Improving a Multi-Objective Multipopulation Artificial Immune Network for Biclustering

Guilherme Palermo Coelho,Fabrício Olivetti de França and Fernando J. Von Żuben

DCA/FEEC/Unicamp – Brazil

2009 IEEE CEC – Trondheim, Norway – May 18th.-21st.
Outline

- Conceptual Aspects and Motivation;
- The improved multi-objective multipopulation approach for biclustering (MOM-aiNet+);
- Experimental results and discussion;
- Final comments.
Outline

- Conceptual Aspects and Motivation;

- The improved multi-objective multipopulation approach for biclustering (MOM-aiNet+);

- Experimental results and discussion;

- Final comments.
The process of finding correlated subsets of rows and columns of a given data matrix

(objects and attributes in rows and columns)
Conceptual Aspects & Motivation

Biclustering

- The correlation (or coherence) among elements in rows and columns of the data matrix can be evaluated according to different metrics:
  - Mean Squared Residue (Cheng & Church, 2000);

The correlation (or coherence) among elements in rows and columns of the data matrix can be evaluated according to different metrics:

- **Mean Squared Residue** (Cheng & Church, 2000);
Conceptual Aspects & Motivation

**Biclustering**

The correlation (or coherence) among elements in rows and columns of the data matrix can be evaluated according to different metrics:

- **Mean Squared Residue** (Cheng & Church, 2000);

In order to be functional and to allow a deeper analysis of the data, it is usually required that a bicluster presents a **large volume**;

![Biclustering Diagram]

\[
\text{Volume} = h \cdot w
\]
Conceptual Aspects & Motivation

Biclustering

Minimize the Mean-Squared Residue
Maximize the Volume

Multi-Objective Optimization Problem
Biclustering requires the extraction of several distinct biclusters from a single dataset.

Other aspects should also be considered: coverage, overlap and occupancy.
The aiNet algorithm (de Castro & Von Zuben, 2001):

- Originally proposed to solve clustering problems;
- Based on principles from the Natural Immune System;
- Evolves multiple populations in parallel;
- Stimulates the convergence of such subpopulations to distinct regions of the search space, preserving diversity.

In our previous work (Coelho et al., 2008):

- aiNet was modified to generate a pool of diverse biclusters, optimizing both the mean squared error and the volume of such biclusters.

**Conceptual Aspects & Motivation**

**The original MOM-aiNet algorithm**

**In our previous work (Coelho et al., 2008):**

- aiNet was modified to generate a pool of diverse biclusters, optimizing both the mean squared error and the volume of such biclusters.

**MOM-aiNet**

Multi-Objective Multipopulation Artificial Immune Network

Multiple populations of different biclusters:

- Different correlations extracted from the dataset
- Exploration of different regions of the data matrix;

Each sub-population corresponds to a possibly distinct “local set of non-dominated solutions”; 

All final sub-populations are returned (not only the final set of non-dominated solutions):

- Returns the maximum number of possible data correlation found;
Conceptual Aspects & Motivation

The original MOM-aiNet algorithm

Original Data Matrix

Set of Non-dominated Solutions

Mean-Squared Residue

Final subpopulation 1

Final subpopulation 2

Final subpopulation 3

Final subpopulation 4

1/(volume)

1/(volume)
Conceptual Aspects & Motivation

The original MOM-aiNet algorithm

Original Data Matrix

Set of Non-dominant Solutions

Mean-Squared Residue

1/(volume)

Final subpopulation 1

Final subpopulation 2

Final subpopulation 3

Final subpopulation 4

Different Data Correlation
Conceptual Aspects & Motivation

The original MOM-aiNet algorithm

Original Data Matrix

\[
\frac{1}{\text{volume}} \quad \text{Set of Non-dominated Solutions}
\]

Mean-Squared Residue
Conceptual Aspects & Motivation

The original MOM-aiNet algorithm

Original Data Matrix

1/(volume)

Set of Non-dominated Solutions

Mean-Squared Residue
Inclusion of mechanisms capable of increasing the average volume of the biclusters in the population without a significant impact on the mean squared residue;

These mechanisms were based on similar approaches already adopted in Cheng & Church’s CC algorithm (Cheng & Church, 2000):

- Multiple Node Removal;
- Single Node Removal;
- Multiple Node Addition.

Outline

- Conceptual Aspects and Motivation;
- The improved multi-objective multipopulation approach for biclustering (MOM-aiNet+);
- Experimental results and discussion;
- Final comments.
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

**While** ($\text{iteration} < \text{maximum number of iterations}$) **do**

**For** (each subpopulation $i$) **do**

1. Generate $N_c$ clones for each individual in the population;
2. Apply hypermutation to the generated clones;

*****For** (each unfeasible clone $j$) **do**

3. Make clone $j$ feasible;

*****End For**

4. Combine parents and mutated clones in a single subpopulation;
5. Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

**End For**

**If** ($\text{mod(\text{iteration}, N_{gs}}) == 0$) **then**

6. Determine the affinity among the largest individuals in each subpopulation;
7. **If** (two of these individuals present affinity higher than a given threshold $\sigma_s$) **then**

8. Merge the two subpopulations;

9. Select the non-dominated individuals (maximum $N_c$);

**End If**

9. Introduce new randomly generated sub-populations;

**End If**

*****If** ($\text{mod(\text{iteration}, N_{LS}}) == 0$) **then**

10. Perform multiple node insertion, to increase the volume of the biclusters;

*****End If**

**End While**
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;
2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

***3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;
5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If (mod($iteration$, $N_{gs}$) == 0) then

6 – Determine the affinity among the largest individuals in each subpopulation;
If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

7 – Merge the two subpopulations;
8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated sub-populations;

End If

***If (mod($iteration$, $N_{LS}$) == 0) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do
    For (each subpopulation $i$) do
        1 – Generate $N_c$ clones for each individual in the population;
        2 – Apply hypermutation to the generated clones;
        ***For (each unfeasible clone $j$) do
            3 – Make clone $j$ feasible;
        ***End For
        4 – Combine parents and mutated clones in a single subpopulation;
        5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;
    End For
    If (mod($iteration$, $N_{gs}$) == 0) then
        6 – Determine the affinity among the largest individuals in each subpopulation;
        If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then
            7 – Merge the two subpopulations;
            8 – Select the non-dominated individuals (maximum $N_c$);
        End If
        9 – Introduce new randomly generated sub-populations;
    End If
    ***If (mod($iteration$, $N_{LS}$) == 0) then
        10 – Perform multiple node insertion, to increase the volume of the biclusters;
    ***End If
End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;
2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do***

***3 – Make clone $j$ feasible;***

***End For***

4 – Combine parents and mutated clones in a single subpopulation;
5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If (mod($iteration$, $N_{gs}$) == 0) then

6 – Determine the affinity among the largest individuals in each subpopulation;
If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

7 – Merge the two subpopulations;
8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated sub-populations;

End If

***If (mod($iteration$, $N_{LS}$) == 0) then***

***10 – Perform multiple node insertion, to increase the volume of the biclusters;***

***End If***

End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;

2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

***3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;

5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

End If

***If ($\text{mod}(iteration, N_{LS}) == 0$) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($\text{iteration} < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;

2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

***3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;

5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

End If

***If (mod($\text{iteration}$, $N_{\text{LS}}$) == 0) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1. Generate $N_c$ clones for each individual in the population;
2. Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

3. Make clone $j$ feasible;

***End For

4. Combine parents and mutated clones in a single subpopulation;
5. Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

End If

***If $(\text{mod}(iteration, N_{LS}) == 0)$ then

***10. Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration <$ maximum number of iterations) do

For (each subpopulation $i$) do

1. Generate $N_c$ clones for each individual in the population;
2. Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

3. Make clone $j$ feasible;

***End For

4. Combine parents and mutated clones in a single subpopulation;
5. Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If (mod($iteration$, $N_{gs}$) == 0) then

6. Determine the affinity among the largest individuals in each subpopulation;
7. If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then
8. Merge the two subpopulations;
9. Select the non-dominated individuals (maximum $N_c$);

End If
9. Introduce new randomly generated sub-populations;

End If

***If (mod($iteration$, $N_{LS}$) == 0) then

10. Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
The MOM-aiNet+ Algorithm

Randomly create \( N \) initial subpopulations with a single individual;

While (\( \text{iteration} < \text{maximum number of iterations} \)) do

For (each subpopulation \( i \)) do

1 – Generate \( N_c \) clones for each individual in the population;

2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone \( j \)) do

***3 – Make clone \( j \) feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;

5 – Select all the non-dominated individuals (maximum \( N_c \)) to remain in the subpopulation;

End For

9 – Introduce new randomly generated sub-populations;

End If

***If (mod(\( \text{iteration} \), \( N_{LS} \)) == 0) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;

2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

***3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;

5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If ($\text{mod}(iteration, N_{gs}) == 0$) then

6 – Determine the affinity among the largest individuals in each subpopulation;

If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

7 – Merge the two subpopulations;

8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated subpopulations;

End If

***If ($\text{mod}(iteration, N_{LS}) == 0$) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
Randomly create $N$ initial subpopulations with a single individual;
While ($iteration < \text{maximum number of iterations}$) do
    For (each subpopulation $i$) do
        1 – Generate $N_c$ clones for each individual in the population;
        2 – Apply hypermutation to the generated clones;
        ***For (each unfeasible clone $j$) do
            3 – Make clone $j$ feasible;
        ***End For
        4 – Combine parents and mutated clones in a single subpopulation;
        5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;
    End For
    If (mod($iteration, N_{gs}$) == 0) then
        6 – Determine the affinity among the largest individuals in each subpopulation;
        If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then
            7 – Merge the two subpopulations;
            8 – Select the non-dominated individuals (maximum $N_c$);
        End If
        9 – Introduce new randomly generated sub-populations;
    End If
    ***If (mod($iteration, N_{LS}$) == 0) then
        10 – Perform multiple node insertion, to increase the volume of the biclusters;
    ***End If
End While

The MOM-aiNet+ Algorithm
Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;
2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;
5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If ($\text{mod}(iteration, \ N_{gs}) == 0$) then

6 – Determine the affinity among the largest individuals in each subpopulation;

If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

7 – Merge the two subpopulations;
8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated sub-populations;

End If

***If ($\text{mod}(iteration, \ N_{LS}) == 0$) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While

The MOM-aiNet+ Algorithm

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;
2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;
5 – Select all the non-dominated individuals (maximum $N_c$);

End For

If ($\text{mod}(iteration, \ N_{gs}) == 0$) then

6 – Determine the affinity among the largest individuals in each subpopulation;

If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

7 – Merge the two subpopulations;
8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated sub-populations;

End If

***If ($\text{mod}(iteration, \ N_{LS}) == 0$) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
Randomly create $N$ initial subpopulations with a single individual;

**While** $(iteration < \text{maximum number of iterations})$ **do**

For (each subpopulation $i$) **do**

1. Generate $N_c$ clones for each individual in the population;
2. Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do***

3. Make clone $j$ feasible;

***End For***

4. Combine parents and mutated clones in a single subpopulation;
5. Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If $(\text{mod}(iteration, N_{gs}) == 0)$ **then**

6. Determine the affinity among the largest individuals in each subpopulation;

   If (two of these individuals present affinity higher than a given threshold $\sigma_s$) **then**

   7. Merge the two subpopulations;
   8. Select the non-dominated individuals (maximum $N_c$);

End If

9. Introduce new randomly generated subpopulations;

End If

***If (mod(iteration, $N_{LS}$) == 0) then***

***10. Perform multiple node insertion, to increase the volume of the biclusters;***

***End If***

End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do
  For (each subpopulation $i$) do
    1 – Generate $N_c$ clones for each individual in the population;
    2 – Apply hypermutation to the generated clones;
    ***For (each unfeasible clone $j$) do
      3 – Make clone $j$ feasible;
    ***End For
    4 – Combine parents and mutated clones in a single subpopulation;
    5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;
  End For
  If (mod($iteration$, $N_{gs}$) == 0) then
    6 – Determine the affinity among the largest individuals in each subpopulation;
    7 – Merge the two subpopulations;
    8 – Select the non-dominated individuals (maximum $N_c$);
  End If
  9 – Introduce new randomly generated sub-populations;
End While

***If (mod($iteration$, $N_{LS}$) == 0) then
  ***10 – Perform multiple node insertion, to increase the volume of the biclusters;
***End If
Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do
  For (each subpopulation $i$) do
    1 – Generate $N_c$ clones for each individual in the population;
    2 – Apply hypermutation to the generated clones;
    ***For (each unfeasible clone $j$) do
      3 – Make clone $j$ feasible;
    ***End For
    4 – Combine parents and mutated clones in a single subpopulation;
    5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;
  End For
  If (mod($iteration, N_{gs}) == 0$) then
    6 – Determine the affinity among the largest individuals in each subpopulation;
    If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then
      7 – Merge the two subpopulations;
      8 – Select the non-dominated individuals (maximum $N_c$);
    End If
    9 – Introduce new randomly generated sub-populations;
  End If
  ***If (mod($iteration, N_{LS}) == 0$) then
    ***10 – Perform multiple node insertion, to increase the volume of the biclusters;
  ***End If
End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;
2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do ***

3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;
5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If ($\text{mod}(iteration, N_{gs}) == 0$) then

6 – Determine the affinity among the largest individuals in each subpopulation;

   If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

   7 – Merge the two subpopulations;

   8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated sub-populations;

End If

End While

***If ($\text{mod}(iteration, N_{LS}) == 0$) then

***10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

   For (each subpopulation $i$) do

      1 – Generate $N_c$ clones for each individual in the population;
      2 – Apply hypermutation to the generated clones;

      ***For (each unfeasible clone $j$) do
      ***   3 – Make clone $j$ feasible;
      ***End For

      4 – Combine parents and mutated clones in a single subpopulation;
      5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

   End For

   If ($\text{mod}(iteration, N_{gs}) == 0$) then

      6 – Determine the affinity among the largest individuals in each subpopulation;
      If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

         7 – Merge the two subpopulations;
         8 – Select the non-dominated individuals (maximum $N_c$);

      End If

      9 – Introduce new randomly generated sub-populations;

   End If

   ***If ($\text{mod}(iteration, N_{LS}) == 0$) then

      ***10 – Perform multiple node insertion, to increase the volume of the biclusters;

   ***End If

End While
The MOM-aiNet+ Algorithm

Randomly create \( N \) initial subpopulations with a single individual;

\[ \text{While (iteration < maximum number of iterations) do} \]

\[ \text{For (each subpopulation \( i \)) do} \]

\[ 1 \text{ – Generate } N_c \text{ clones for each individual in the population;} \]

\[ 2 \text{ – Apply hypermutation to the generated clones;} \]

\[ \text{***For (each unfeasible clone } j \text{) do} \]

\[ 3 \text{ – Make clone } j \text{ feasible;} \]

\[ \text{***End For} \]

\[ 4 \text{ – Combine parents and mutated clones in a single subpopulation;} \]

\[ 5 \text{ – Select all the non-dominated individuals (maximum } N_c \text{) to remain in the subpopulation;} \]

\[ \text{End For} \]

\[ \text{If (mod(iteration, } N_{LS} \text{) == 0) then} \]

\[ 6 \text{ – Determine the affinity among the largest individuals in each subpopulation;} \]

\[ \text{If (two of these individuals present affinity higher than a given threshold } \sigma_s \text{) then} \]

\[ 7 \text{ – Merge the two subpopulations;} \]

\[ 8 \text{ – Select the non-dominated individuals (maximum } N_c \text{);} \]

\[ \text{End If} \]

\[ 9 \text{ – Introduce new randomly generated sub-populations;} \]

\[ \text{End If} \]

\[ \text{***If (mod(iteration, } N_{LS} \text{) == 0) then} \]

\[ 10 \text{ – Perform multiple node insertion, to increase the volume of the biclusters;} \]

\[ \text{***End If} \]

\[ \text{End While} \]
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

  For (each subpopulation $i$) do
    1 – Generate $N_c$ clones for each individual in the population;
    2 – Apply hypermutation to the generated clones;
    ***For (each unfeasible clone $j$) do
      3 – Make clone $j$ feasible;
    ***End For
    4 – Combine parents and mutated clones in a single subpopulation;
    5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;
  End For

  If ($\text{mod}(iteration, N_{gs}) == 0$) then
    6 – Determine the affinity among the largest individuals in each subpopulation;
    If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then
      7 – Merge the two subpopulations;
      8 – Select the non-dominated individuals (maximum $N_c$);
    End If
    9 – Introduce new randomly generated sub-populations;
  End If

  ***If ($\text{mod}(iteration, N_{LS}) == 0$) then
    10 – Perform multiple node insertion, to increase the volume of the biclusters;
  ***End If
End While
The MOM-aiNet+ Algorithm

Randomly create $N$ initial subpopulations with a single individual;

While ($iteration < \text{maximum number of iterations}$) do

For (each subpopulation $i$) do

1 – Generate $N_c$ clones for each individual in the population;
2 – Apply hypermutation to the generated clones;

***For (each unfeasible clone $j$) do

3 – Make clone $j$ feasible;

***End For

4 – Combine parents and mutated clones in a single subpopulation;
5 – Select all the non-dominated individuals (maximum $N_c$) to remain in the subpopulation;

End For

If ($\text{mod}(iteration, N_{gs}) == 0$) then

6 – Determine the affinity among the largest individuals in each subpopulation;
If (two of these individuals present affinity higher than a given threshold $\sigma_s$) then

7 – Merge the two subpopulations;
8 – Select the non-dominated individuals (maximum $N_c$);

End If

9 – Introduce new randomly generated sub-populations;

End If

***If ($\text{mod}(iteration, N_{LS}) == 0$) then

10 – Perform multiple node insertion, to increase the volume of the biclusters;

***End If

End While
The MOM-aiNet+ Algorithm

Trying to make unfeasible clones feasible (residue reduction)

- **Multiple Node Removal:**
  - Removes all rows and columns of the bicluster that present a mean squared residue larger than $\alpha.H(I,J)$.

- **Single Node Removal:**
  - It is applied when the Multiple Node Removal is not capable of reducing the mean squared residue of the bicluster so that it attends the upper constraint;
  - Iteratively removes those rows and columns of the bicluster that have the largest contribution to the overall mean squared residue.
Multiple Node Addition:

- It is performed at fixed intervals ($N_{LS}$ iterations);
- It is an attempt to increase the volume of each bicluster in the population without increasing their mean squared residue;
- Inserts all rows and columns, not yet present in the bicluster, that do not increase the overall mean squared residue of such bicluster.
Outline

- Conceptual Aspects and Motivation;
- The improved multi-objective multipopulation approach for biclustering (MOM-aiNet+);
- Experimental results and discussion;
- Final comments.
Experimental Results

Datasets

- **Saccharomyces cerevisiae (Yeast):**
  - 2,884 genes;
  - 17 conditions;

- **Human B-cell Lymphoma (Human):**
  - 4,026 genes;
  - 96 conditions;

- **Movielens:**
  - 80,000 ratings given by 943 users to 1,650 movies;
  - Highly sparse dataset – approx 6.4% of occupancy.
Experimental Results

Algorithms

- **CC (Cheng & Church’s algorithm);**
  

- **BIC-aiNet (Artificial Immune Network for Biclustering);**
  

- **Mitra & Banka’s algorithm;**
  

- **MOM-aiNet;**
### Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSR</th>
<th>Volume</th>
<th>Coverage (%)</th>
<th>Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yeast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOM-aiNet</td>
<td>294.36 ± 1.53</td>
<td>1834.70 ± 30.70</td>
<td>85.91 ± 1.06</td>
<td>7.27 ± 0.26</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>298.71 ± 0.05</td>
<td>2335.90 ± 13.30</td>
<td>61.05 ± 0.48</td>
<td>13.57 ± 0.07</td>
</tr>
<tr>
<td><strong>Human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOM-aiNet</td>
<td>1097.50 ± 18.24</td>
<td>2953.00 ± 126.62</td>
<td>41.65 ± 0.76</td>
<td>2.87 ± 0.21</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1198.10 ± 0.11</td>
<td>11055.00 ± 288.88</td>
<td>57.07 ± 0.36</td>
<td>18.09 ± 0.50</td>
</tr>
<tr>
<td><strong>Movielens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOM-aiNet</td>
<td>1.61 ± 0.03</td>
<td>557.14 ± 22.55</td>
<td>1.58 ± 0.03</td>
<td>4.39 ± 0.33</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1.96 ± 0.01</td>
<td>619.92 ± 16.63</td>
<td>1.12 ± 0.03</td>
<td>5.44 ± 0.22</td>
</tr>
</tbody>
</table>

- MOM-aiNet+ was capable of increasing the average volume of the bicluster, with a low impact on the mean squared residue;
- However, **Coverage** (except for *Human*) and **Overlap** were worsened;
### Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSR</th>
<th>Volume</th>
<th>Coverage (%)</th>
<th>Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yeast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>144.96 ± 2.90</td>
<td>126.10 ± 0.20</td>
<td>76.75 ± 0.12</td>
<td>10.67 ± 0.01</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>298.71 ± 0.05</td>
<td>2335.90 ± 13.30</td>
<td>61.05 ± 0.48</td>
<td>13.57 ± 0.07</td>
</tr>
<tr>
<td><strong>Human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>915.50 ± 6.27</td>
<td>610.00 ± 3.61</td>
<td>31.57 ± 0.19</td>
<td>0.00 ± 0.00</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1198.10 ± 0.11</td>
<td>11055.00 ± 288.88</td>
<td>57.07 ± 0.36</td>
<td>18.09 ± 0.50</td>
</tr>
<tr>
<td><strong>Movielens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>0.51 ± 0.37</td>
<td>6.72 ± 8.29</td>
<td>0.04 ± 0.06</td>
<td>2.97 ± 1.03</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1.96 ± 0.01</td>
<td>619.92 ± 16.63</td>
<td>1.12 ± 0.03</td>
<td>5.44 ± 0.22</td>
</tr>
</tbody>
</table>

- CC explicitly minimizes the MSR, without considering the volume of the biclusters;
- Its mechanisms also tend to reduce coverage (not observed in *Yeast*) and overlap.
Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSR</th>
<th>Volume</th>
<th>Coverage (%)</th>
<th>Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>144.96 ± 2.90</td>
<td>126.10 ± 0.20</td>
<td>76.75 ± 0.12</td>
<td>10.67 ± 0.01</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>298.71 ± 0.05</td>
<td>2335.90 ± 13.30</td>
<td>61.05 ± 0.48</td>
<td>13.57 ± 0.07</td>
</tr>
<tr>
<td>CC</td>
<td>915.50 ± 6.27</td>
<td>610.00 ± 3.61</td>
<td>31.57 ± 0.19</td>
<td>0.00 ± 0.00</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1198.10 ± 0.11</td>
<td>11055.00 ± 288.88</td>
<td>57.07 ± 0.36</td>
<td>18.09 ± 0.50</td>
</tr>
<tr>
<td>CC</td>
<td>0.51 ± 0.37</td>
<td>6.72 ± 8.29</td>
<td>0.04 ± 0.06</td>
<td>2.97 ± 1.03</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1.96 ± 0.01</td>
<td>619.92 ± 16.63</td>
<td>1.12 ± 0.03</td>
<td>5.44 ± 0.22</td>
</tr>
</tbody>
</table>

- MOM-aiNet+ obtained sets of much larger biclusters, respecting the upper residue constraint;
### Experimental Results

#### MOM-aiNet+ vs. BIC-aiNet

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSR</th>
<th>Volume</th>
<th>Coverage (%)</th>
<th>Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yeast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>232.04 ± 25.30</td>
<td>3659.00 ± 1214.80</td>
<td>68.20 ± 11.28</td>
<td>20.67 ± 7.58</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>298.71 ± 0.05</td>
<td>2335.90 ± 13.30</td>
<td>61.05 ± 0.48</td>
<td>13.57 ± 0.07</td>
</tr>
<tr>
<td><strong>Human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>1156.70 ± 2.62</td>
<td>11606.00 ± 38.39</td>
<td>23.08 ± 0.11</td>
<td>32.00 ± 0.21</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1198.10 ± 0.11</td>
<td>11055.00 ± 288.88</td>
<td>57.07 ± 0.36</td>
<td>18.09 ± 0.50</td>
</tr>
<tr>
<td><strong>Movielens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>1.33 ± 0.03</td>
<td>216.43 ± 2.05</td>
<td>0.62 ± 0.01</td>
<td>2.80 ± 0.14</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1.96 ± 0.01</td>
<td>619.92 ± 16.63</td>
<td>1.12 ± 0.03</td>
<td>5.44 ± 0.22</td>
</tr>
</tbody>
</table>

- BIC-aiNet is better in *Yeast*: better MSR, volume and coverage;
- In *Human*, volume and MSR are similar, but MOM-aiNet+ achieved better coverage and overlap;
Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSR</th>
<th>Volume</th>
<th>Coverage (%)</th>
<th>Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yeast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>232.04 ± 25.30</td>
<td>3659.00 ± 1214.80</td>
<td>68.20 ± 11.28</td>
<td>20.67 ± 7.58</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>298.71 ± 0.05</td>
<td>2335.90 ± 13.30</td>
<td>61.05 ± 0.48</td>
<td>13.57 ± 0.07</td>
</tr>
<tr>
<td><strong>Human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>1156.70 ± 2.62</td>
<td>11606.00 ± 38.39</td>
<td>23.08 ± 0.11</td>
<td>32.00 ± 0.21</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1198.10 ± 0.11</td>
<td>11055.00 ± 288.88</td>
<td>57.07 ± 0.36</td>
<td>18.09 ± 0.50</td>
</tr>
<tr>
<td><strong>Movielens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>1.33 ± 0.03</td>
<td>216.43 ± 2.05</td>
<td>0.62 ± 0.01</td>
<td>2.80 ± 0.14</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1.96 ± 0.01</td>
<td>619.92 ± 16.63</td>
<td>1.12 ± 0.03</td>
<td>5.44 ± 0.22</td>
</tr>
</tbody>
</table>

- In *Movielens*, MOM-aiNet+ respected the upper residue constraint and presented a much higher volume and coverage, although the overlap was also higher.
## Experimental Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSR</th>
<th>Volume</th>
<th>Coverage (%)</th>
<th>Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Yeast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mitra &amp; Banka</td>
<td>288.62 ± 3.35</td>
<td>4800.40 ± 76.40</td>
<td>83.26 ± 0.83</td>
<td>20.76 ± 0.72</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>298.71 ± 0.05</td>
<td>2335.90 ± 13.30</td>
<td>61.05 ± 0.48</td>
<td>13.57 ± 0.07</td>
</tr>
<tr>
<td><strong>Human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mitra &amp; Banka</td>
<td>1052.90 ± 2.33</td>
<td>7559.00 ± 83.41</td>
<td>17.08 ± 0.20</td>
<td>34.57 ± 0.27</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1198.10 ± 0.11</td>
<td>11055.00 ± 288.88</td>
<td>57.07 ± 0.36</td>
<td>18.09 ± 0.50</td>
</tr>
<tr>
<td><strong>Movielens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mitra &amp; Banka</td>
<td>1.18 ± 0.38</td>
<td>73.42 ± 10.73</td>
<td>0.28 ± 0.07</td>
<td>4.63 ± 2.10</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>1.96 ± 0.01</td>
<td>619.92 ± 16.63</td>
<td>1.12 ± 0.03</td>
<td>5.44 ± 0.22</td>
</tr>
</tbody>
</table>

- Mitra & Banka’s algorithm is better in *Yeast*: it presented better MSR, volume and coverage;

- MOM-aiNet+ was better in the other two problems, with much higher volume and coverage, and a better overlap in *Human*;
Experimental Results

*MovieLens Dataset – Occupancy*

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Occupancy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>100.00 ± 0.00</td>
</tr>
<tr>
<td>BIC-aiNet</td>
<td>92.31 ± 0.22</td>
</tr>
<tr>
<td>Mitra &amp; Banka</td>
<td>96.54 ± 88.84</td>
</tr>
<tr>
<td>MOM-aiNet</td>
<td>88.84 ± 0.16</td>
</tr>
<tr>
<td>MOM-aiNet+</td>
<td>87.56 ± 0.13</td>
</tr>
</tbody>
</table>

- The average occupancy achieved by MOM-aiNet+ for this problem was the smallest, but close-to-100% occupancy is not desired in collaborative filtering applications;
- It is necessary to predict missing values, so that recommendations can be made.
Outline

- Conceptual Aspects and Motivation;

- The improved multi-objective multipopulation approach for biclustering (MOM-aiNet+);

- Experimental results and discussion;

- Final comments.
When compared to MOM-aiNet, MOM-aiNet+ was able to improve the average volume of the final population of biclusters for the three problems studied;

The impact of such volume increase on the mean square residue was small (the upper constraint imposed was always respected);

However, this improvement in volume led to a reduction in coverage and an increase in the overlap among biclusters, specially for the Yeast and Human problems.

As a future work, we intend to perform additional experiments to better understand how the modifications proposed are correlated to the observed changes in coverage and overlap.
Improving a Multi-Objective Multipopulation Artificial Immune Network for Biclustering

Presentation available at:

http://sites.google.com/site/guilhermecoelho/my-files/

This research was sponsored by UOL (www.uol.com.br), through its “UOL Bolsa Pesquisa” (UOL Research Grant) program (proc. num. 20080129215000), CAPES and CNPq.

Guilherme Palermo Coelho: gcoelho@dca.fee.unicamp.br
Fabrício Olivetti de França: olivetti@dca.fee.unicamp.br
Fernando J. Von Zuben: vonzuben@dca.fee.unicamp.br