

## تنفيذ خوارزميات المعلوماتية الحيوية لدراسة تسلسل البروتين باستخدام الحوسبة التفرعية ومنصات الحوسبة السحابية

د.م. حسن محمد البستاني\*

م. سنان غازي شاش\*\*

(تاريخ الإيداع 2023/7/11 . قبل للنشر في 2023/9/13 )

□ ملخص □

أي تغيير يطرأ على سلاسل البروتين أو المادة الوراثية ينتج عنه مرض معين، هذه الأمراض قد تكون بغاية الخطورة والتأخر في الكشف عنها قد يؤدي بحياة المرضى، ونظراً للحاجة الماسة لمعرفة التغيرات التي تطرأ على هذه السلاسل والكشف عنها بشكل مبكر، تم تطبيق خوارزميات المعلوماتية الحيوية المصممة للعمل على البيئة التسلسلية لدراسة سلاسل البروتين ودراسة تسلسل الشيفرة ضمن DNA. وبما أن البيانات التي تتعامل مع المعطيات الحيوية ضخمة جداً، لذلك فإن تنفيذها على الحاسب التسلسلي يشكل مشكلة كبيرة من حيث الزمن اللازم للمعالجة. استخدمنا في هذا البحث خوارزمية تسلسلية من خوارزميات المعلوماتية الحيوية للكشف عن سلاسل البروتين، وقمنا بتطويرها إلى خوارزمية تفرعية وطبقناها على البيئة التفرعية GPU وهي متاحة عبر الحوسبة السحابية Google Colab ، ومن ثم قارنا النتائج من حيث الدقة وزمن التنفيذ وحجم العتاد الصلب مع الخوارزمية التسلسلية، وذلك لإيجاد أفضل طريقة لمعرفة التغيير الحاصل على السلاسل النظامية بدقة عالية وزمن قصير وكلفة أقل. كلمات مفتاحية: المعلوماتية الحيوية، الأحماض الأمينية، الأسس الأزوتية، الخلية، الجينوم، البروتين، سلاسل DNA، NCBI، BLAST، Google Colab، معالج GPU .

\*: مدرس في كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس  
\*\*: طالب ماجستير في قسم هندسة النظم الحاسوبية والالكترونية، كلية هندسة تكنولوجيا المعلومات والاتصالات، جامعة طرطوس .

## Implementation of bioinformatics algorithms to study protein sequences using parallel computing and cloud computing platforms

**Dr-Eng. Hasan Albustani\***  
**Eng. Sinan Shash\*\***

(Received 11/7/2023 . Accepted 13/9/2023)

### □ ABSTRACT

Any change that occurs in the protein chains or genetic material results in a specific disease. These diseases may be extremely dangerous, and delay in detecting them may lead to the death of patients. Given the urgent need to know the changes that occur in these chains and detect them early, bioinformatics algorithms designed To work on the sequencing environment to study protein sequences and study the code sequences within DNA. Since the data that deals with vital data is very large, Therefore, implementing it on a serial computer poses a major problem in terms of processing time.

In this research, we used a serial algorithm from bioinformatics algorithms to detect protein chains, and developed it into a parallel algorithm and applied it to the GPU parallel environment that is available via Google Colab cloud computing, and then compared the results in terms of accuracy, execution time, and hardware size with the serial algorithm. To find the best way to find out the change in regular series with high accuracy, short time and lower cost..

**Keywords:** bioinformatics, amino acids, nitrogenous bases, cell, genome, protein, DNA sequences, NCBI, BLAST, Google Colab , Processor GPU.

---

\* Assistant Professor, Faculty of Communication and Information Engineering Technology – Tartous University.

\*\* Master Student at CESE Department, Faculty of Communication and Information Engineering Technology, Tartous University.

## 1. مقدمة

شهدت تسعينات القرن العشرين تقدماً كبيراً في علوم الوراثة والجينات، خصوصاً في المشاريع الخاصة بالتعرف على التركيب الوراثي لعدد من الكائنات الحية، مثل الخميرة والفأر وذبابة الفاكهة وغيرها. وتوجت هذه الجهود بالكشف عن التركيب الوراثية (الجينية) للإنسان [5]، ومن الواضح أن تلك الإنجازات شكلت كمية كبيرة من المعرفة والمعلومات المتطورة في مجال علم البيولوجيا، تضمنت معرفة الخريطة الجينية لكثير من الكائنات المختلفة، والمركبات البروتينية الناتجة من الجينات المختلفة والتفاعل بينها وما ينتج ذلك التفاعل من بروتينات أساسية وأنزيمات وغيرها.

وتعود بداية استخدام علم المعلوماتية الحيوية إلى العالمة "مارجريت داي هوف" في عام 1968 حيث قامت هذه العالمة بإنشاء خريطة للتتابعات البروتينية توضح فيه تركيب وتتابع مجموعة من البروتينات الفيروسية، وقد ساعد ذلك على إنشاء برنامج حوسبي للبحث عن مناطق التشابه والاختلاف بين التتابعات الجينية والبروتينية المختلفة وتعد هذه الخطوة من أهم الخطوات البارزة في تطور علم المعلوماتية الحيوية. حيث استخدم العلماء في هذه الدراسة أول برنامج بحثي للكشف عن تشابه التتابعات والذي أطلق عليه اسم FASTP وبذلك تم توضيح أن تتابع الجينوم الفيروسي المسبب لسرطان V-sis مشابه جداً لتتابع الجين PDGF الخلوي، وزودت هذه النتيجة علماء البيولوجيا بمعلومات كثيرة عن كيفية تسبب هذا التتابع الفيروسي في إحداث السرطان [5].

ويعد مشروع الجينوم البشري من أكثر المشاريع العلمية التي استخدمت علم المعلوماتية الحيوية بكثافة، ويزود علم المعلوماتية الحيوية العلماء بالبرامج البسيطة والميسرة لتحليل البيانات الضخمة والتي تعتبر الآن وسيلة أساسية لحل الكثير من المشاكل العلمية الجديدة، كما ويعتبر علم المعلوماتية الحيوية علم حيوي مثير ويعبر عن طريقة حديثة للتفكير في المشاكل البيولوجية والتي سوف يقود تطور الحاسب الآلي فيها اكتشافات بيولوجية كبيرة، فقد أصبحت برامج الحاسب الآلي وقواعد البيانات الخاصة بالمعلوماتية الحيوية جزء لا يتجزأ من العمل البيولوجي الحديث لكل من يعمل في هذا المجال في القرن الحادي والعشرين. ويقوم هذا العلم على حل المشاكل البيولوجية الهامة والتي تشمل فهم الارتباط بين التركيب الجيني والشكل المظهري للأمراض البشرية، وكذلك فهم العلاقة بين تراكيب ووظائف البروتينات المختلفة.



الشكل (1) شكل توضيحي يبين استخدام الحاسب في تحليل المعلومات البيولوجية.

قام الباحثون في المرجع [2]، بإجراء دراسة تم فيها نقل البرنامج الأساسي للتنبؤ بالتفاعل بين البروتين والبروتين (protein-protein docking), MEGADOCK ، إلى Microsoft Azure كمثال على بيئة سحابة HPC. تم إنشاء بيئة الحوسبة السحابية المتوازية مع ما يصل إلى 1600 نواة لوحدة المعالجة المركزية و 960 وحدة معالجة رسومات باستخدام أربعة أنواع متماثلة من وحدة المعالجة المركزية ونوعين من وحدة معالجة الرسومات، ثم تقييم أداء الحوسبة المتوازية.

قام الباحثون في المرجع [3]، بإجراء دراسة اقترحوا فيها العديد من الأساليب والأدوات التي يمكن تقسيمها إلى ثلاثة أنواع: تخزين البيانات الضخمة وتصميم الخوارزميات الفعالة والحوسبة المتوازية. الغرض من هذه الدراسة هو التحقيق في تقنيات البرمجة المتوازية الشائعة لمعالجة تسلسل الجينوم. فقد تم تقديم ثلاثة نماذج مشتركة للحوسبة المتوازية وفقاً لبنى الأجهزة الخاصة بهم، تم تصنيف كل منها إلى نوعين أو ثلاثة أنواع وقد تم تحليلها بشكل أكبر مع ميزاتهما. بعد ذلك، تم مناقشة الحوسبة المتوازية لمعالجة تسلسل الجينوم من خلال أربعة تطبيقات شائعة: محاذاة تسلسل الجينوم واستدعاء تعدد الأشكال للنكليوتيدات الأحادية و المعالجة المسبقة لتسلسل الجينوم و اكتشاف النمط والبحث، وقد تم مناقشة القيود والاتجاهات المستقبلية لتقنيات الحوسبة المتوازية.

قام الباحثون في المراجع [4]-[14]، باستعراض استخدام منصة Hadoop في تطبيقات المعلوماتية الحيوية الهيكلية. حيث يمثل الـ Hadoop إطاراً جديداً ضمن الحوسبة الموزعة لتحليل كميات كبيرة من بيانات البروتين الذي يعد مفتاحاً لدراسة الإنتاجية العالية لرسو البروتين الترابطي، وتجميع مجتمعات البروتينات الترابطية والمحاذاة الهيكلية. حيث تم استعراض عدداً من التطبيقات باستخدام الـ Hadoop للتحليلات عالية الإنتاجية وقابليتها للتوسع، بالإضافة إلى بعض تطبيقات المعلوماتية الحيوية التي تساعد على فهم مبادئ تنفيذ الخوارزميات باستخدام نمط البرمجة MapReduce.

## 2. أهمية البحث وأهدافه

تبرز أهمية البحث في الدراسة التفصيلية لخوارزميات المعلوماتية الحيوية واستخدام البيئة التفرعية لتحقيق نتائج بزمن أقل ودقة مماثلة، وتتجلى أهمية البحث أيضاً في تحويل خوارزميات المعلوماتية الحيوية المكتوبة بشكل تسلسلي إلى شكل تفرعي بحيث يسمح بتنفيذها على البيئة الموزعة واستخدام بيئة الحوسبة السحابية Google Colab للعمل عليه، ودراسة تأثير زيادة عدد النيايب Threads على عمل الخوارزمية من حيث الدقة والزمن والكلفة.

## 3. طرائق البحث ومواده

### 3.1. المعلوماتية الحيوية

علم المعلوماتية الحيوية هو مجال بحث ناشئ حديثاً ومتعدد التخصصات يجمع بين علوم الكمبيوتر والرياضيات والعلوم البيولوجية لحل وشرح وتفسير العديد من التساؤلات البيولوجية، ويتعامل هذا العلم مع مجموعة البيانات البيولوجية الهائلة الناتجة من المشاريع العلمية الكبيرة مثل مشروع الجينوم البشري، ويوجد مجالين كبيرين من العلوم البيولوجية يستخدم فيهما علم المعلوماتية الحيوية بشكل أساسي ومتزايد وهما علم الجينوم Genomics (دراسة التركيب والوظيفة للمجموع الكلي لجينات الكائن الحي)، وعلم البروتين Proteomics (دراسة التركيب والوظيفة للمجموع الكلي لبروتينات الكائن الحي) [6]. بالإضافة إلى ذلك فإن علم المعلوماتية الحيوية يستخدم في فروع كثيرة من العلم البيولوجي مثل علم خرائط التمثيل الغذائي وشبكات الاتصال الخلوية والتفاعلات البيوكيميائية وغيرها، وتهدف كل هذه النواحي في علم المعلوماتية الحيوية إلى فهم الأنظمة البيولوجية المعقدة.

وقد أدى التوسع الكبير في هذا المجال إلى ارتباطه بمجموعة كبيرة من العلوم الأخرى في محيط الكائن الحي ولينتج مايسمى بالنظام البيولوجي المتكامل "Integrated System Biology" حيث أن النظام البيولوجي المتكامل يعدّ وسيلة لحل وتفسير العديد من التساؤلات البيولوجية المعقدة، وهو يتكون من اندماج علم الجينوم وعلم البروتين وعلم المعلوماتية الحيوية والعديد من العلوم الأخرى المتعلقة بها، وذلك لشرح وتفسير كيفية عمل المسار الحيوي في الخلية والتعرف على كيفية تفاعل الجينات والبروتينات والعناصر الأخرى الداخلية والخارجية المرتبطة بالمسار الحيوي مع بعضها البعض ومع البيئة المحيطة [6].

### 3.2. أهداف المعلوماتية الحيوية

الهدف الأساسي للمعلوماتية الحيوية هو فهم الخلية الحية بشكل أفضل وكيف تعمل على المستوى الجزيئي. من خلال تحليل التسلسل الجزيئي الخام والبيانات الهيكلية، يمكن لأبحاث المعلوماتية الحيوية أن تولد رؤى جديدة وتوفر منظوراً عالمياً للخلية.

السبب في أن وظائف الخلية يمكن فهمها بشكل أفضل عن طريق تحليل بيانات التسلسل لأن تدفق المعلومات الوراثية في نهاية المطاف تمليه "العقيدة المركزية" لعلم الأحياء التي يتم فيها نسخ الحمض النووي DNA [5] إلى الحمض النووي الريبي RNA، والذي يتم ترجمته إلى البروتينات. ويتم تنفيذ الوظائف الخلوية (من الخلية) بشكل رئيسي عن طريق البروتينات التي يتم تحديد قدراتها في نهاية المطاف من خلال تسلسلها، لذلك ثبت أن حل المشكلات الوظيفية باستخدام التسلسل والنهج الهيكلي في بعض الأحيان هو محاولة مثمرة.

#### 3.2.1. المادة الوراثية DNA:

هو أحد المركبات الجزيئية البيولوجية المشهورة في الجسم (الشكل (2))، فكل خلية في الجسم تحتوي على الحمض النووي والذي يقع في داخل نواة الخلية، والتي تميزك عن غيرك من الكائنات الحية. تحمل المادة الوراثية المعلومات والإرشادات الخاصة بالتطور والنمو والتكاثر على الجينات، حيث يتم انتقال المعلومات الجينية من الوالدين إلى الأطفال، فتقريباً نصف هذه المعلومات يكون من الأب والنصف الثاني من الأم. التغيير الجيني في هذه المادة الوراثية يعد السبب وراء امتلاك أحد الأشخاص عيوناً زرقاء اللون بدلاً من البنية، والسبب من وراء إصابة البعض بأمراض معينة، وكل الأمور التي تؤدي إلى تكوينك أنت وتميزك عن الآخرين [5]. بالمختصر تحتوي المادة الوراثية على الرموز الجينية المميزة لكل إنسان، والمعلومات اللازمة لبناء البروتينات المهمة لوظائف الجسم المختلفة.



الشكل (2) شريط الـ DNA.

#### 3.2.1.1. تركيب المادة الوراثية DNA

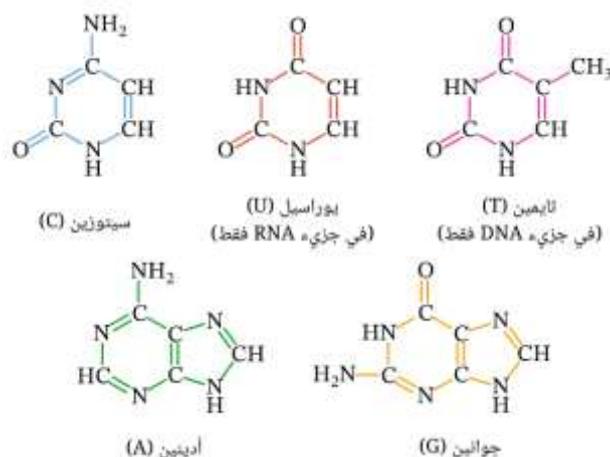
تتكون المادة الوراثية من جزيئين يلتفان حول بعضهما البعض، كل واحد من هذه الجزيئات عبارة عن تسلسل طويل من النيوكليوتيدات (Nucleotides) والتي تتكون من الآتي [5]:

- جزيء الفوسفات.
- جزيء السكر ويدعى ريبوز منقوص الأكسجين (Deoxyribose) وهو يتكون من 5 ذرات كربون.
- جزيء يحتوي على النيتروجين، وهناك عدة أنواع منها:

- الأدينين (Adenine).
- السيتوزين (Cytosine).
- الغوانين (Guanine).
- الثيمين (Thymine).

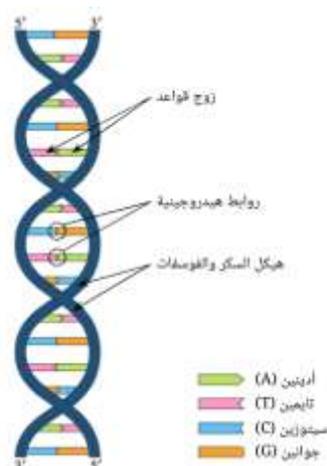
يوضح الشكل (3) التركيب الكيميائي للقواعد النيتروجينية الخمس. وترتيب هذه المناطق هو ما يشكل الرموز الجينية.

والجدير بالذكر أن ارتباط السلاسل مع بعضها لتشكيل المادة الوراثية يكون من خلال ارتباط الأدينين مع الثيمين والسيتوزين مع الغوانين، وأن طول المادة الوراثية يتحدد من قبل جزيئات السكر والفوسفات.



الشكل(3) التركيب الكيميائي للقواعد النيتروجينية الخمس.

يوضح الشكل (4) نموذج الحمض النووي DNA و يوضح التركيب اللولبي المزدوج وأزواج القواعد التي تتصل معاً بروابط هيدروجينية.



الشكل(4) نموذج للحمض النووي DNA.

### 3.2.2. الأحماض الأمينية

الأحماض الأمينية هي الوحدات الصغيرة المتكررة المرتبطة مع بعضها بعض لتكون مركبات معقدة ذات أوزان جزيئية عالية هي البروتينات، ومركبات أبسط منها، مكونة من عدد أقل من الأحماض الأمينية هي الببتيدات.

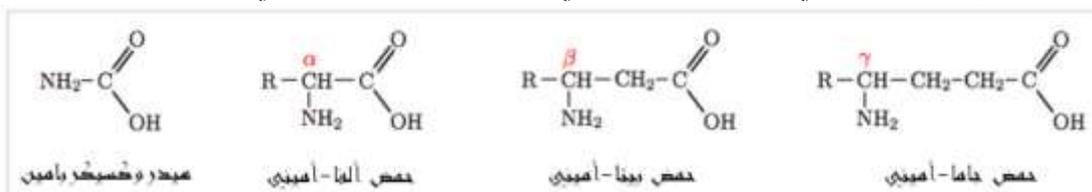
وهي مركبات عضوية تحتوي في الجزئ الواحد منها على مجموعتين وظيفيتين فعاليتين، هما:

• مجموعة الأمين ( $-NH_2$ ) القاعدية

• مجموعة الكربوكسيل ( $-COOH$ ) الحمضية.

يوضح الشكل (5) البنية الكيميائية العامة للأحماض الأمينية مصنفة حسب مكان ترابط الجذر الأميني  $NH_2$

فوق السلسلة الكربونية، حيث R هي المجموعة الجانبية التي تحدد طبيعة كل حمض أميني.



الشكل(5) البنية الكيميائية العامة للأحماض الأمينية.

### ❖ البروتينات:

تتكون من أعداد كبيرة من الأحماض الأمينية. وهي عبارة عن ميلمرات طويلة من الأحماض الأمينية[5].

• تلعب البروتينات دور حيوي في الجسم، فهي:

1- تشكل نصف وزنه الجاف.

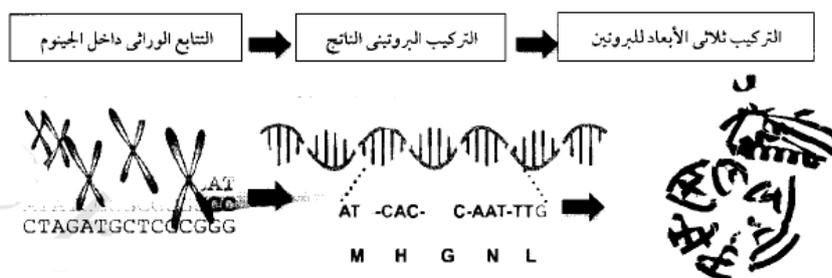
2- تتصل بمعظم نشاطاته، فهي موجودة في الجلد والشعر، والعضلات لتأمين

الحماية لها للحركة، وفي العظام لتعطيها الهيكل، وفي الدم والأعصاب.

3- لها دور مهم في تحفيز التفاعلات التي تمد الجسم بالطاقة ومواد النمو، هذه

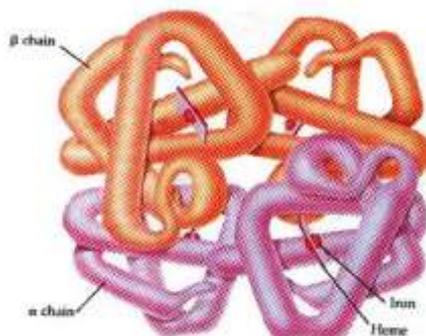
البروتينات تسمى بالإنزيمات ( التي تحفز هذه التفاعلات).

الشكل (6) يوضح التركيب ثلاثي الأبعاد للبروتين والذي يؤثر في نشاط و وظيفة البروتين:



الشكل(6) التركيب ثلاثي الأبعاد للبروتين والذي يؤثر في نشاط و وظيفة البروتين.

الشكل (7) يُظهر تركيب بروتين الهيموغلوبين المستخدم في دراستنا.



الشكل(7) تركيب بروتين الهيموغلوبين.

❖ **البيبتيدات:**

تتكون من أعداد صغيرة من الأحماض الأمينية. أصغرها حمضين أمينيين يرتبطان برابطة بيبتيدية واحدة. أي أن البيبتيدات عبارة عن مبلمرات قصيرة من الأحماض الأمينية.

**3.3. تطبيقات المعلوماتية الحيوية**

لم تصبح المعلوماتية الحيوية ضرورية فقط لأبحاث البيولوجيا الجينومية والجزيئية الأساسية، فقد أصبح لها تأثير كبير على العديد من مجالات التكنولوجيا الحيوية وعلوم الطب الحيوي، وقد تم استخدامها في تصميم الأدوية القائمة على المعرفة وتحليل الحمض النووي الشرعي، والتكنولوجيا الحيوية الزراعية [7].

تسمح معرفة الهياكل ثلاثية الأبعاد للبروتينات بتصميم الجزيئات القادرة على الارتباط بموقع المستقبل للبروتين المستهدف مع تقارب وخصوصية كبيرة، يقلل هذا النهج القائم على المعلوماتية بشكل كبير من الوقت والتكلفة اللازمين لتطوير أدوية ذات فاعلية أعلى، وأعراض جانبية أقل، ودرجة سُمية أقل من استخدام نهج التجربة والخطأ التقليدي، في الطب الشرعي تم قبول نتائج التحليل الجزيئي للسلاسل كدليل في المحاكم الجنائية [3][7].

**3.4. خوارزميات المعلوماتية الحيوية**

غالباً ماتقدم الخوارزميات تنبؤات غير صحيحة لا معنى لها عند وضعها في سياق بيولوجي، يمكن أن تؤثر الأخطاء في محاذاة التسلسل على نتائج التحليل الهيكلي أو تطور السلالات، تعتمد نتيجة الحساب أيضاً على قوة الحوسبة المتاحة، لذلك لا يمكن استخدام العديد من الخوارزميات الدقيقة والشاملة بسبب بطء معدل الحساب، بدلاً من ذلك يجب استخدام خوارزميات أقل دقة وأكثر سرعة، هذه مفاضلة ضرورية بين الدقة والجودة الحسابية، أو حتى المفاضلة بين الأناقة والبساطة في الخوارزميات كما يقال في أوساط البرمجة، لذلك من المهم الأخذ بعين الاعتبار احتمال حدوث أخطاء تنتج عن برامج المعلوماتية الحيوية.

الخوارزميات المستخدمة في المعلوماتية الحيوية [8-11]:

❖ **خوارزمية Needleman-Wunsch algorithm**

خوارزمية Needleman-Wunsch هي خوارزمية مستخدمة في المعلوماتية الحيوية لمحاذاة تسلسل البروتين أو النوكليوتيدات. تقسم الخوارزمية المشكلات الكبيرة (مثل التسلسل الكامل) إلى سلسلة من المشكلات الأصغر، وتستخدم حلول المشكلات الأصغر لإيجاد الحل الأمثل للمشكلة الأكبر. يشار إليها أيضاً باسم خوارزمية المطابقة المثلى وتقنية المحاذاة العامة، والغرض من الخوارزمية هو العثور على جميع المحاذاة الممكنة التي لها أعلى درجة. سنعتمد على هذه الخوارزمية في هذا البحث.

❖ **خوارزمية Smith-Waterman algorithm**

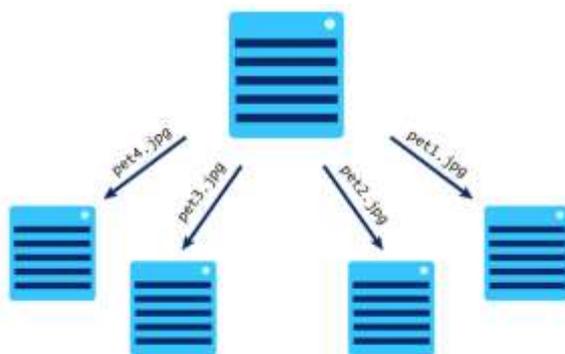
تقوم خوارزمية Smith - Waterman بمحاذاة التسلسل المحلي، أي لتحديد مناطق متشابهة بين سلسلتين من متواليات الحمض النووي أو متواليات البروتين، بدلاً من النظر إلى التسلسل بأكمله، وهذه الطريقة تكون مناسبة لسلسلة من التتابعات المتباعدة أو الحاوية على مجالات متعددة والتي تكون من الأصل مختلفة.

**3.5. الحوسبة التفرعية وبيئة الحوسبة الموزعة**

هي مجال من مجالات هندسة الحاسبات، يدرس ما يسمى بالأنظمة الموزعة distributed system وهي عبارة عن نظام برمجيات يتألف من مجموعة من أجهزة الحواسيب التي تتصل مع بعضها بعض من خلال شبكة موصلة فيما بينها. يكون التواصل والتفاعل بين هذه الأجهزة من خلال تمرير رسائل passing messages،

وتتفاعل هذه المكونات مع بعضها بعض من أجل تحقيق هدف مشترك. تسمى برامج الحاسوب التي تعمل على الأنظمة الموزعة ببرامج موزعة.

في الحوسبة الموزعة، يتم تقسيم المشكلة إلى مشاكل صغيرة، ثم يتم توزيعها على الحواسيب ليتم حل هذه المشكلة كما هو موضح في الشكل(8).



الشكل(8) مثال على التحضير للعمل في البيئة الموزعة.

### 3.6. موقع NCBI

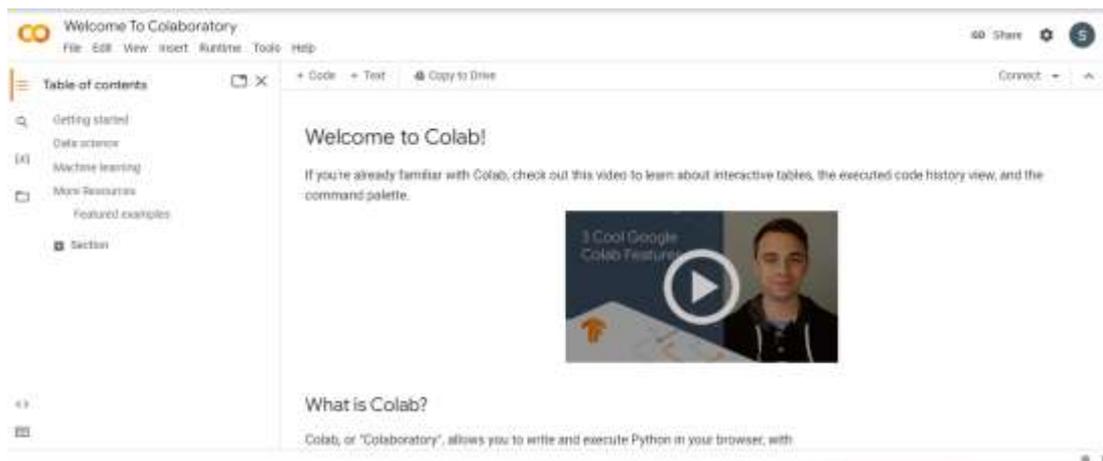
موقع NCBI (National Center for Biotechnology Information) هو جزء من المكتبة الوطنية للطب في الولايات المتحدة المتفرعة عن معاهد الصحة العالمية. يضم المركز بيانات الجينات المتسلسلة في بنك الجينات وفهرساً للمقالات والبحوث الطبية الحيوية، بالإضافة الى معلومات ضخمة عن سلاسل ال DNA و RNA والبروتين لمختلف الكائنات، وجميع هذه البيانات متوفرة على شبكة الإنترنت، يبين الشكل (9) واجهة هذه البيئة.



الشكل(9) واجهة موقع NCBI.

### 3.7. Google Colab

يُعد Google Colab نسخةً سحابيةً مجانيةً من Jupyter، تعمل على خوادم غوغل السحابية[9]، والتي تتيح للمستخدم الاستفادة من العديد من الميزات مثل وحدة معالجة الرسومات GPU والذاكرة الإضافية والأجهزة الخلفية، ويمكنه من القيام بكل الوظائف التي يتم تنفيذها عادةً على برنامج Jupyter Notebook المثبت على جهاز الحاسوب دون الحاجة إلى تهيئة أي من الإعدادات. يقدم Google Colab جميع الإعدادات التي نحتاجها لبدء تنفيذ المشاريع البرمجية المتعلقة بعلم البيانات والتعلم العميق[10]، واجهة Google Colab تظهر كما في الشكل (10).



الشكل (10) واجهة موقع Google Colab.

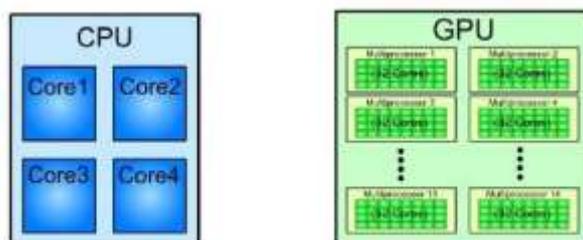
### 3.8. معالجة GPU:

وحدة معالجة الرسوم GPU (Graphics Processing Unit) عبارة عن معالج متوازي مخصص لتسريع العمليات الحسابية الرسومية وإجراء عمليات الفاصلة العائمة اللازمة لعرض الرسوم ثلاثية الأبعاد، والتي تفوق قدرة وحدة المعالجة المركزية CPU لتنفيذها.

تم تصميم وحدات معالجة الرسوم لتسريع عرض الرسومات ثلاثية الأبعاد في الزمن الحقيقي، مثل تطبيقات الألعاب. لذا تزايدت الطلبات على معالجات رسومية أكثر سرعة خصوصاً بعد ظهور تقنيات العرض المتطورة مثل شاشات 4K وظهور ألعاب الواقع الافتراضي.

بسبب التطور الكبير في تقنيات الذكاء الاصطناعي AI والتعلم الآلي ML وانتشرت الأشياء IoT [13] وحاجتها إلى كميات هائلة من البيانات، أصبحت الحاجة إلى تحسين أداء الحوسبة ملحة كون الحواسيب النموذجية غير قادرة على القيام بهذه المهام بشكل سريع. تم تحقيق هذه المتطلبات من خلال استخدام الحوسبة عالية الأداء التي يمكنها معالجة المليارات من العمليات الحسابية المعقدة بسرعات عالية جداً وبالتوازي [1].

تختلف نواة GPU تماماً عن نواة CPU. تم تصميم نواة CPU بمنطق تحكم معقد للغاية لتنفيذ البرامج التسلسلية، بينما تم تصميم نواة GPU بمنطق تحكم أبسط يركز على تشغيل مهام البيانات المتوازية، حيث تحتوي GPU على آلاف النوى الصغيرة والأكثر كفاءة من CPU متعددة الأنوية multicores كما في الشكل (11). توفر CPU متعددة الأنوية ذاكرة تخزين مؤقت كبيرة وتتخذ تعليمات بطول x86 كاملة على كل نواة، في حين أن نوى وحدة معالجة الرسوم الأصغر مخصصة للنقطة العائمة للإنتاجية. يتم تشغيل الأجزاء المتسلسلة من حمل العمل في سلسلة واحدة على وحدة المعالجة المركزية للجهاز، وتعمل الأجزاء كثيفة الحوسبة بالتوازي على آلاف من نوى وحدة معالجة الرسوم [1].



الشكل (11) بنية معالج GPU و معالج CPU.

#### 4. خوارزمية Needleman – Wunsch

خوارزمية Needleman–Wunsch هي خوارزمية مستخدمة في المعلوماتية الحيوية لمحاذاة تسلسل البروتين أو النوكليوتيدات. وهي أحد التطبيقات الأولى للبرمجة الديناميكية لمقارنة التسلسلات البيولوجية. تم تطوير الخوارزمية بواسطة Saul B. Needleman و Christian D. Wunsch وتم نشرها في عام 1970 [8–11].

تقسم الخوارزمية أساساً مشكلة كبيرة (مثل التسلسل الكامل) إلى سلسلة من المشكلات الأصغر، وتستخدم الحلول للمشكلات الأصغر لإيجاد الحل الأمثل للمشكلة الأكبر. يشار إليها أيضاً باسم خوارزمية المطابقة المثلى وتقنية المحاذاة العامة. لا تزال خوارزمية Needleman–Wunsch مستخدمة على نطاق واسع لتحقيق التوافق الشامل الأمثل، لا سيما عندما تكون جودة المحاذاة الشاملة ذات أهمية قصوى. تقوم الخوارزمية بتعيين درجة لكل محاذاة ممكنة، والغرض من الخوارزمية هو العثور على جميع المحاذاة الممكنة التي لها أعلى درجة (الشكل (12)) [8].

Needleman-Wunsch

match = 1    mismatch = -1    gap = -1

		G	C	A	T	G	C	G
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

الشكل (12) مثال عملي للخوارزمية.

#### 4.1. خطوات تنفيذ الخوارزمية

يمكن شرح خطوات الخوارزمية على النحو الآتي:

- يتم أولاً بناء مصفوفتين ذات بعدين  $(M+1, N+1)$  حيث  $M$  هي طول السلسلة الأولى، و  $N$  طول السلسلة الثانية. نملأ السطر الأول بحارف السلسلة الأولى والعمود الأول بحارف السلسلة الثانية. المصفوفة الأولى هي مصفوفة الدرجات حيث نتبع أفضل طريقة ممكنة للوصول إلى النقطة التي يصطف فيها كل زوج من الأزواج المتقابلة. والثانية هي مصفوفة التتبع الخاصة بنا (لاختيار أفضل مسار) كما هو موضح في الشكل (13).

	-	G	A	T	A	C	C	A		-	G	A	T	A	C	C	A
-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0
G	-	-	-	-	-	-	-	-	-	G	0	0	0	0	0	0	0
C	-	-	-	-	-	-	-	-	-	C	0	0	0	0	0	0	0
A	-	-	-	-	-	-	-	-	-	A	0	0	0	0	0	0	0
T	-	-	-	-	-	-	-	-	-	T	0	0	0	0	0	0	0
G	-	-	-	-	-	-	-	-	-	G	0	0	0	0	0	0	0
C	-	-	-	-	-	-	-	-	-	C	0	0	0	0	0	0	0
T	-	-	-	-	-	-	-	-	-	T	0	0	0	0	0	0	0

الشكل (13) مصفوفة الدرجات، مصفوفة التتبع.

- نبدأ عملية تحديد الدرجات عند الصف الأول ثم ننتقل إلى الصف الثاني وهكذا...

- نضع في المربع الأول الدرجة صفر ومنتقل إلى المربع المجاور له لملئه، نقارن المحرف المقابل له في العمود مع المحرف المقابل له في السطر.
- يتم حساب الدرجات باستخدام أحد مصفوفات الاستبدال على سبيل المثال مصفوفة الاستبدال BLOSUM62 [12] المبينة في الشكل (14).

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

الشكل (14) مصفوفة الاستبدال BLOSUM62.

- بفرض أن  $x[i,1]$  و  $y[1,j]$  هم التتابعين المراد دراسة تطابقهما ، نعرف مصفوفة حساب الدرجات  $D(i,j)$  وفق العلاقة (1):

$$D(i,j) = \max \begin{cases} \text{diag} = D(i-1, j-1) + s(x_i, y_j) \\ \text{up} = D(i-1, j) + g \\ \text{left} = D(i, j-1) + g \end{cases} \quad (1)$$

حيث:

➤  $s(x_i, y_j)$  هي درجة تطابق  $x_i$  مع  $y_j$  وتحتسب من أحد مصفوفات الاستبدال

مثل BLOSUM62

➤  $g$  هي قيمة الفجوة الخطية، أي  $g$  هي جزء إضافة فجوة  $(y_j, -)$  أو  $(x_i, -)$

➤  $D(i-1, j-1)$  هي قيمة الدرجة القطرية السابقة.

➤  $D(i-1, j)$  قيمة الدرجة من الصف السابق.

➤  $D(i, j-1)$  قيمة الدرجة من العمود السابق.

• يتم ملئ مصفوفة التتبع بالتوازي مع مصفوفة لدرجات خطوة بخطوة.

• استنتاج التقابل الأفضل من مصفوفة التتبع

• تصبح مصفوفة الدرجات والتتبع بعد ملء القيم، تظهر المصفوفة المبينة في الشكل (15).

-	G	A	T	A	C	C	A	-	G	A	T	A	C	C	A	
-	x	+	+	+	+	+	+	+	0	-1	-2	-3	-4	-5	-6	-7
G	↑	↘	←	←	←	←	←	G	-1	1	0	-1	-2	-3	-4	-5
C	↑	↑	↘	←	←	←	←	C	-2	0	0	-1	-2	-1	-2	-3
A	↑	↑	↘	←	←	←	←	A	-3	-1	1	0	0	-1	-2	-1
T	↑	↑	↑	↘	←	←	←	T	-4	2	0	2	1	0	-1	2
G	↑	↑	↑	↑	↘	←	←	G	-5	-3	-1	1	1	0	-1	2
C	↑	↑	↑	↑	↑	↘	←	C	-6	-4	-2	0	0	2	1	0
T	↑	↑	↑	↑	↑	↑	↘	T	-7	-5	-3	-1	-1	1	1	0

الشكل (15) مصفوفة الدرجات، مصفوفة التتبع بعد التنفيذ.

#### 4.2. تحضير سلسلة البروتين

الذهاب إلى موقع NCBI وإحضار سلسلة بروتين هيموغلوبين الدم، الشكل (16) يوضح الشكل الرسومي

لبروتين الهيموغلوبين.



الشكل (16) البنية الرسومية لبروتين الهيموغلوبين.

نحصل على سلسلة البروتين بالنقر على FASTA فتظهر لدينا كما هو موضح في الشكل (17).

```
FASTA
Arabidopsis thaliana chromosome 3, partial sequence
NCBI Reference Sequence: NC_003074.8
GenBank Graphics
>NC_003074.8:c3277930-3276163 Arabidopsis thaliana chromosome 3, partial sequence
ATTTAGTCAAAATAAAATAAAATTTGAGAGAACTTCTACATAAGTAGCTCACAGACCACCAAGGACC
ATTGAATACCATATATATAGATACACAGACATATAAACACACAAATATTCGTGTTTTTTTCAAACGT
GAGAGAAAAAGAGAGAGAGAGATGGGAGAGATTGGGTTTACAGAGAAAGCAAGCAAGCTTTGGTGAAG
GAATCGTGGAGATCTGAAACAGACATCCCCAAATCAAGCCTTCACTTCTTCCACAGTAACCATATA
TACTTAGTTATATAAAGCTCTTACATGTTGTTTATATATGCGAGCTAATGAACAATATAATTTGATGATA
GATAC TGGAGATAGCACAGCAGCAAGAGGCTTTGTTCTTTCTTAAGABACTCAGATGAAAGTCCCTCA
CAACATCTCAACCTCAAGCTCACTGCTTTTAAAGCTTTCAAGATGGTAAATTAATTAATTTTCCGATTTTC
CACATCTACATATATGTAATCACTTGCATATACATGATCATATCTTACACTTCTTAAATTTGAAGGT
AGAAATGTTTCAATTTTACAGCTAAGAACTTTATTTACTGATTATATACATTATATATAAATGAAAAA
TAGTAGTCTGAATTAACCTTTCTTTGCTATTATTTGACCGAGCCTATGTCACACAAGTTGCGAATTTGTTT
GAACCTTGGCTAGTTGGCTTTGTTCTTCTTTTGAAGATAAAATCTCATACTAGTAAAGAACTCTGTG
ATATTTTATTTTAAAGCAAAAGATAGATTTCTCTGTCAATAAAGAAATTTGTTACTGAAAGAACTCAAGTGG
TTGCGGCTGCTTTATGATTTTAACTTTTTCCTAATCTTATTAATAAGAACCATATAAACCAAAATTC
GTTTACTTTTAAATTTGGGTTTATGACTTGGTTTGGTTCAACTGACTTTTGGCTCTAAGACTTTGCAT
AACATGTTTTAGACAGACAAAAAGAAAAAGACTTTGCATAACATGATGAATTTTTTATTTATTTGTT
TGTGTTAGACATGTGAAACAGCTATACAGCTGAGGGAGGAAAGGTTGGTAGTGGCTGACACAACCC
TCCAAATTTAGGCTCAATTCATCTCAAAAGCGGCTTATTTGACCTCACTTCGAGCTGTATATGTTAA
AAAAAATATATATACACATTAATTTTGGCTGATTTTGAATTTGATTGAAAGCATTTAATAAGGTTG
GAATGTGAAGCAGGTGGTGAAGAAAGCTTTGCTAAGGACATTGAAAGAGGGGTTGGGGGAAATACAA
TGAAGAGTGGAAAGTGTGTTGCTCAAGCTTATGATCACTTGGCTTTAAGCATCAAGCCGAGATGAAA
CAAGAAAGATCATAAAACCTTATTTGATTTGGGATCCGATACATGAACTATTCCACATACATGATA
CACATATACGTGTTTCTGTGTGACTATGTTGCTCTGACTTCTCAGATTCACTATTTAATTTATA
AAGAAAGATCTTGTGCTATCATTAGGAGATACGTGATAGTTCTTCTTGAATTTGTTATTCGTGA
GAAATATCATGGTTTGAAGTATTTATTTTACAAAGATGGATTTAACGTGGGATCATTTTACAACTATT
CTACAAATAATTTACTTCTCCAGACCAAGATTTGGATCTATTTAGTTCAATTTTTCTTTTTTTAA
AAAAAGTACACTAAAAA
```

الشكل (17) سلسلة البروتين.

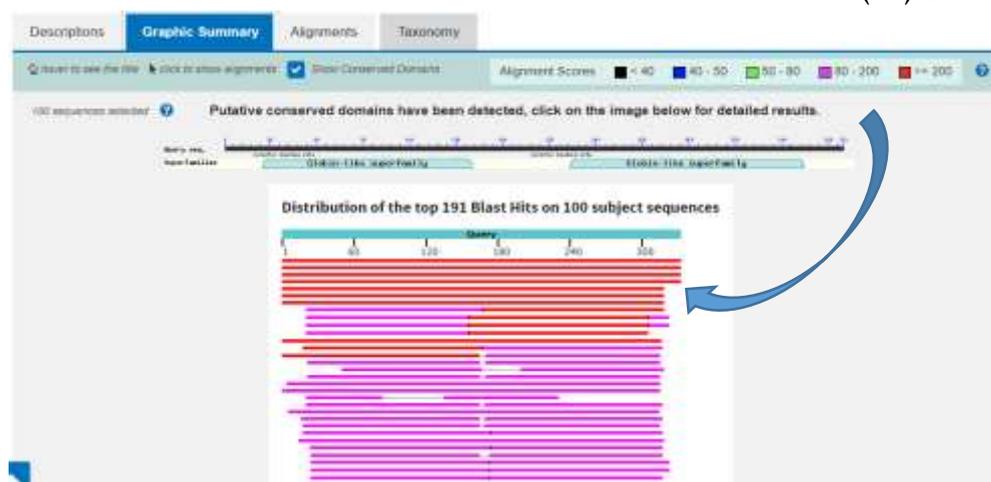
نقوم بنسخ السلسلة وحقن بعض الأخطاء بها ثم الذهاب إلى موقع BLAST لمعرفة درجة التطابق بين السلسلة

المحقونة بالأخطاء والسلسلة الأساسية الموجودة في قاعدة بيانات الموقع، تظهر لدينا النتائج المبينة في الشكل (18).

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
blastp Pseudomonas deciesera	Pseudomonas	692	692	100%	0.0	100.00%	303	AAA29706.1
blastp Pseudomonas deciesera	Pseudomonas	655	655	100%	0.0	96.58%	333	CAA77741.1
blastp Pseudomonas deciesera	Pseudomonas	654	654	100%	0.0	96.20%	333	E20834.1
blastp Acetabacter simplex	Acetabacter simplex	550	550	100%	0.0	79.04%	332	BAX3000.1
blastp Acetabacter simplex	Acetabacter simplex	537	537	99%	0.0	79.69%	332	AS138218.2
blastp Acetabacter simplex	Acetabacter simplex	537	537	95%	0.0	79.69%	332	BAX3002.1
blastp Acetabacter simplex	Acetabacter simplex	533	533	95%	0.0	79.06%	332	BAX3004.1
blastp Acetabacter simplex	Acetabacter simplex	502	502	99%	1e-176	79.46%	308	AF193326.1
blastp Acetabacter simplex	Acetabacter simplex	445	703	90%	1e-155	78.33%	283	AY04621.1
blastp Acetabacter simplex	Acetabacter simplex	443	701	90%	3e-154	78.33%	283	AY04623.1

الشكل (18) نتائج المحاكاة على BLAST.

نلاحظ درجة التطابق في الرتل الأول 100% ثم يبدأ بالتناقص والسبب في ذلك هو أنه في بداية السلسلة لم يتم حقن أي خطأ ثم البدء بحقن أخطاء عشوائية متباعدة ومتقاربة، والشكل البياني لنتيجة المحاكاة تتوضح في الشكل (19).



الشكل (19) الرسم البياني لنتائج المحاكاة.

الألوان المختلفة التي تظهر في الرسم البياني تعود لدرجة التطابق بين السلسلتين والموضحة في بداية الرسم.

#### 4.3 تنفيذ خوارزمية Needleman-Wunsch على معالج تسلسلي

خوارزمية Needleman-Wunsch هي خوارزمية تسلسلية مكتوبة للعمل على البرامج التسلسلية، تمت في هذه المرحلة تعديل الخوارزمية بإضافة سلاسل البروتين الخاصة بدراستنا وتنفيذ هذه الخوارزمية على موقع Google Colab باستخدام معالج GPU فظهرت النتيجة كما يلي:

```

47 ry.append('/')
48 i -= 1
49 elif P[i, j] in [4, 6, 7, 9]:
50 ry.append('/')
51 ry.append(y[j - 1])
52 j -= 1
53 # Reverse the strings:
54 rx = ''.join(rx[::-1])
55 ry = ''.join(ry[::-1])
56 return '\n'.join([rx, ry])
57
58 x = "ZTGDCEAAATMYTAAATTTGAGQSKDCDFACATAGTACTCACAGACCCACCAAGGACCTTGAWHB
59 y = "ATTTAGTCEKJTHAAGTAAATTUJPSGMAATCTTETACATAGTACTCACAGACCCACCAAGGACCTTGAWH
60 print(rx, y)
61
62
-ZTGDCE-----MATAMHYTAAATTTGAGQSKDCDFACATAGTACTCACAGACCCACCAAGGACCTTGAWH
ATTTAGTCEKJTHAAGTAAATTUJPSGMAATCTTETACATAGTACTCACAGACCCACCAAGGACCTTGAWH
    
```

الشكل (20) نتائج تنفيذ الخوارزمية.

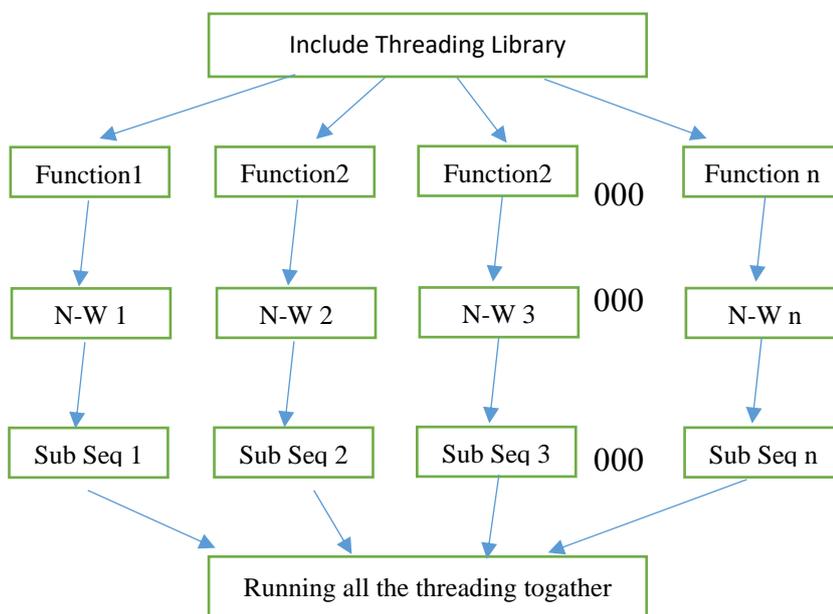
نلاحظ أن تنفيذ هذه الخوارزمية استغرق زمن قدره 20s

#### 4.4 تطوير نسخة تفرعية من خوارزمية Needleman-Wunsch

نبين تطوير خوارزمية Needleman-Wunsch التسلسلية إلى خوارزمية تفرعية وفق الخطوات التالية:

1. تضمين مكتبة Threading.
2. تضمين عدد من التتابع بعدد النيايب المراد التنفيذ عليها.
3. إدخال خوارزمية Needleman-Wunsch على كل تابع من التتابع.
4. تقسيم السلسلة المدروسة وتوزيعها على التتابع.
5. إعطاء الأمر لجميع النيايب للعمل بشكل متوازٍ.

ويمكن تلخيص هذه الخطوات وفق المخطط الصندوقي التالي (الشكل (21)):

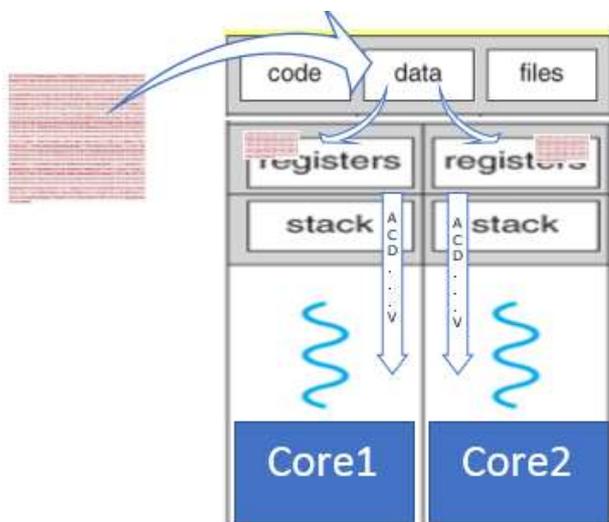


الشكل (21) المخطط الصندوقي لتطوير خوارزمية Needleman-Wunsch التسلسلية إلى خوارزمية تفرعية

#### 4.4.1 تنفيذ خوارزمية Needleman-Wunsch على معالج تفرعي باستخدام نيسبين متوازيين

two threads

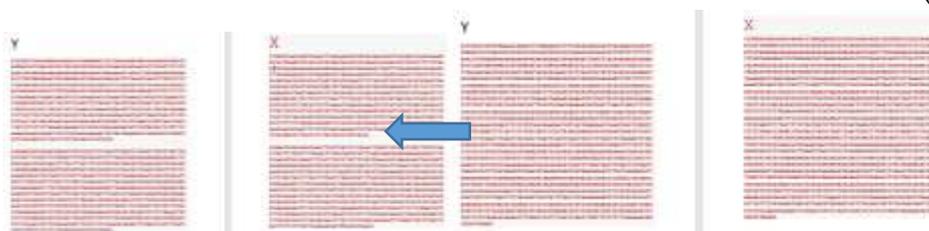
قمنا بتهيئة بيئة العمل في Google Colab للعمل على معالجات GPU باستخدام 2Thread بشكل متوازٍ وفق الشكل (22):



الشكل (22) معالج GPU يعمل على 2Thread

ثم قمنا بتطوير آلية عمل الخوارزمية السابقة لتصبح قادرة للعمل على معالج GPU باستخدام two threads وفق الخطوات الآتية:

- تحضير البيانات للعمل على two threads بتتصيف كل سلسلة من السلسلتين السابقتين (الشكل (23)).



الشكل (23) تحضير السلاسل.

- تضمين مكتبة Threading في الكود:

```
1 import numpy as np
2 import threading
```

- إعطاء الأمر لعمل التابعين بشكل متوازي:

```
131 t1 = threading.Thread(target=f1)
132 t2 = threading.Thread(target=f2)
133 t1.start()
134 t2.start()
```

- اختبار الخوارزمية للتأكد من عملها بشكل صحيح، ثم إدخال السلاسل بعد التعديل وتنفيذ هذه الخوارزمية فظهرت النتيجة كما في الشكل (24).

```

120 print(nw(x, y))
121
122
123 t1 = threading.Thread(target=f1)
124 t2 = threading.Thread(target=f2)
125 t1.start()
126 t2.start()
127

```

Resources X  
Python 3 Google Compute Engine backend (GPU)  
Showing resources from 10:05 AM to 10:30 AM

System RAM  
1.6 / 12.7 GB

GPU RAM

Disk  
24.1 / 78.2 GB

الشكل (24) نتائج تنفيذ الخوارزمية.

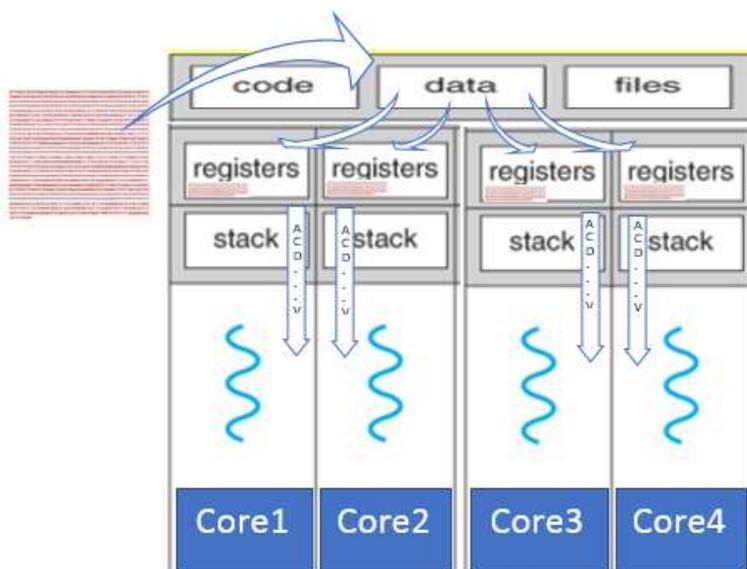
نلاحظ أن تنفيذ هذه الخوارزمية استغرق زمن قدره 5s، أي ما يعادل ربع زمن التنفيذ للخوارزمية السابقة، أما بالنسبة للنتائج التي ظهرت فهي ذاتها في الخوارزمتين.

#### 4.4.2 تنفيذ خوارزمية Needleman-Wunsch على معالج تفرعي باستخدام 4 نيايب متوازية

##### 4threads

قمنا بتهيئة بيئة العمل في Google Colab للعمل على معالجات GPU باستخدام 4Thread بشكل متوازٍ

وفق الشكل(25):



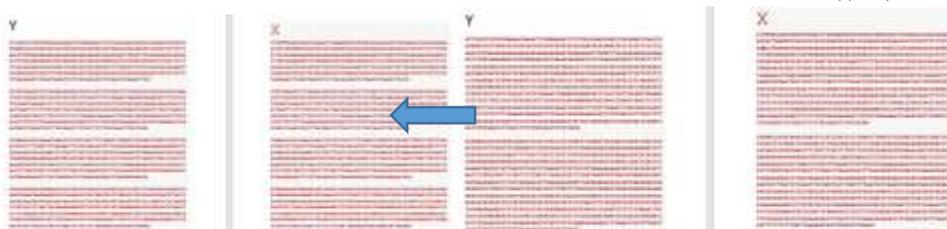
الشكل(25) معالج GPU يعمل على 4Thread

بعد ذلك قمنا بتطوير آلية عمل الخوارزمية السابقة لتصبح قادرة للعمل على معالج GPU باستخدام

4threads وفق الخطوات الآتية:

- تحضير البيانات للعمل على 4thread بتتصيف كل سلسلة من السلسلتين السابقتين في 2thread

(الشكل (26)).



الشكل(26) تحضير السلاسل.

## • تضمين مكتبة Threading في الكود:

```
1 import numpy as np
2 import threading
```

## • إعطاء الأمر لعمل التتابع الأربعة بشكل متوازي

```
39 t1 = threading.Thread(target=f1)
40 t2 = threading.Thread(target=f2)
41 t3 = threading.Thread(target=f3)
42 t4 = threading.Thread(target=f4)
43 t1.start()
44 t2.start()
45 t3.start()
46 t4.start()
```

## • اختبار الخوارزمية لتأكد من عملها بشكل صحيح، ثم إدخال السلاسل بعد التعديل وتنفيذ

هذه الخوارزمية فظهرت النتيجة كما في الشكل (27).

```
+ Code + Text
236 print(w(x, y))
237
238
239 t1 = threading.Thread(target=f1)
240 t2 = threading.Thread(target=f2)
241 t3 = threading.Thread(target=f3)
242 t4 = threading.Thread(target=f4)
243 t1.start()
244 t2.start()
245 t3.start()
246 t4.start()
247

-ITSWEE-...AAATAAHTAAAT- TGAGQASXC- DFCACATAAGTAGCTCACAGCCCAACCAANGGALCATTGAANNI
ATTGATCEIKKJHAA-AA--TAAATTLDSAG-A-ATCTTETACATAAGTAGCTCACAGCCCAACCAANGGALCATTGAANNI
TAGARCCATATAAACCAAAATTCGGTTTAC TTTTAAATTTGGGTTTAFGACTGGGTTGGTTCACACTCACTTTTGGCTTCAAGI
ATAGARCCATATAAACCAAAATTCGGTTTAC TTTTAAATTTGGGTTTAFGACTGGGTTGGTTCACACTCACTTTTGGCTTCAAGI
--ATGAGSAGTGGAGGTC TFGGCTCANGCTTATGATCACTTGGCTTTAGCCATCAAGACCGAGATGAACAGAGAGTCA
CAATGAAGAGTGGAGGTC TFGGCTCANGCTTATGATCACTTGGCTTTAGCCATCAAGACCGAGATGAACAGAGAGTCA
-CTGTTAANGTCTCANGATGGTAATTAETTAETTCOGATTTCCACATC TACATATATGTGAATCACTTGCATATACTGTATCA
GCTGTTAANGTCTCANGATGGTAATTAETTAETTCOGATTTCCACATC TACATATATGTGAATCACTTGCATATACTGTATCA
```

الشكل (27) نتائج تنفيذ الخوارزمية.

نلاحظ أن تنفيذ هذه الخوارزمية استغرق زمن قدره **1.52s**، أي ما يعادل ربع زمن التنفيذ للخوارزمية المنفذة على 2thread تقريباً، أما بالنسبة للنتائج التي ظهرت فهي ذاتها في الخوارزميتين السابقتين.

الجدول (1) يوضح مقارنة حول أهم النتائج التي توصلنا إليها من الخوارزميات الثلاث السابقة.

الجدول (1) يوضح مقارنة حول أهم النتائج التي توصلنا إليها من الخوارزميات الثلاث السابقة.

4 Thread	2 Thread	NO Thread	خوارزمية Needleman- Wunsch
1.52 s	5 s	20 s	زمن التنفيذ
أربع معالجات/ معالج بأربع أنوية	معالجين اثنين/ معالج بنواتين	معالج واحد	عدد المعالجات
أداء ممتاز وسريع	أداء متوسط	أداء ضعيف	الأداء
مماثلة للخوارزميتين	مماثلة للخوارزميتين	مماثلة للخوارزميتين	نتيجة التنفيذ
في هذه الخوارزمية لقد كان الربح في زمن التنفيذ و لكن على حساب العتاد الصلب المستخدم	في هذه الخوارزمية لقد كان زمن التنفيذ والعتاد الصلب المستخدم مقبول مقارنة مع التنفيذ التسلسلي	في هذه الخوارزمية استخدمنا معالج واحد فالربح هنا في العتاد الصلب أما زمن التنفيذ فقد كان سيئ جداً	الكلفة

من الجدول السابق نلاحظ أن تنفيذ الخوارزمية على البيئة التفرعية باستخدام 2Thread خفّض زمن التنفيذ بنسبة 75%، كما أن تنفيذها على البيئة التفرعية باستخدام 4Thread خفّض زمن التنفيذ بنسبة 92.4%، لكن هذا الزمن نحصل عليه على حساب الزيادة الكبيرة والمكلفة في حجم العتاد الصلب، لذلك لا بد من الأخذ بعين الاعتبار حجم العتاد الصلب المستخدم، فبالمقارنة نجد أن تنفيذ الخوارزمية على 2Thread هو الخيار الأفضل بما يضمن لنا أفضل زمن تنفيذ مع أقل حجم للعتاد الصلب.

## 5. الاستنتاجات والتوصيات

من خلال تقييم أداء خوارزمية Needleman-Wunsch تم التوصل الى الاستنتاجات الآتية:

- 1- إمكانية إضافة أي سلسلة إلى الخوارزمية مهما كان حجمها.
  - 2- نقصان زمن التنفيذ في الخوارزمية الثانية إلى ربع الزمن الذي استغرقته الخوارزمية الأولى مع الحفاظ على النتائج نفسها.
  - 3- نقصان زمن التنفيذ في الخوارزمية الثالثة إلى ربع الزمن الذي استغرقته الخوارزمية الثانية مع الحفاظ على النتائج نفسها.
  - 4- زمن تنفيذ الخوارزمية في البيئة الموزعة أقل بكثير من زمن تنفيذها في البيئة التسلسلية.
- يمكننا تطوير بحثنا باتجاه إجراء اختبار للخوارزمية السابقة بسلاسل مختلفة الطول وتنفيذ المهمة عدة مرات ومراقبة زمن التنفيذ، لمعرفة أثر تغير طول السلسلة على مدى فعالية الخوارزمية للتوصل الى أفضل أداء للخوارزمية من حيث طول السلسلة المستخدم وزمن التنفيذ ودقة النتائج وحجم الهاردوير.
- كما يمكن زيادة تفريع الخوارزمية الى 8Thread و 16Thread و 32Thread و دراسة تأثيرها على فاعلية الخوارزمية.

## المراجع العلمية

- [1] Dr. Hasan Albustani, Eng. Noor khder, "Analysis of the training time of convolutional neural networks on high-performance processors", Tartous University Journal for Research and Scientific Studies - engineering Sciences Series, 2022.
- [2] Masahito Ohue et al; "High-Performance Cloud Computing for Exhaustive Protein-Protein Docking", Tokyo Institute of Technology, Japan, (2020).
- [3] You Zou et al; "Parallel computing for genome sequence processing", OXFORD, Briefings in Bioinformatics,2021.
- [4] Jamie J Alnasir et al; "The Application of Hadoop in Structural Bioinformatics",NLM, Briefings in Bioinformatics,2020.
- [5] Ahmed Alzohairy;"Introduction to bioinformatics and genomics", Zagazig University,2013.
- [6] Marketa Zvelebil, Jeremy O. Baum; "Understanding Bioinformatics", Garland Science, Taylor & Francis Group, LLC, 2008.
- [7] P.M. Selzer et al; "Applied Bioinformatics", Springer, Verlag Berlin Heidelberg, 2008.
- [8] Miguel Rocha, Pedro G. Ferreira;"Bioinformatics Algorithms Design and Implementation in Python", ACADEMIC PRESS.
- [9] Aisling O'Driscoll, et al ;" 'Big Data', Hadoop and Cloud Computing in Genomics", Journal of Biomedical Informatics 46 (2013) 774–781.
- [10] Dr. Poornima Naik, et al ;"Conceptualizing Python in Google COLAB ",2022.
- [11] FN Muhamad, et al; " Performance Analysis Of Needleman-Wunsch Algorithm (Global) And Smith-Waterman Algorithm (Local) In Reducing Search Space And Time For Dna Sequence Alignment", 1st International Conference on Green and Sustainable Computing (ICoGeS) 2017, IOP Publishing, Series: Journal of Physics: Conf. Series 1019 (2018) 012085.
- [12] STEVEN HENIKOFF, et al ;"Amino acid substitution matrices from protein blocks", Proc. Natl. Acad. Sci. USA, Vol. 89, pp. 10915-10919, November 1992, Biochemistry.
- [13] Jamal Bzai, et al ;" Machine Learning-Enabled Internet of Things (IoT): Data, Applications, and Industry Perspective ", MDPI ,2022.
- [14] Paul Hodor, et al ;" cl-dash: rapid configuration and deployment of Hadoop clusters for bioinformatics research in the cloud ", OXFORD, 2015.