Présentation du projet : Data-Camp

Classification automatique de cellules par analyse RNA-seq

METIN Samuel, TREMBLAY Marion et BERTRAND Flavie

Professeur : **Arthur Leroy**

12 janvier 2026

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Le problème

Le problème

- Classification automatique de cellules par analyse RNA-seq
- 13551 variables/gènes
- 1500 observations/cellules
- Variable cible : Type cellulaire (Cancer_cells, NK_cells, T_cells_CD4+, T_cells_CD8+)
- 1000 cellules d'entraînement / 500 cellules de test

1 Présentation du problème

2 Analyse des données
- Analyse des variables cibles
- Description des potentiels problèmes

3 Pre-processing

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

1 Présentation du problème

2 Analyse des données
- Analyse des variables cibles
- Description des potentiels problèmes

3 Pre-processing

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Analyse des variables cibles

4 types cellulaires

- Cancer_cells
- NK_cells
- T_cells_CD4+
- T_cells_CD8+

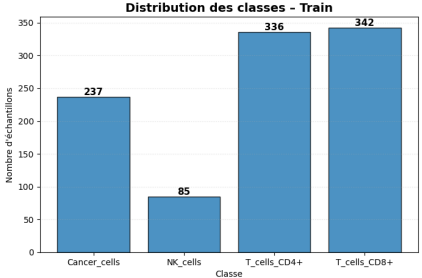

Figure 1: Distribution des variables cibles -train

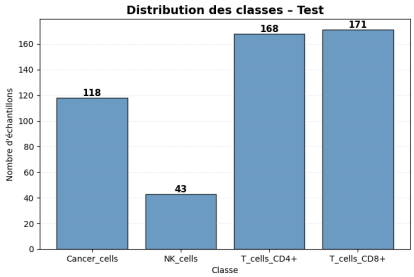

Figure 2: Distribution des variables cibles -test

1 Présentation du problème

2 Analyse des données
- Analyse des variables cibles
- Description des potentiels problèmes

3 Pre-processing

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Variables constantes

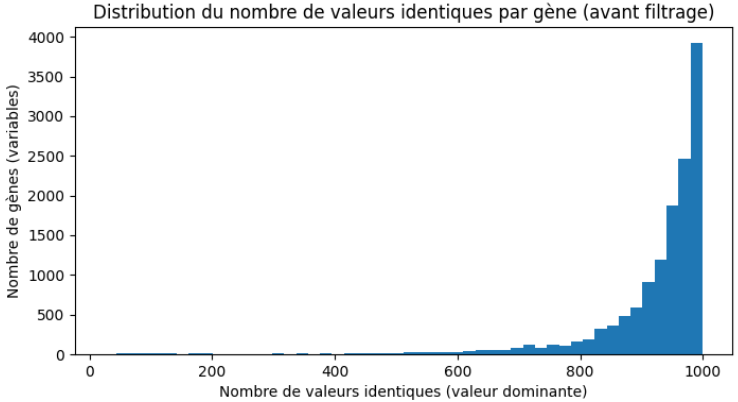

Figure 3: Nombre de valeurs identiques par variables

Sans équilibrage

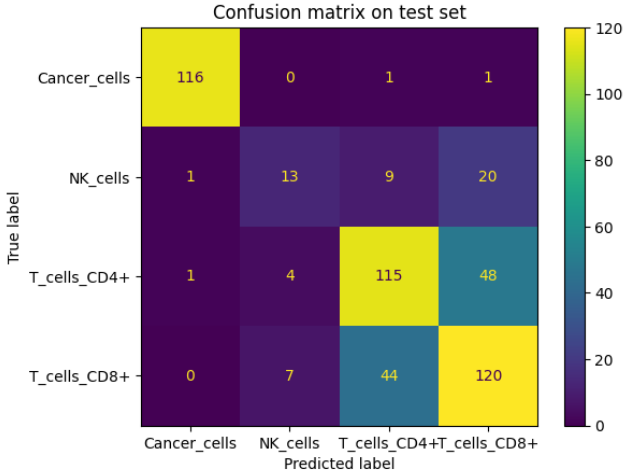

Figure 4: Matrice de confusion pour gradient boosting au début

Défis

Défis

- Haute dimensionnalité (13550 gènes pour 1000 observations)
- Déséquilibre des classes
- Confusion inter-classes
- Bruit

1 Présentation du problème

2 Analyse des données

3 Pre-processing
- Normalisation et transformation
- Sélection des gènes informatifs

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

1 Présentation du problème

2 Analyse des données

3 Pre-processing
- Normalisation et transformation
- Sélection des gènes informatifs

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Normalisation CPM + log1p

Actions de pre-processing

1. **Normalisation CPM** (Counts Per Million)
   - Corrige les différences de profondeur de séquençage
   - $X_{norm} = \frac{X}{total\_counts} \times 10000$
2. **Transformation log1p**
   - Stabilise la variance
   - Réduit l'impact des valeurs extrêmes
   - $X_{final} = \log(1 + X_{norm})$

1 Présentation du problème

2 Analyse des données

3 Pre-processing
- Normalisation et transformation
- Sélection des gènes informatifs

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Sélection des gènes hautement variables (HVG)

Deux stratégies selon l'algorithme

1. Pour Gradient Boosting : Variance brute

- 2000 gènes sélectionnés
- Sélection par variance directe (sans normalisation)

→ Capture plus de features pour un modèle plus complexe

2. Pour Régression Logistique : Coefficient de Variation (CV)

- 800 gènes sélectionnés
- CV = $\frac{\text{variance}}{\text{moyenne}}$ (filtrage si moyenne < 0.1)

→ Normalise le biais vers les gènes très exprimés

→ Privilégie les gènes avec forte variation *relative*

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification
- Réduction de dimension : PCA
- Gestion du déséquilibre
- Architecture Hiérarchique

5 Résultat des algorithmes utilisés

6 Conclusion

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification
- Réduction de dimension : PCA
- Gestion du déséquilibre
- Architecture Hiérarchique

5 Résultat des algorithmes utilisés

6 Conclusion

Comparaison avec et sans PCA

Comparaison

- 25 composantes expliquent seulement 20% de variance
- Pourtant, la PCA améliore les performances

Méthode	**Train Acc**	**Test Acc**	**Gap**
Sans PCA	100%	81%	19%
PCA (25 comp.)	88%	85%	3%

Table 1: Comparaison avec/sans pca

Détermination automatique du nombre optimal de composantes

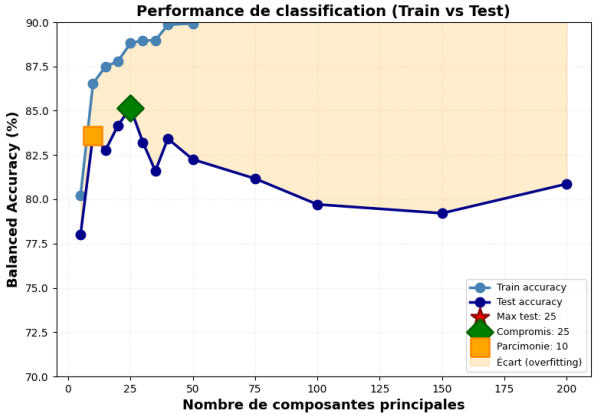

Figure 5: Compromis entre la performance et la généralisation

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification
- Réduction de dimension : PCA
- Gestion du déséquilibre
- Architecture Hiérarchique

5 Résultat des algorithmes utilisés

6 Conclusion

Gestion du déséquilibre des classes

Stratégies différentes selon l'algorithme

1. **Régression Logistique : class_weight='balanced'**
   - Pondération automatique
   - Utilise toutes les données d'entraînement

2. **Gradient Boosting : Équilibrage strict**
   - Limite toutes les classes au minimum (NK_cells)
   - NK_cells mieux prédites
   - Perte d'information (réduction du dataset)
   - Nécessaire car GB n'a pas de class_weight natif

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification
- Réduction de dimension : PCA
- Gestion du déséquilibre
- Architecture Hiérarchique

5 Résultat des algorithmes utilisés

6 Conclusion

Architecture hiérarchique

Motivation

T_cells_CD4+ et T_cells_CD8+ génétiquement très proches
→ Difficile à différencier

Niveau 1 : Classification principale

- Cancer_cells
- NK_cells
- **T_cells** (CD4+ et CD8+ regroupées)

Niveau 2 : Spécialisation T_cells

- T_cells_CD4+
- T_cells_CD8+

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Comparaison : Gradient Boosting vs Régression Logistique

Algorithme	Train Acc	Test Acc	Gap
Gradient Boosting	96%	**86%**	10%
Régression Logistique	88%	85%	**3%**

Table 2: Performances des modèles

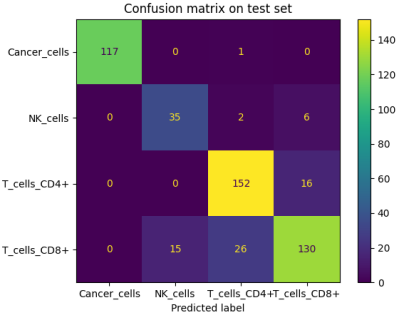

Figure 6: Matrice de confusion Régression Logistique

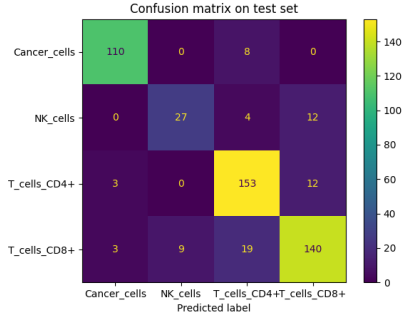

Figure 7: Matrice de confusions Gradient Boosting

1 Présentation du problème

2 Analyse des données

3 Pre-processing

4 Optimisation de la classification

5 Résultat des algorithmes utilisés

6 Conclusion

Conclusion

- Problèmes : données sparses, 1 cible sous représentée, 2 cibles proches
- Stratégies : capter les informations en réduisant la dimension, équilibrer les classes, hiérarchiser la classification en la décomposant
- Résultats : balanced accuracy satisfaisante (pour les 2 meilleurs classifieurs testés) sur un jeu de test inconnu