structure of the data and by not introducing priors (or a supervised scheme) we don’t add bias
- consistent results, i.e. initialising with the same conditions produces the same results

Disadvantages

- Produces clusters even when the data has no structure
- Not clear which method is best or which parameters to set
- Rarely produce any indication of the robustness of the clusters themselves or the members of the clusters (so not good for prioritisation within a cluster)
- The noise inherent in biological data sets is not particularly well suited to unsupervised clustering

Heirarchical Clustering

Description

- Heirarchical clustering uses either a bottom-up (agglomerative) or top-down (divisive) approach to group elements
- The differences between elements are calclated using a distance metric, often one of euclidean, manhattan or cosine (for high-D)
- For agglomerative clustering an iterated process begins with each element as a cluster
- In the single-linkage method the two closest clusters are merged, the minimum distance is then calculated between the closest elements of this cluster and the closest member of the next closest cluster
- The process is repeated until there is only one cluster left
- The output is a tree (dendrogram) which has to be cut at an appropriate height to reveal the clusters (next slide)

Heirarchical Clustering

Description

- Heirarchical clustering uses either a bottom-up (agglomerative) or top-down (divisive) approach to group elements
- The differences between elements are calclated using a distance metric, often one of euclidean, manhattan or cosine (for high-D)
- For agglomerative clustering an iterated process begins with each element as a cluster
- In the single-linkage method the two closest clusters are merged, the minimum distance is then calculated between the closest elements of this cluster and the closest member of the next closest cluster
- The process is repeated until there is only one cluster left
- The output is a tree (dendrogram) which has to be cut at an appropriate height to reveal the clusters (next slide)

Heirarchical Clustering

Description

- Heirarchical clustering uses either a bottom-up (agglomerative) or top-down (divisive) approach to group elements
- The differences between elements are calclated using a distance metric, often one of euclidean, manhattan or cosine (for high-D)
- For agglomerative clustering an iterated process begins with each element as a cluster
- In the single-linkage method the two closest clusters are merged, the minimum distance is then calculated between the closest elements of this cluster and the closest member of the next closest cluster
- The process is repeated until there is only one cluster left
- The output is a tree (dendrogram) which has to be cut at an appropriate height to reveal the clusters (next slide)

Heirarchical Clustering

Description

- Heirarchical clustering uses either a bottom-up (agglomerative) or top-down (divisive) approach to group elements
- The differences between elements are calclated using a distance metric, often one of euclidean, manhattan or cosine (for high-D)
- For agglomerative clustering an iterated process begins with each element as a cluster
- In the single-linkage method the two closest clusters are merged, the minimum distance is then calculated between the closest elements of this cluster and the closest member of the next closest cluster
- The process is repeated until there is only one cluster left
- The output is a tree (dendrogram) which has to be cut at an appropriate height to reveal the clusters (next slide)

Heirarchical Clustering

Description

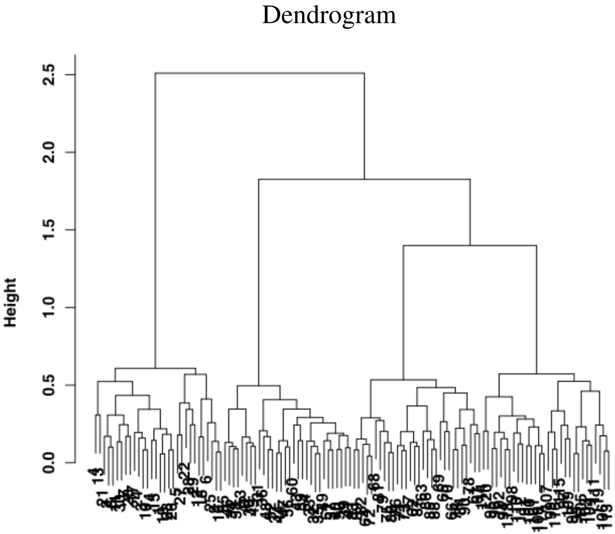
- Heirarchical clustering uses either a bottom-up (agglomerative) or top-down (divisive) approach to group elements
- The differences between elements are calclated using a distance metric, often one of euclidean, manhattan or cosine (for high-D)
- For agglomerative clustering an iterated process begins with each element as a cluster
- In the single-linkage method the two closest clusters are merged, the minimum distance is then calculated between the closest elements of this cluster and the closest member of the next closest cluster
- The process is repeated until there is only one cluster left
- The output is a tree (dendrogram) which has to be cut at an appropriate height to reveal the clusters (next slide)

Heirarchical Clustering

Description

- Heirarchical clustering uses either a bottom-up (agglomerative) or top-down (divisive) approach to group elements
- The differences between elements are calclated using a distance metric, often one of euclidean, manhattan or cosine (for high-D)
- For agglomerative clustering an iterated process begins with each element as a cluster
- In the single-linkage method the two closest clusters are merged, the minimum distance is then calculated between the closest elements of this cluster and the closest member of the next closest cluster
- The process is repeated until there is only one cluster left
- The output is a tree (dendrogram) which has to be cut at an appropriate height to reveal the clusters (next slide)

Heirarchical Clustering



Heirarchical Clustering

Varieties

- single linkage - minimum distance between elements of each cluster
- complete linkage - maximum distance between elements of each cluster
- UPGMA - average linkage clustering, i.e. the average distance between elements of each cluster
- various others based on changes in variance, such as minimise the variance on merging etc..
- can also do the reverse "divisive" heirarchical clustering

Heirarchical Clustering

Varieties

- single linkage - minimum distance between elements of each cluster
- complete linkage - maximum distance between elements of each cluster
- UPGMA - average linkage clustering, i.e. the average distance between elements of each cluster
- various others based on changes in variance, such as minimise the variance on merging etc..
- can also do the reverse "divisive" heirarchical clustering

Heirarchical Clustering

Varieties

- single linkage - minimum distance between elements of each cluster
- complete linkage - maximum distance between elements of each cluster
- UPGMA - average linkage clustering, i.e. the average distance between elements of each cluster
- various others based on changes in variance, such as minimise the variance on merging etc..
- can also do the reverse "divisive" heirarchical clustering

Heirarchical Clustering

Varieties

- single linkage - minimum distance between elements of each cluster
- complete linkage - maximum distance between elements of each cluster
- UPGMA - average linkage clustering, i.e. the average distance between elements of each cluster
- various others based on changes in variance, such as minimise the variance on merging etc..
- can also do the reverse "divisive" heirarchical clustering

Heirarchical Clustering

Varieties

- single linkage - minimum distance between elements of each cluster
- complete linkage - maximum distance between elements of each cluster
- UPGMA - average linkage clustering, i.e. the average distance between elements of each cluster
- various others based on changes in variance, such as minimise the variance on merging etc..
- can also do the reverse "divisive" heirarchical clustering

Partitional Clustering

Description

- Again we chose a distance metric to quantify the properties of each element, in addition we must chose the cluster number (k) at the start
- We begin by randomly chosing k centoids (centres) from the elements
- Next we find the closest element to each center and calculate the centroid of the two (nominally the average)
- We repeat this process until a convergence criterion has been met, often maximising distance between clusters and minimising variance within clusters
- Note that unlike the heirarchical clustering described previously k-means can produce different results depending on the initial centroids and on the success of convergence

Partitional Clustering

Description

- Again we chose a distance metric to quantify the properties of each element, in addition we must chose the cluster number (k) at the start
- We begin by randomly chosing k centoids (centres) from the elements
- Next we find the closest element to each center and calculate the centroid of the two (nominally the average)
- We repeat this process until a convergence criterion has been met, often maximising distance between clusters and minimising variance within clusters
- Note that unlike the heirarchical clustering described previously k-means can produce different results depending on the initial centroids and on the success of convergence

Partitional Clustering

Description

- Again we chose a distance metric to quantify the properties of each element, in addition we must chose the cluster number (k) at the start
- We begin by randomly chosing k centoids (centres) from the elements
- Next we find the closest element to each center and calculate the centroid of the two (nominally the average)
- We repeat this process until a convergence criterion has been met, often maximising distance between clusters and minimising variance within clusters
- Note that unlike the heirarchical clustering described previously k-means can produce different results depending on the initial centroids and on the success of convergence

Partitional Clustering

Description

- Again we chose a distance metric to quantify the properties of each element, in addition we must chose the cluster number (k) at the start
- We begin by randomly chosing k centoids (centres) from the elements
- Next we find the closest element to each center and calculate the centroid of the two (nominally the average)
- We repeat this process until a convergence criterion has been met, often maximising distance between clusters and minimising variance within clusters
- Note that unlike the heirarchical clustering described previously k-means can produce different results depending on the initial centroids and on the success of convergence

Partitional Clustering

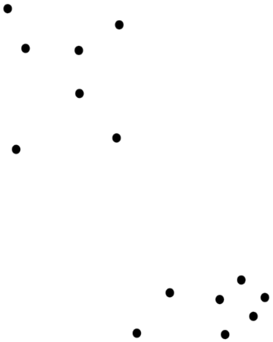
Description

- Again we chose a distance metric to quantify the properties of each element, in addition we must chose the cluster number (k) at the start
- We begin by randomly chosing k centoids (centres) from the elements
- Next we find the closest element to each center and calculate the centroid of the two (nominally the average)
- We repeat this process until a convergence criterion has been met, often maximising distance between clusters and minimising variance within clusters
- Note that unlike the heirarchical clustering described previously k-means can produce different results depending on the initial centroids and on the success of convergence

Partitional Clustering

K-means clustering

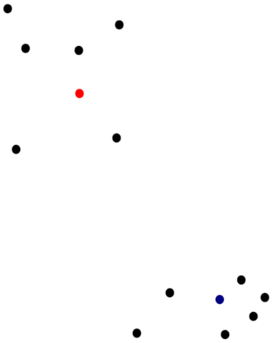
- We start with a simple example of data points distributed in 2D space



Partitional Clustering

K-means clustering

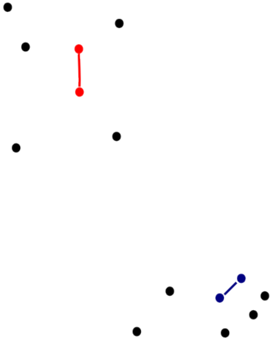
- Begin by assigning start points for k clusters



Partitional Clustering

K-means clustering

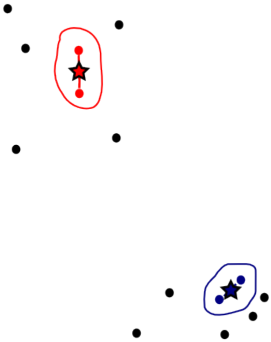
- Find the closest member



Partitional Clustering

K-means clustering

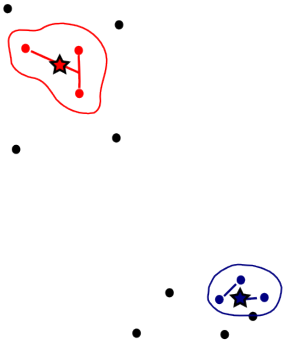
- Recalculate the centre of the cluster (often this is the medoid rather than average as shown here



Partitional Clustering

K-means clustering

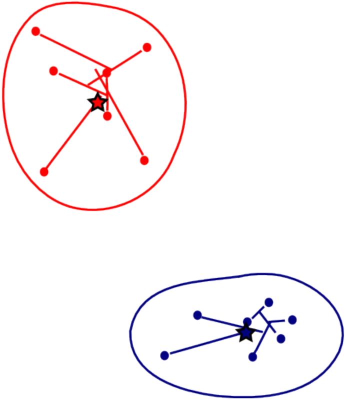
- Repeat the process



Partitional Clustering

K-means clustering

- Finish when the change in centre is minimised
- i.e. if we now included a member from the other cluster the centre would move a lot
- we minimise intra-cluster variation and maximise inter-cluster variation (distance)



Problems with the clustering process

- Most clustering algorithms need to be provided with the cluster number
- There are many classes of clustering method

  partitional
  hierarchical
  fuzzy
  density based
  modelled
- There are many distance metrics (similarity scoring methods)

  euclidean, pearson, Manhattan, cosine, Mahalanobis, Hamming...
- There are many scoring systems to assess success

  GAP statistic, Mean, Median Split Silhouette, Elbow plot...

We need methods that help us to chose the algorithm, conditions and cluster number

Problems with the clustering process

- Most clustering algorithms need to be provided with the cluster number
- There are many classes of clustering method

  partitional
  hierarchical
  fuzzy
  density based
  modelled
- There are many distance metrics (similarity scoring methods)

  euclidean, pearson, Manhattan, cosine, Mahalanobis, Hamming...
- There are many scoring systems to assess success

  GAP statistic, Mean, Median Split Silhouette, Elbow plot...

We need methods that help us to chose the algorithm, conditions and cluster number

Problems with the clustering process

- Most clustering algorithms need to be provided with the cluster number
- There are many classes of clustering method

  partitional
  hierarchical
  fuzzy
  density based
  modelled
- There are many distance metrics (similarity scoring methods)

  euclidean, pearson, Manhattan, cosine, Mahalanobis, Hamming...
- There are many scoring systems to assess success

  GAP statistic, Mean, Median Split Silhouette, Elbow plot...

We need methods that help us to chose the algorithm, conditions and cluster number

Problems with the clustering process

- Most clustering algorithms need to be provided with the cluster number
- There are many classes of clustering method

  partitional
  hierarchical
  fuzzy
  density based
  modelled
- There are many distance metrics (similarity scoring methods)

  euclidean, pearson, Manhattan, cosine, Mahalanobis, Hamming...
- There are many scoring systems to assess success

  GAP statistic, Mean, Median Split Silhouette, Elbow plot...

We need methods that help us to chose the algorithm, conditions and cluster number

Problems with the clustering process

- Most clustering algorithms need to be provided with the cluster number
- There are many classes of clustering method

  partitional
  hierarchical
  fuzzy
  density based
  modelled
- There are many distance metrics (similarity scoring methods)

  euclidean, pearson, Manhattan, cosine, Mahalanobis, Hamming...
- There are many scoring systems to assess success

  GAP statistic, Mean, Median Split Silhouette, Elbow plot...

We need methods that help us to chose the algorithm, conditions and cluster number

Properties of an clustering efficacy method

- Statistically principled - we need to be able to assess cluster and membership robustness
- Applicable to the general case - it needs to work for any algorithm
- Computationally tractable - relatively fast with possibility of parallelisation
- Integratation of clustering results from different methods for comparison
- Ideally assist in cluster number determination

consensus clustering

Properties of an clustering efficacy method

- Statistically principled - we need to be able to assess cluster and membership robustness
- Applicable to the general case - it needs to work for any algorithm
- Computationally tractable - relatively fast with possibility of parallelisation
- Integratation of clustering results from different methods for comparison
- Ideally assist in cluster number determination

consensus clustering

Properties of an clustering efficacy method

- Statistically principled - we need to be able to assess cluster and membership robustness
- Applicable to the general case - it needs to work for any algorithm
- Computationally tractable - relatively fast with possibility of parallelisation
- Integratation of clustering results from different methods for comparison
- Ideally assist in cluster number determination

consensus clustering

Properties of an clustering efficacy method

- Statistically principled - we need to be able to assess cluster and membership robustness
- Applicable to the general case - it needs to work for any algorithm
- Computationally tractable - relatively fast with possibility of parallelisation
- Integratation of clustering results from different methods for comparison
- Ideally assist in cluster number determination

consensus clustering

Properties of an clustering efficacy method

- Statistically principled - we need to be able to assess cluster and membership robustness
- Applicable to the general case - it needs to work for any algorithm
- Computationally tractable - relatively fast with possibility of parallelisation
- Integratation of clustering results from different methods for comparison
- Ideally assist in cluster number determination

consensus clustering

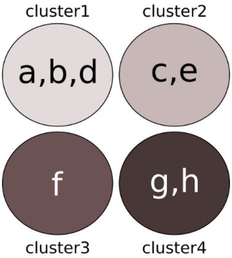
Properties of an clustering efficacy method

- Statistically principled - we need to be able to assess cluster and membership robustness
- Applicable to the general case - it needs to work for any algorithm
- Computationally tractable - relatively fast with possibility of parallelisation
- Integratation of clustering results from different methods for comparison
- Ideally assist in cluster number determination

consensus clustering

The connectivity matrix

cluster membership



cluster membership indices

Indices	Members
I_1 = 1,2,4	a,b,d
I_2 = 3,5	c,e
I_3 = 6	f
I_4 = 7,8	g,h

simple connectivity matrix

	a	b	c	d	e	f	g	h
a	1	1	0	1	0	0	0	0
b	1	1	0	1	0	0	0	0
c	0	0	1	0	1	0	0	0
d	1	1	0	1	0	0	0	0
e	0	0	1	0	1	0	0	0
f	0	0	0	0	0	1	0	0
g	0	0	0	0	0	0	1	1
h	0	0	0	0	0	0	1	1

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

- In order to assess robustness we will cluster the expression data may times using only a sample of the rows
- From these results we will calculate the connectivity matrix and the identity matrix (which were drawn)
- We calculate the average connectivity between any two members normalised against their sampling frequency
- The resulting matrix is called the consensus matrix and measures the average connectedness of any two members
- This process can be carried out using any combination of clustering algorithms and/or parameters
- The variation of consensus matrix over cluster number (k) can be used to derive the optimal k
- The consensus matrix can be used to calculate cluster robustness and membership robustness

Ensemble clustering - a re-sampling approach

Example of a re-sample where the clusters produced are always the same

connectivity matrix

	a	b	c	d
a	2	1	0	0
b	1	2	0	0
c	0	0	2	2
d	0	0	2	3

identity matrix

	a	b	c	d
a	2	1	1	2
b	1	2	1	2
c	1	1	2	2
d	2	2	2	3

consensus matrix

	a	b	c	d
a	1	1	0	0
b	1	1	0	0
c	0	0	1	1
d	0	0	1	1

i.e. (a,b) and (c,d) always cluster together if they are in the draw together

Cluster consensus

	a	b	c	d
1	1	1	0	0
2	0	0	1	1

Metrics to assess the efficacy of clustering

connectivity matrix

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items i and j belong to the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

consensus matrix

$$\mathcal{M}(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)}$$

cluster robustness

$$m(k) = \frac{1}{N_k(N_k-1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} \mathcal{M}(i,j)$$

member confidence

$$m_i(k) = \frac{1}{N_k - 1\{e_i \in I_k\}} \sum_{\substack{j \in I_k \\ j \neq i}} \mathcal{M}(i,j)$$

Monti et al. Machine Learning:52,91-118 (2003)

Metrics to assess the efficacy of clustering

connectivity matrix

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items i and j belong to the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

consensus matrix

$$\mathcal{M}(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)}$$

cluster robustness

$$m(k) = \frac{1}{N_k(N_k-1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} \mathcal{M}(i,j)$$

member confidence

$$m_i(k) = \frac{1}{N_k - 1\{e_i \in I_k\}} \sum_{\substack{j \in I_k \\ j \neq i}} \mathcal{M}(i,j)$$

Monti et al. Machine Learning:52,91-118 (2003)

Metrics to assess the efficacy of clustering

connectivity matrix

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items i and j belong to the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

consensus matrix

$$\mathcal{M}(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)}$$

cluster robustness

$$m(k) = \frac{1}{N_k(N_k-1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} \mathcal{M}(i,j)$$

member confidence

$$m_i(k) = \frac{1}{N_k - 1\{e_i \in I_k\}} \sum_{\substack{j \in I_k \\ j \neq i}} \mathcal{M}(i,j)$$

Monti et al. Machine Learning:52,91-118 (2003)

Metrics to assess the efficacy of clustering

connectivity matrix

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items i and j belong to the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

consensus matrix

$$\mathcal{M}(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)}$$

cluster robustness

$$m(k) = \frac{1}{N_k(N_k - 1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} \mathcal{M}(i,j)$$

member confidence

$$m_i(k) = \frac{1}{N_k - 1\{e_i \in I_k\}} \sum_{\substack{j \in I_k \\ j \neq i}} \mathcal{M}(i,j)$$

Monti et al. Machine Learning:52,91-118 (2003)

clusterCons an R package for consensus clustering

- Collection of methods for performing consensus clustering in R
- Currently implemented for the major Bioconductor clustering methods :- agnes, pam, kmeans, hclust and diana. This is user extensible through simple generic wrapper template.
- Uses native command line arguments of existing clustering methods via a method wrapper
- Fully configurable analysis using any number of algorithms with user customised parameters
- Primary outputs are S4 class objects holding consensus matrices, cluster robustness matrices, and membership robustness matrices.
- S4 class slots hold a range of data and analysis objects for downstream applications e.g. plotting, cluster ouput and post-hoc matrix manipulation

clusterCons an R package for consensus clustering

- Collection of methods for performing consensus clustering in R
- Currently implemented for the major Bioconductor clustering methods :- agnes, pam, kmeans, hclust and diana. This is user extensible through simple generic wrapper template.
- Uses native command line arguments of existing clustering methods via a method wrapper
- Fully configurable analysis using any number of algorithms with user customised parameters
- Primary outputs are S4 class objects holding consensus matrices, cluster robustness matrices, and membership robustness matrices.
- S4 class slots hold a range of data and analysis objects for downstream applications e.g. plotting, cluster ouput and post-hoc matrix manipulation

clusterCons an R package for consensus clustering

- Collection of methods for performing consensus clustering in R
- Currently implemented for the major Bioconductor clustering methods :- agnes, pam, kmeans, hclust and diana. This is user extensible through simple generic wrapper template.
- Uses native command line arguments of existing clustering methods via a method wrapper
- Fully configurable analysis using any number of algorithms with user customised parameters
- Primary outputs are S4 class objects holding consensus matrices, cluster robustness matrices, and membership robustness matrices.
- S4 class slots hold a range of data and analysis objects for downstream applications e.g. plotting, cluster ouput and post-hoc matrix manipulation

clusterCons an R package for consensus clustering

- Collection of methods for performing consensus clustering in R
- Currently implemented for the major Bioconductor clustering methods :- agnes, pam, kmeans, hclust and diana. This is user extensible through simple generic wrapper template.
- Uses native command line arguments of existing clustering methods via a method wrapper
- Fully configurable analysis using any number of algorithms with user customised parameters
- Primary outputs are S4 class objects holding consensus matrices, cluster robustness matrices, and membership robustness matrices.
- S4 class slots hold a range of data and analysis objects for downstream applications e.g. plotting, cluster ouput and post-hoc matrix manipulation

clusterCons an R package for consensus clustering

- Collection of methods for performing consensus clustering in R
- Currently implemented for the major Bioconductor clustering methods :- agnes, pam, kmeans, hclust and diana. This is user extensible through simple generic wrapper template.
- Uses native command line arguments of existing clustering methods via a method wrapper
- Fully configurable analysis using any number of algorithms with user customised parameters
- Primary outputs are S4 class objects holding consensus matrices, cluster robustness matrices, and membership robustness matrices.
- S4 class slots hold a range of data and analysis objects for downstream applications e.g. plotting, cluster ouput and post-hoc matrix manipulation

clusterCons an R package for consensus clustering

- Collection of methods for performing consensus clustering in R
- Currently implemented for the major Bioconductor clustering methods :- agnes, pam, kmeans, hclust and diana. This is user extensible through simple generic wrapper template.
- Uses native command line arguments of existing clustering methods via a method wrapper
- Fully configurable analysis using any number of algorithms with user customised parameters
- Primary outputs are S4 class objects holding consensus matrices, cluster robustness matrices, and membership robustness matrices.
- S4 class slots hold a range of data and analysis objects for downstream applications e.g. plotting, cluster ouput and post-hoc matrix manipulation

An example analysis with clusterCons

Running the consensus clustering experiment

- the general resampling function cluscomp

```
cluscomp<-function(x,
        algorithms=list('kmeans'),
        alparams=list(),
        alweights=list(),
        clmin=2,clmax=10,
        prop=0.8,reps=50,merge=1)
```

- an example

```
cmr<-cluscomp(testdata,
        algorithms=c('kmeans','pam','agnes','hclust','diana'),merge=1,clmin=2,clmax=10,reps=500)
```

- returns a list of S4 class objects of class *consmatrix* and/or *mergematrix*

An example analysis with clusterCons

Running the consensus clustering experiment

- the general resampling function cluscomp

```
cluscomp<-function(x,
        algorithms=list('kmeans'),
        alparams=list(),
        alweights=list(),
        clmin=2,clmax=10,
        prop=0.8,reps=50,merge=1)
```

- an example

```
cmr<-cluscomp(testdata,
        algorithms=c('kmeans','pam','agnes','hclust','diana'),merge=1,clmin=2,clmax=10,reps=500)
```

- returns a list of S4 class objects of class *consmatrix* and/or *mergematrix*

An example analysis with clusterCons

Getting cluster robustness information

- the cluster robustness method cl_rob

```
cl_rob <- function(x,rm=data.frame())
```

- an example

```
cr<-cl_rob(cmr$kmeans_5)
```

	cluster	robustness
	1	0.6249620
	2	0.9988996
	3	0.6781015
	4	0.7681833
	5	0.9606562

An example analysis with clusterCons

Getting member robustness information

- the member robustness method mem_rob

```
mr <- mem_rob(current$cms$kmeans_5)
```

- an example

```
cluster2 <- mr$cluster2
```

	cluster	robustness
	1626527_at	0.9998077
	1630304_at	0.9998028
	1629886_s_at	0.9996142
	1623909_s_at	0.9996044
	1627000_s_at	0.9996006
	1633936_a_at	0.9994159
	1626485_at	0.9993952
	1624548_at	0.9993932
	1628125_at	0.9993893
	1638183_at	0.9993852
	1633512_at	0.9992331
	1623565_at	0.9992260
	1624393_at	0.9992013
	1637360_at	0.9992013
	1631281_a_at	0.9991935
	1636558_a_at	0.9991830
	1637708_a_at	0.9906468

An example analysis with clusterCons

Calculating the area under the curve

- If we re-sample using an iteration of cluster numbers we can look at the AUC to judge performance

  ac <- aucs(current$cms) - (auc shown just for algorithm 'agnes')

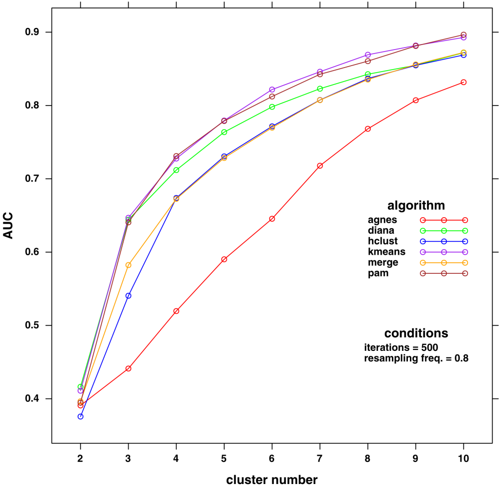
cluster	auc
2	0.3908623
3	0.4412078
4	0.5195906
5	0.5901873
6	0.6455020
7	0.7178445
8	0.7681852
9	0.8071388
10	0.8317600

- an example plot

  auc.plot(ac)

An example analysis with clusterCons

AUC versus cluster number for 5 algorithms and the merge



An example analysis with clusterCons

Calculating the change in the area under the curve

- Any peaks in the chane in the area under the curve represent local maxima for optimal cluster number

  dk <- deltak(current$cms) - (deltak shown just for algorithm agnes)

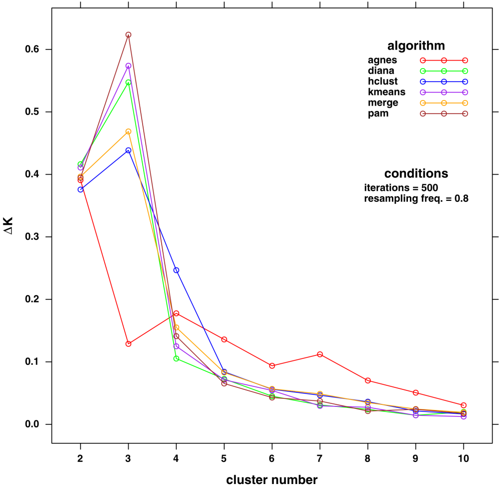
	cluster	Δ k
	2	0.39086234
	3	0.12880611
	4	0.17765514
	5	0.13586986
	6	0.09372386
	7	0.11207177
	8	0.07012760
	9	0.05070854
	10	0.03050431

- an example plot

  deltak.plot(dk)

An example analysis with clusterCons

Change in AUC (Δ k) versus cluster number for 5 algorithms and the merge



Live examples with clusterCons

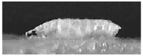
- Example1 - consensus clustering with simulated data by row and class
- Example2 - finding patient cancer sub-type by gene expression microarray clustering
- clusterCons - https://sourceforge.net/projects/clustercons/
- clusterCons - http://cran.r-project.org/web/packages/clusterCons/index.html

Anatomy of the Drosophila PNS - Sense organs

External sense organs

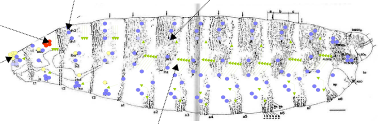
TOUCH
Mechanosensory bristles

TASTE
Chemosensory bristles

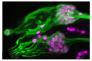


Drosophila larva

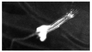
VISION
Photoreceptors



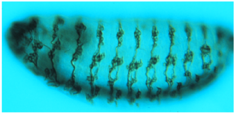
SMELL
Olfactory
sensilla



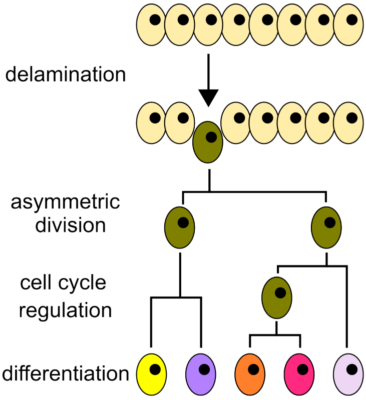
PROPRIOCEPTION and HEARING
Chordotonal stretch receptors



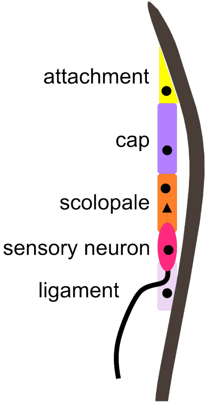
Drosophila embryo



Development of the Drosophila PNS



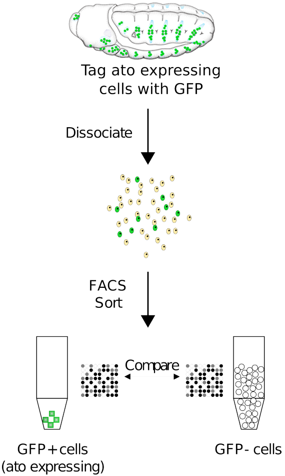
lineage diagram



chordotonal stretch receptor

RNA profiling cells expressing proneural genes throughout PNS development

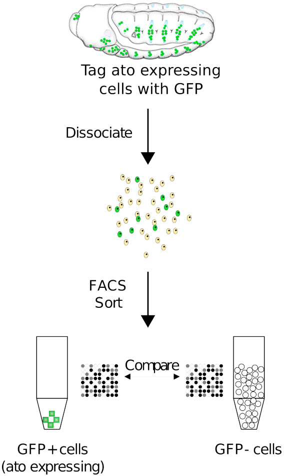
- transgenic flies are made that express GFP under the control of a proneural gene enhancer
- developmentally staged embryos are harvested and the cells dissociated
- cells are sorted by GFP fluorescence, RNA extracted and then hybridised to Affymetrix Dros2.0 microarray chips
- experiments performed for atonal, scute, amos and cato



Sebastian Cachero and Petra zur Lage

RNA profiling cells expressing proneural genes throughout PNS development

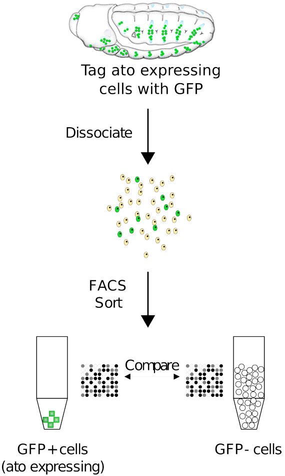
- transgenic flies are made that express GFP under the control of a proneural gene enhancer
- developmentally staged embryos are harvested and the cells dissociated
- cells are sorted by GFP fluorescence, RNA extracted and then hybridised to Affymetrix Dros2.0 microarray chips
- experiments performed for atonal, scute, amos and cato



Sebastian Cachero and Petra zur Lage

RNA profiling cells expressing proneural genes throughout PNS development

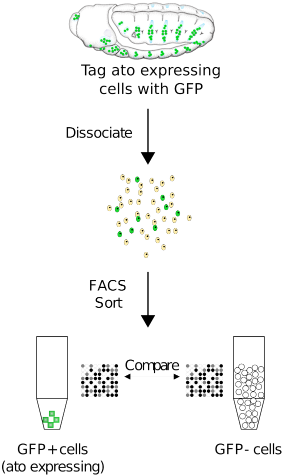
- transgenic flies are made that express GFP under the control of a proneural gene enhancer
- developmentally staged embryos are harvested and the cells dissociated
- cells are sorted by GFP fluorescence, RNA extracted and then hybridised to Affymetrix Dros2.0 microarray chips
- experiments performed for atonal, scute, amos and cato



Sebastian Cachero and Petra zur Lage

RNA profiling cells expressing proneural genes throughout PNS development

- transgenic flies are made that express GFP under the control of a proneural gene enhancer
- developmentally staged embryos are harvested and the cells dissociated
- cells are sorted by GFP fluorescence, RNA extracted and then hybridised to Affymetrix Dros2.0 microarray chips
- experiments performed for atonal, scute, amos and cato



Sebastian Cachero and Petra zur Lage

Identifying expression programmes and profiles

- expression programmes
  - analysis of genes enriched in proneural expressing cell types at each developmental time-point
  - candidate lists of network members
  - cis-regulatory motif analysis of candidate network members -> state based module discovery
- expression profiling (co-expression analysis)
  - grouping of genes with shared expression profiles - target discovery and local network assembly
  - cis-regulatory motif analysis - developmental module discovery
- module integration
  - intersection of state and developmental modules defines the global membership of the neurogenetic regulatory network
  - modules that are active at each stage can be separated from developmental modules
  - intersection of developmental modules with state based candidate lists reveals control switching

Identifying expression programmes and profiles

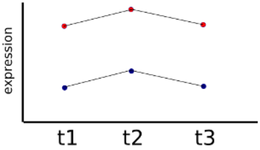
- expression programmes
  - analysis of genes enriched in proneural expressing cell types at each developmental time-point
  - candidate lists of network members
  - cis-regulatory motif analysis of candidate network members -> state based module discovery
- expression profiling (co-expression analysis)
  - grouping of genes with shared expression profiles - target discovery and local network assembly
  - cis-regulatory motif analysis - developmental module discovery
- module integration
  - intersection of state and developmental modules defines the global membership of the neurogenetic regulatory network
  - modules that are active at each stage can be separated from developmental modules
  - intersection of developmental modules with state based candidate lists reveals control switching

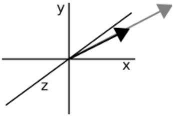
Identifying expression programmes and profiles

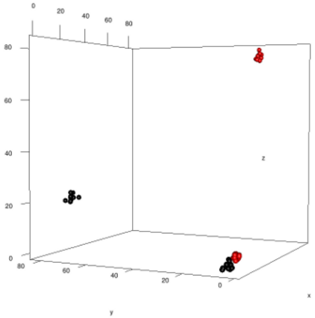
- expression programmes
  - analysis of genes enriched in proneural expressing cell types at each developmental time-point
  - candidate lists of network members
  - cis-regulatory motif analysis of candidate network members -> state based module discovery
- expression profiling (co-expression analysis)
  - grouping of genes with shared expression profiles - target discovery and local network assembly
  - cis-regulatory motif analysis - developmental module discovery
- module integration
  - intersection of state and developmental modules defines the global membership of the neurogenetic regulatory network
  - modules that are active at each stage can be separated from developmental modules
  - intersection of developmental modules with state based candidate lists reveals control switching

Grouping genes by expression measures

- grouping genes by expression is not the same as by profile
- genes sharing similar expression profiles need not cluster together

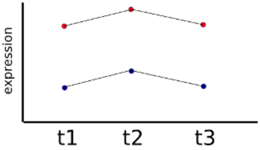


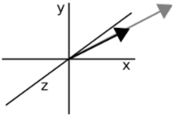


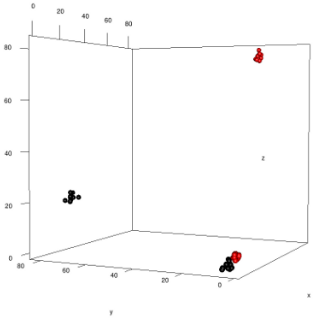


Grouping genes by expression measures

- grouping genes by expression is not the same as by profile
- genes sharing similar expression profiles need not cluster together

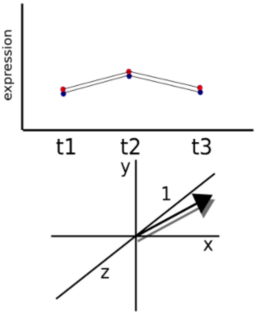


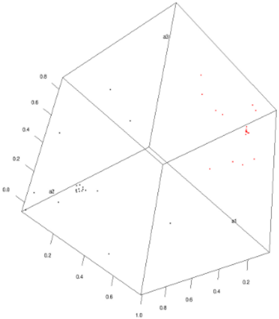




Grouping genes by expression profiles

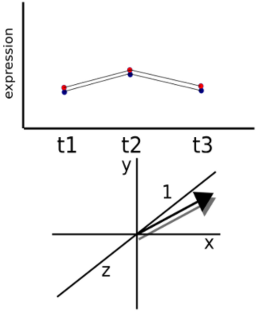
- using the same simulated data we can show expression profile groups by unitising the vector space
- genes sharing similar expression profiles now cluster together

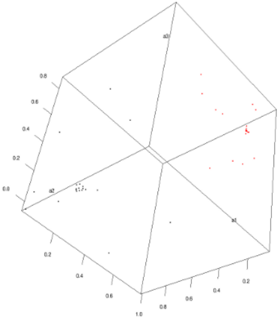




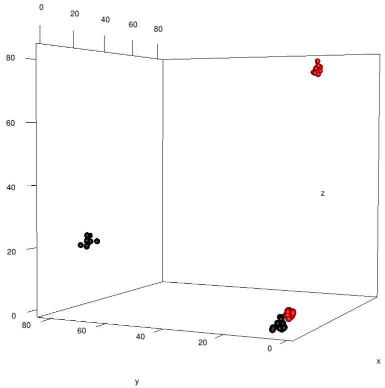
Grouping genes by expression profiles

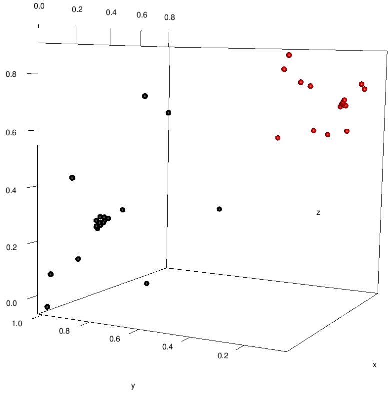
- using the same simulated data we can show expression profile groups by unitising the vector space
- genes sharing similar expression profiles now cluster together





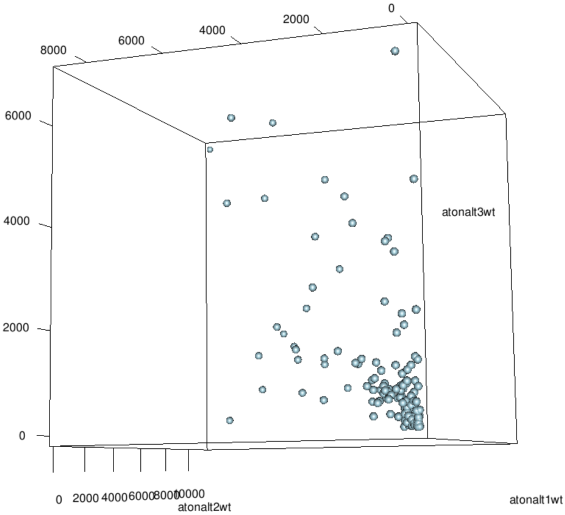
Before and After Unitisation





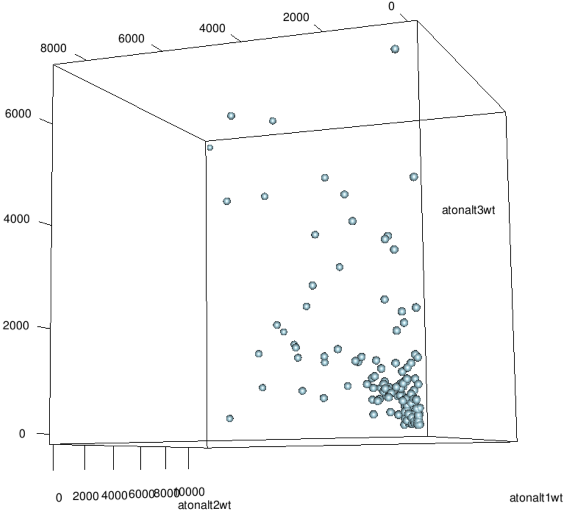
Following the expression of early atonal genes

- isolated genes that are enriched at atonal timepoint 1 (fold-change >=2, 1%FDR) - 159 genes
- followed their expression at wt t1, t2, t3 and at t1 in the atonal mutant
- before unitisation genes are mainly clustered around the origin



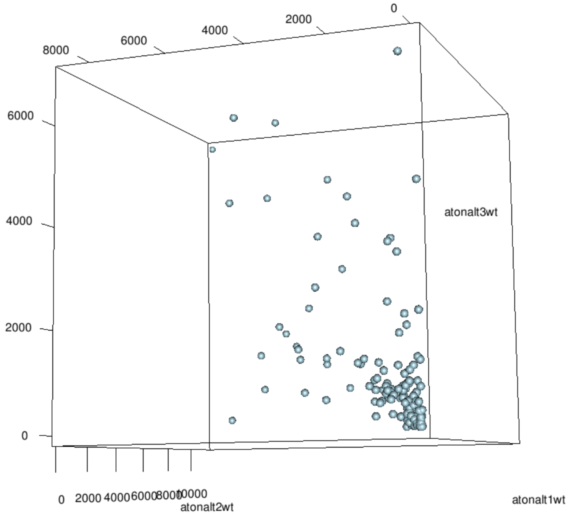
Following the expression of early atonal genes

- isolated genes that are enriched at atonal timepoint 1 (fold-change >=2, 1%FDR) - 159 genes
- followed their expression at wt t1, t2, t3 and at t1 in the atonal mutant
- before unitisation genes are mainly clustered around the origin



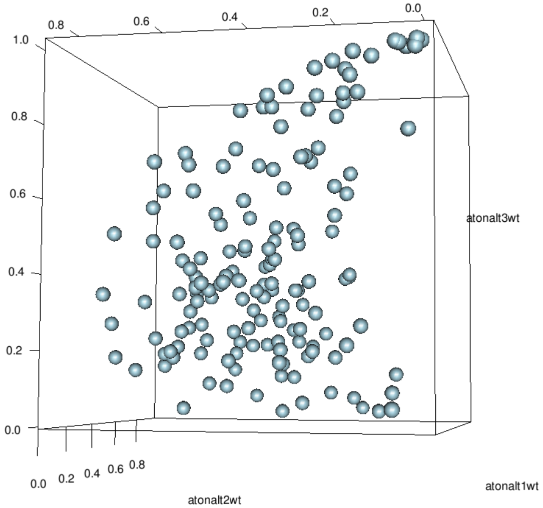
Following the expression of early atonal genes

- isolated genes that are enriched at atonal timepoint 1 (fold-change >=2, 1%FDR) - 159 genes
- followed their expression at wt t1, t2, t3 and at t1 in the atonal mutant
- before unitisation genes are mainly clustered around the origin



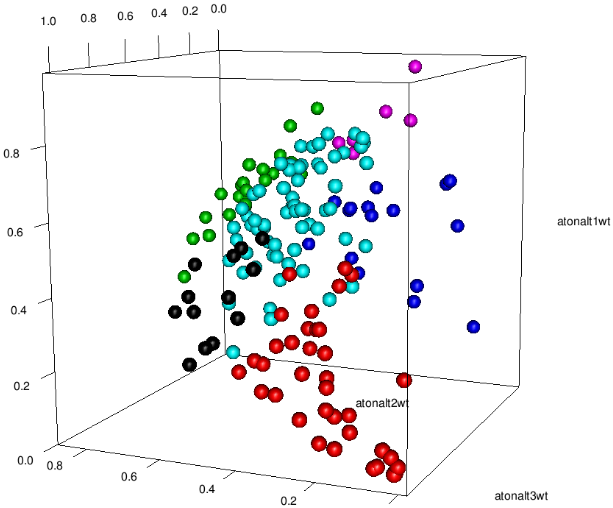
Following the expression of early atonal genes

- isolated genes that are enriched at atonal timepoint 1 (fold-change >=2, 1%FDR) - 159 genes
- followed their expression at wt t1, t2, t3 and at t1 in the atonal mutant
- after unitisation genes are distributed throughout the expression space



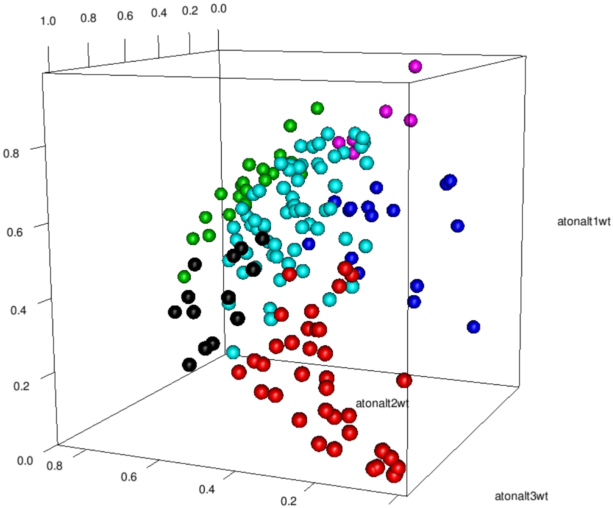
Following the expression of early atonal genes

- unitised expression data are now clustered
- this example uses an agglomerative hierarchical algorithm
- the plot is colour coded by cluster membership



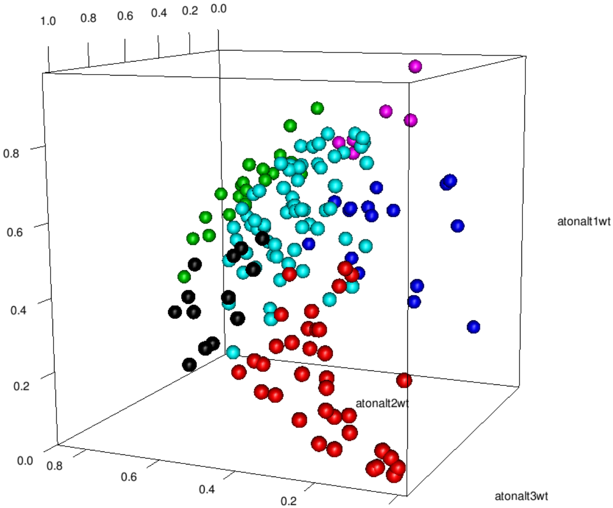
Following the expression of early atonal genes

- unitised expression data are now clustered
- this example uses an agglomerative hierarchical algorithm
- the plot is colour coded by cluster membership



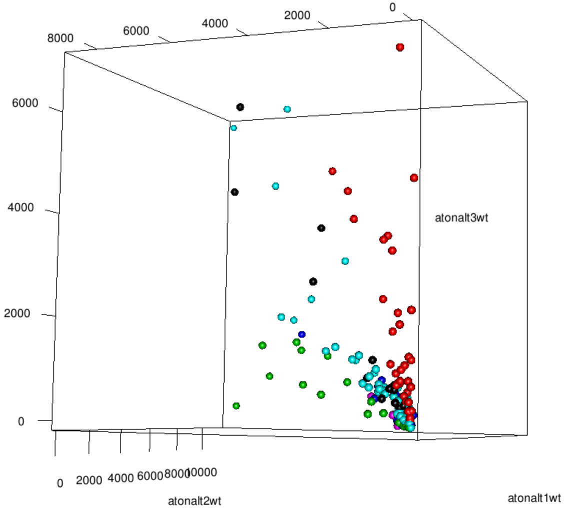
Following the expression of early atonal genes

- unitised expression data are now clustered
- this example uses an agglomerative hierarchical algorithm
- the plot is colour coded by cluster membership



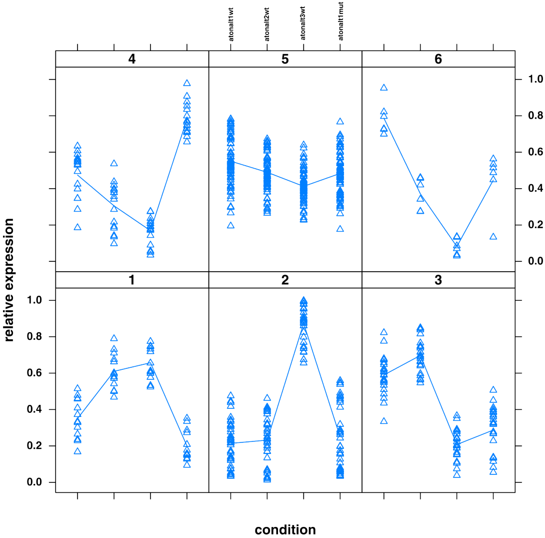
Following the expression of early atonal genes

- mapping the cluster membership colours onto the non-unitised expression data



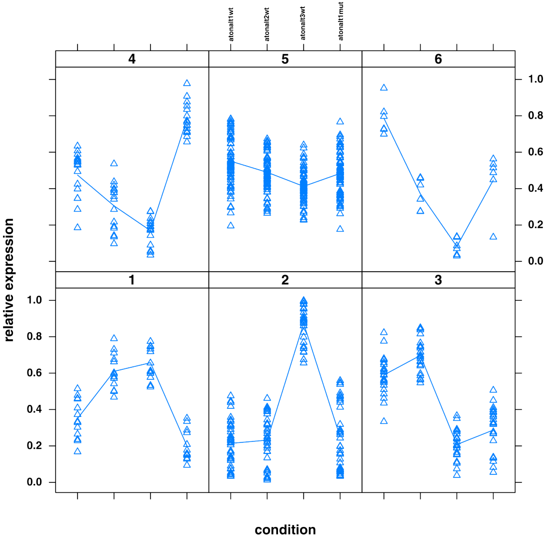
Following the expression of early atonal genes

- plot the actual unitised expression values atonal-GFP+ cells by cluster
- there are discrete expression profiles for these groups of genes
- profiles are broadly consistent with the categories we would expect to see



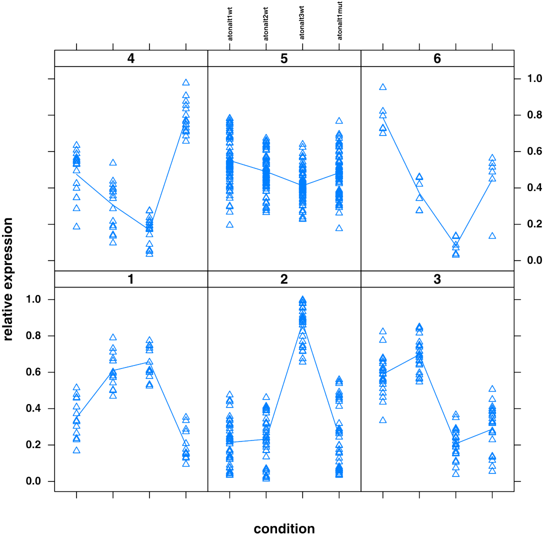
Following the expression of early atonal genes

- plot the actual unitised expression values atonal-GFP+ cells by cluster
- there are discrete expression profiles for these groups of genes
- profiles are broadly consistent with the categories we would expect to see



Following the expression of early atonal genes

- plot the actual unitised expression values atonal-GFP+ cells by cluster
- there are discrete expression profiles for these groups of genes
- profiles are broadly consistent with the categories we would expect to see



Following the expression of early atonal genes

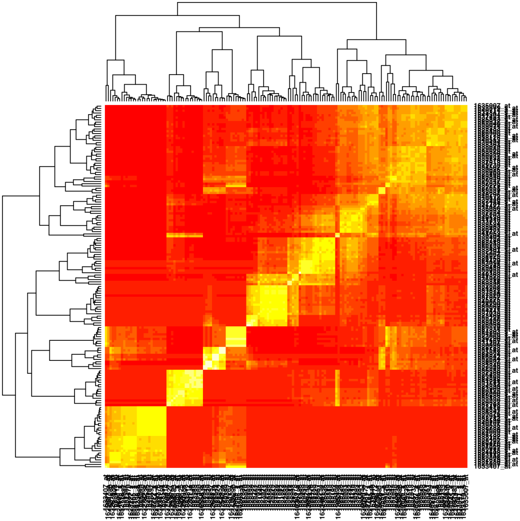
cluster membership

Cluster	Size
C_1	13
C_2	36
C_3	23
C_4	16
C_5	65
C_6	6

cluster 3

Sensory Organ Development GO:0007423 (p=6e-6)	
Gene name	
argos	ato
CG6330	CG31464
CG13653	nrm
unc	sca
rho	ImpL3
CG11671	CG7755
CG16815	CG15704
CG32150	knrl
CG32037	Toll-6
phyl	nvy
cato	

Heatmap of the consensus matrix



Ensemble clustering for early enriched atonal genes

Re-sampling using hclust, it=1000, rf=80%

cluster robustness

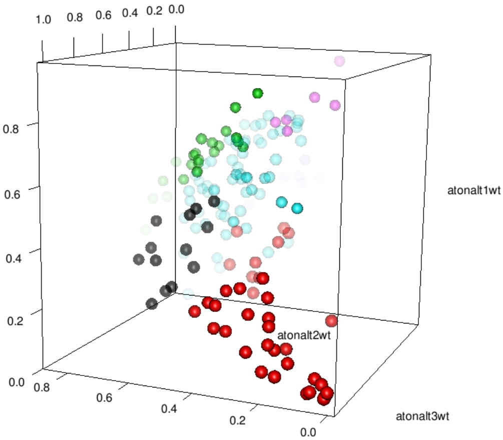
cluster	rob
1	0.4731433
2	0.7704514
3	0.7295124
4	0.7196309
5	0.7033960
6	0.6786388

membership robustness

cluster3			
affy_id	mem	affy_id	mem
1639896_at	0.68	1641578_at	0.56
1640363_a_at	0.54	1623314_at	0.53
1636998_at	0.49	1637035_at	0.36
1631443_at	0.35	1639062_at	0.31
1623977_at	0.31	1627520_at	0.3
1637824_at	0.28	1632882_at	0.27
1624262_at	0.26	1640868_at	0.26
1631872_at	0.26	1637057_at	0.24
1625275_at	0.24	1624790_at	0.22
1635227_at	0.08	1623462_at	0.07
1635462_at	0.03	1628430_at	0.03
1626059_at	0.02		

there are 8 out of 23 genes with <25% conservation in the cluster

Membership confidence mapped back onto unitised expression plots



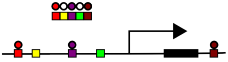
Application to the study of ciliogenesis

- Ciliated sensory neurons
  - Most sensory neurons have cilia at their dendritic tips
  - Cilia play crucial and highly conserved roles in motility, molecular transport and developmental processes such as left-right symmetry and sense organ development
  - Mutations in Rfx proteins are associated with defects in ciliogenesis in many organisms including Drosophila
- The X-box, comparative genetics and the ciliome
  - Rfx proteins bind to the X-box RYYNYYN[1-3]RRNRAC is bound by Rfx proteins
  - Genome screens for conserved X-boxes have recently been used to identify novel targets of Rfx proteins in Drosophila (Laurencon et al. Genome Biology(2007)**8**,R195)
  - Compared D.mel and D.pse common ancestor 40-60 mya
  - intron sequences 40% identical, known binding sites from the literature mapped on are 63% identical

Application to the study of ciliogenesis

- Ciliated sensory neurons
  - Most sensory neurons have cilia at their dendritic tips
  - Cilia play crucial and highly conserved roles in motility, molecular transport and developmental processes such as left-right symmetry and sense organ development
  - Mutations in Rfx proteins are associated with defects in ciliogenesis in many organisms including Drosophila
- The X-box, comparative genetics and the ciliome
  - Rfx proteins bind to the X-box RYYNYYN[1-3]RRNRAC is bound by Rfx proteins
  - Genome screens for conserved X-boxes have recently been used to identify novel targets of Rfx proteins in Drosophila (Laurencon et al. Genome Biology(2007)**8**,R195)
  - Compared D.mel and D.pse common ancestor 40-60 mya
  - intron sequences 40% identical, known binding sites from the literature mapped on are 63% identical

cis-regulatory modules (CRMs) an entry point for network assembly



Sites >=75% identical between D.mel and D.Pse that for genes that also contain an X-box (13/27) from the sensory cilium biogenesis cluster.

	Binding Factor								
Gene	X-box	Bcd	Abd-B	E-box	Ubx	Abd-A	Trl	Ftz	dl
ImpL3	■	■			■	■	■		
sca	■	■	■	■			■		■
Rfx	■	■	■	■				■	■
cato	■	■	■	■	■	■	■	■	■
CG8353	■	■	■	■	■	■	■	■	■
CG3769	■	■	■	■	■	■	■	■	■
CG1625	■	■	■	■	■	■	■	■	■
Oseg4	■	■	■	■	■	■		■	■
CG6486	■	■	■	■	■	■	■	■	■
CG14714	■	■	■	■	■	■	■		
CG32037	■	■	■	■	■	■	■	■	■
CG32150	■	■	■	■	■	■		■	
fd3f	■	■	■	■	■	■	■	■	■

- based on 75% conservation there are 7823 X-boxes in the fly genome (0.5/gene) so we expect 13 in list of 27
- sensory cluster has 50 conserved X-boxes an enrichment of x3.8

Summary

Summary

- The large variability in results from different clustering methodologies makes it difficult to be confident of clustering experiments performed in isolation
- Implementation of consensus clustering methodologies can allow the prioritisation of clusters allowing prioritisation of both groups and members of groups
- Unsupervised clustering methods have to be used in situations where the supervising data is sparse or of low quality (as is often the case with biological data).
- Clustering can reveal novel biological groupings in high order data and inform gene prioritisation efforts.

Summary

Summary

- The large variability in results from different clustering methodologies makes it difficult to be confident of clustering experiments performed in isolation
- Implementation of consensus clustering methodologies can allow the prioritisation of clusters allowing prioritisation of both groups and members of groups
- Unsupervised clustering methods have to be used in situations where the supervising data is sparse or of low quality (as is often the case with biological data).
- Clustering can reveal novel biological groupings in high order data and inform gene prioritisation efforts.

Summary

Summary

- The large variability in results from different clustering methodologies makes it difficult to be confident of clustering experiments performed in isolation
- Implementation of consensus clustering methodologies can allow the prioritisation of clusters allowing prioritisation of both groups and members of groups
- Unsupervised clustering methods have to be used in situations where the supervising data is sparse or of low quality (as is often the case with biological data).
- Clustering can reveal novel biological groupings in high order data and inform gene prioritisation efforts.

Summary

Summary

- The large variability in results from different clustering methodologies makes it difficult to be confident of clustering experiments performed in isolation
- Implementation of consensus clustering methodologies can allow the prioritisation of clusters allowing prioritisation of both groups and members of groups
- Unsupervised clustering methods have to be used in situations where the supervising data is sparse or of low quality (as is often the case with biological data).
- Clustering can reveal novel biological groupings in high order data and inform gene prioritisation efforts.

Summary

Summary

- The large variability in results from different clustering methodologies makes it difficult to be confident of clustering experiments performed in isolation
- Implementation of consensus clustering methodologies can allow the prioritisation of clusters allowing prioritisation of both groups and members of groups
- Unsupervised clustering methods have to be used in situations where the supervising data is sparse or of low quality (as is often the case with biological data).
- Clustering can reveal novel biological groupings in high order data and inform gene prioritisation efforts.