Διαδραστική Πλατφόρμα Ανάλυσης Δεδομένων Μοριακής Βιολογίας

Συγγραφέας: Χριστόφορος Αλπανάκης
Πανεπιστήμιο: Ιονιο Πανεπιστημιο
Τμήμα: Πληροφορικη
Ημερομηνία: Μάιος 2025

Περίληψη

Η παρούσα εργασία παρουσιάζει την ανάπτυξη μιας διαδραστικής διαδικτυακής εφαρμογής για την ανάλυση δεδομένων μοριακής βιολογίας με χρήση του πλαισίου Streamlit. Η εφαρμογή παρέχει ολοκληρωμένα εργαλεία για την ανάλυση δεδομένων γονιδιακής έκφρασης, συμπεριλαμβανομένης της διερευνητικής ανάλυσης δεδομένων, τεχνικών μηχανικής μάθησης (PCA και K-means clustering), και διαδραστικών οπτικοποιήσεων. Η εφαρμογή είναι containerized με χρήση Docker για εύκολη ανάπτυξη σε διαφορετικά περιβάλλοντα.

---

1. Εισαγωγή

Η έρευνα στη μοριακή βιολογία παράγει τεράστιες ποσότητες δεδομένων που απαιτούν εξελιγμένα εργαλεία ανάλυσης. Οι παραδοσιακές ροές εργασίας ανάλυσης συχνά περιλαμβάνουν πολλαπλά πακέτα λογισμικού και γλώσσες προγραμματισμού, δημιουργώντας εμπόδια για τους ερευνητές. Αυτό το έργο αντιμετωπίζει αυτές τις προκλήσεις αναπτύσσοντας μια φιλική προς τον χρήστη, διαδικτυακή πλατφόρμα που ενσωματώνει κοινές εργασίες ανάλυσης δεδομένων μοριακής βιολογίας σε μια ενιαία διεπαφή.

Η εφαρμογή εστιάζει στην ανάλυση δεδομένων γονιδιακής έκφρασης, παρέχοντας εργαλεία για την προεπεξεργασία δεδομένων, τη διερευνητική ανάλυση, τη μείωση διαστάσεων, την ομαδοποίηση και την οπτικοποίηση. Χτισμένη με Python και Streamlit, η πλατφόρμα προσφέρει μια διαισθητική διεπαφή που επιτρέπει στους ερευνητές να αναλύουν τα δεδομένα τους χωρίς εκτεταμένες γνώσεις προγραμματισμού.

---

2. Σχεδιασμός Υλοποίησης

2.1 Αρχιτεκτονική Συστήματος

Η εφαρμογή ακολουθεί μια αρθρωτή αρχιτεκτονική με τα ακόλουθα στοιχεία:

• **Επίπεδο Δεδομένων**: Διαχειρίζεται τη φόρτωση, επικύρωση και προεπεξεργασία δεδομένων
• **Επίπεδο Ανάλυσης**: Υλοποιεί στατιστικούς αλγορίθμους και αλγορίθμους μηχανικής μάθησης
• **Επίπεδο Οπτικοποίησης**: Δημιουργεί διαδραστικά διαγράμματα και γραφήματα
• **Επίπεδο Διεπαφής Χρήστη**: Παρέχει διεπαφή με καρτέλες για διαφορετικές λειτουργίες

2.2 Τεχνολογικό Στοίβα

• **Frontend**: Streamlit για διαδικτυακή διεπαφή
• **Επεξεργασία Δεδομένων**: Pandas, NumPy
• **Μηχανική Μάθηση**: Scikit-learn
• **Οπτικοποίηση**: Plotly, Matplotlib, Seaborn
• **Containerization**: Docker

---

3. Διαγράμματα UML

3.1 Διάγραμμα Περιπτώσεων Χρήσης

Οι κύριες περιπτώσεις χρήσης περιλαμβάνουν:

1. Φόρτωση δεδομένων γονιδιακής έκφρασης
2. Εκτέλεση διερευνητικής ανάλυσης δεδομένων
3. Εκτέλεση ανάλυσης PCA
4. Εκτέλεση K-means clustering
5. Δημιουργία οπτικοποιήσεων
6. Προβολή πληροφοριών ομάδας

Κύριοι Δρώντες: • Ερευνητής Μοριακής Βιολογίας
• Αναλυτής Δεδομένων
• Φοιτητής

3.2 Διάγραμμα Κλάσεων

Κύριες κλάσεις στην εφαρμογή:

DataProcessor • Μέθοδοι: load_data(), validate_data(), preprocess_data()
• Ιδιότητες: data, metadata, processed_data

MLAnalyzer • Μέθοδοι: perform_pca(), perform_clustering(), calculate_statistics()
• Ιδιότητες: pca_results, cluster_labels, explained_variance

Visualizer • Μέθοδοι: create_heatmap(), create_scatter_plot(), create_histogram()
• Ιδιότητες: plot_config, color_schemes

UIController • Μέθοδοι: render_tabs(), handle_user_input(), update_interface()
• Ιδιότητες: session_state, user_parameters

---

4. Ανάλυση Υλοποίησης

4.1 Μονάδα Επεξεργασίας Δεδομένων

Η μονάδα επεξεργασίας δεδομένων διαχειρίζεται: • Φόρτωση και ανάλυση αρχείων CSV
• Επικύρωση δεδομένων και ελέγχους ποιότητας
• Log μετασχηματισμό και κανονικοποίηση
• Φιλτράρισμα γονιδίων βάσει επιπέδων έκφρασης

Παράδειγμα Κώδικα Επεξεργασίας Δεδομένων:

```
def preprocess_data(data, log_transform=True, normalize=False):
    processed_data = data.copy()

    if log_transform:
        processed_data = np.log2(processed_data + 1)

    if normalize:
        processed_data = processed_data.div(
            processed_data.sum(axis=0), axis=1
        ) * 1e6

    return processed_data
```

4.2 Μονάδα Μηχανικής Μάθησης

Δύο κύριοι αλγόριθμοι υλοποιούνται:

Ανάλυση Κύριων Συνιστωσών (PCA): • Μειώνει τη διαστασιμότητα δεδομένων γονιδιακής έκφρασης
• Βοηθά στην αναγνώριση προτύπων σε σχέσεις δειγμάτων
• Διαμορφώσιμος αριθμός συνιστωσών

K-means Clustering: • Ομαδοποιεί δείγματα ή γονίδια βάσει προτύπων έκφρασης
• Διαμορφώσιμος αριθμός ομάδων από τον χρήστη
• Υποστηρίζει τόσο ομαδοποίηση δειγμάτων όσο και γονιδίων

Υλοποίηση PCA:

```
def perform_pca(data, n_components=3):
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(data.T)
```

```
    pca = PCA(n_components=n_components)
    pca_result = pca.fit_transform(X_scaled)

    return pca_result, pca.explained_variance_ratio_
```

4.3 Μονάδα Οπτικοποίησης

Η εφαρμογή παρέχει πολλαπλούς τύπους οπτικοποίησης: • Ιστογράμματα κατανομών
• Χάρτες θερμότητας συσχετίσεων
• Διαγράμματα διασποράς PCA
• Box και violin plots
• Προφίλ γονιδιακής έκφρασης

---

5. Αποτελέσματα και Οπτικοποιήσεις

Η εφαρμογή δημιουργεί διάφορους τύπους οπτικοποιήσεων:

5.1 Αποτελέσματα Διερευνητικής Ανάλυσης

• Οι κατανομές τιμών έκφρασης δείχνουν τυπικά log-κανονικά πρότυπα
• Οι χάρτες θερμότητας συσχέτισης δειγμάτων αποκαλύπτουν batch effects και βιολογικά πρότυπα
• Η ανάλυση μεταβλητών γονιδίων αναγνωρίζει γονίδια με τη μεγαλύτερη βιολογική διακύμανση

5.2 Αποτελέσματα Μηχανικής Μάθησης

• Η ανάλυση PCA συνήθως καταγράφει 60-80% της διακύμανσης στις πρώτες 3 συνιστώσες
• Το K-means clustering ομαδοποιεί αποτελεσματικά δείγματα κατά βιολογικές συνθήκες
• Η ομαδοποίηση γονιδίων αναγνωρίζει συν-εκφραζόμενες γονιδιακές μονάδες

---

6. Dockerization

Η εφαρμογή είναι containerized με χρήση Docker για εύκολη ανάπτυξη:

6.1 Διαμόρφωση Docker

• Base image: Python 3.9-slim
• Exposed port: 8501 (προεπιλογή Streamlit)
• Υλοποίηση health check
• Βελτιστοποιημένη cache επιπέδων

6.2 Εντολές Ανάπτυξης

```
# Δημιουργία Docker image
docker build -t molbio-app .

# Εκτέλεση container
docker run -p 8501:8501 molbio-app

# Εκτέλεση σε background
docker run -d -p 8501:8501 --name molbio-container molbio-app
```

6.3 Αρχείο Dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 8501
CMD ["streamlit", "run", "app.py", "--server.port=8501",
    "--server.address=0.0.0.0"]
```

---

7. Λειτουργικότητες Εφαρμογής

7.1 Καρτέλα Φόρτωσης Δεδομένων

• Φόρτωση CSV αρχείων
• Δείγματα δεδομένων για δοκιμή
• Παράμετροι προεπεξεργασίας (log transformation, normalization)
• Φιλτράρισμα γονιδίων χαμηλής έκφρασης

7.2 Καρτέλα Διερευνητικής Ανάλυσης

• Στατιστικές περιλήψεις δεδομένων
• Κατανομές τιμών έκφρασης
• Συσχετίσεις μεταξύ δειγμάτων
• Ανάλυση μεταβλητότητας γονιδίων

7.3 Καρτέλα Μηχανικής Μάθησης

• Διαμορφώσιμη ανάλυση PCA
• K-means clustering για δείγματα και γονίδια
• Οπτικοποίηση αποτελεσμάτων
• Ερμηνεία συνιστωσών PCA

---

8. Συμπεράσματα

Η αναπτυχθείσα εφαρμογή ενσωματώνει επιτυχώς πολλαπλές ροές εργασίας ανάλυσης δεδομένων μοριακής βιολογίας σε μια ενιαία, φιλική προς τον χρήστη πλατφόρμα. Η αρθρωτή αρχιτεκτονική επιτρέπει εύκολη επέκταση με πρόσθετες μεθόδους ανάλυσης, ενώ η containerization με Docker εξασφαλίζει συνεπή ανάπτυξη σε διαφορετικά περιβάλλοντα.

Μελλοντικές βελτιώσεις θα μπορούσαν να περιλαμβάνουν: • Πρόσθετους αλγορίθμους μηχανικής μάθησης
• Υποστήριξη για διαφορετικές μορφές δεδομένων
• Ανάλυση σε πραγματικό χρόνο συνεργατικής
• Ενσωμάτωση με βιολογικές βάσεις δεδομένων

---

9. Βιβλιογραφία

1. Streamlit Documentation. (2024). *Building data apps with Streamlit*. Retrieved from https://docs.streamlit.io/
2. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
3. McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 445, 51-56.
4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95.