



رایانش تکاملی

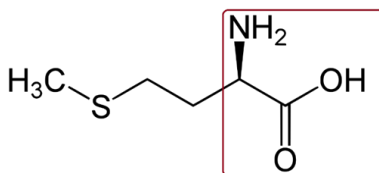
ترم اول ۱۴۰۳ - ۱۴۰۲

تمرین دوم

هدف از این تمرین طراحی یک الگوریتم تکاملی برای حل مسأله تا شدن پروتئین (protein folding problem) است که در آن باید پایدارترین ساختار سه‌بعدی برای پروتئین داده شده را مشخص کرد. مشخصات مسأله و نحوه محاسبه برازندگی یک ساختار سه‌بعدی در ادامه توضیح داده شده است. مواردی که باید در این تمرین تحویل داده شوند در بخش ۴ آورده شده است. مهلت تحویل این تمرین تا پایان روز جمعه ۶ بهمن ۱۴۰۲ خواهد بود.

۱- ساختار یک پروتئین

پروتئین درواقع دنباله‌ای از اسیدهای آمینه (amino acids) است که به صورت سری بهم متصل شده‌اند. بسیاری از ویژگی‌های یک پروتئین که مثلاً در ساخت داروهای جدید مورد توجه است به ساختار سه‌بعدی پروتئین و نحوه قرارگیری دنباله اسیدهای آمینه آن در فضای سه‌بعدی ربط پیدا می‌کند. بنابراین یکی از وظایف مهم در این زمینه پیش‌بینی ساختار سه‌بعدی پروتئین قبل از تشکیل چنین دنباله‌ای از اسیدهای آمینه به صورت واقعی است. در طبیعت حدود ۲۰ نوع اسید آمینه وجود دارد که در ساختار مولکولی آنها بخش مشترکی وجود دارد. به عنوان نمونه، شکل ۱ ساختار مولکولی اسید آمینه متیونین (methionine) را نشان می‌دهد. هر اسید آمینه دارای یک کد اختصاری است که معمولاً در نمایش دنباله اسیدهای آمینه پروتئین مورد استفاده قرار می‌گیرد. برای مشاهده مجموعه اسیدهای آمینه، کدهای اختصاری و ساختار مولکولی آنها می‌توانید به این [سایت](#) مراجعه کنید.



شکل ۱. ساختار مولکولی اسید آمینه متیونین (methionine). بخش مشترک در اسیدهای آمینه در کادر قرمز رنگ مشخص شده است.

ساختار سه‌بعدی پروتئین بر اساس برآیند پیوندهای کووالانسی (covalent) و نیروهای واندروالسی (van der Waals) ایجاد شده بین اتم‌های اسیدهای آمینه تشکیل دهنده پروتئین بدست می‌آید. یکی از مواردی که باعث می‌شود پیش‌بینی نحوه تا شدن پروتئین‌ها پیچیده شود، پیوند اتم‌های اسیدهای آمینه غیر متوالی با همدیگر است (شکل ۲). پیوندهای ایجاد شده بین اسیدهای آمینه معمولاً از طریق پیوند بین اتم‌های اکسیژن (O) و نیتروژن (N) باردار موجود در ساختار مولکولی آنها با سایر اتم‌ها صورت می‌گیرد. اطلاعات دقیق‌تر در مورد جزئیات ساختارهای مولکولی و ساختار سه‌بعدی اسیدهای آمینه متفاوت در قالب پروتئین‌های مختلف را می‌توانید از طریق این [سایت](#) ببینید.



contact in 3D

شکل ۲. پیوند اسیدهای آمینه غیر متوالی در دنباله پروتئین

۲- انرژی یک ساختار سه‌بعدی پروتئین

در دنیای واقعی عوامل متعددی در تعیین ساختار سه‌بعدی نهایی هر اسید آمینه تأثیر می‌گذارد که از آن جمله می‌توان به سایر اسیدهای آمینه تشکیل‌دهنده پروتئین، شرایط محیطی مانند خاصیت بازی (PH) محلول حاوی پروتئین یا میدان مغناطیسی محلی اشاره کرد. در نهایت ساختار سه‌بعدی که دارای مناسب‌ترین انرژی باشد از بقیه ساختارهای ممکن پایدارتر خواهد بود. هدف در این تمرین پیش‌بینی چنین ساختاری برای یک دنباله داده شده از اسیدهای آمینه است. بنابراین باید تغییراتی (مانند ایجاد پیوند، جابجایی، چرخش، ...) که در زنجیره اتمی دنباله اسیدهای آمینه می‌تواند اتفاق بیافتد بررسی شده و وضعیتی با انرژی نزدیک به صفر شناسایی شود.

برای ساده‌سازی تمرین، ساختارهای سه‌بعدی بدست آمده برای اسیدهای آمینه با روش‌های آزمایشگاهی مانند تصویربرداری X-ray^۱ در سایر پروتئین‌ها مورد استفاده قرار می‌گیرد، که این ساختارها از طریق این [سایت](#) (در قالب‌های فایل مختلف) قابل دریافت است. بر این اساس فرض می‌شود ساختار زنجیره اتمی یک اسید آمینه دچار دگرگونی نشده و در فضای سه‌بعدی فقط می‌تواند مانند یک شیء صلب و غیرمنعطف (rigid) در برخی اتم‌هایش با سایر اسیدهای آمینه پیوند بخورد، جابجا شود یا بچرخد. همچنین برای حذف پیچیدگی‌های مرتبط با دانش شیمی مورد نیاز برای شناسایی پیوندهای ممکن بین اتم‌های اسیدهای آمینه مختلف، همه پیوندهای ممکن برای یک دنباله از اسیدهای آمینه در اختیار دانشجویان قرار داده می‌شود. بنابراین وضعیت‌های مختلفی که باید برای تعیین پایداری مورد بررسی قرار گیرد حاصل در نظر گرفتن وقوع پیوندهای ممکن، و میزان جابجایی و چرخش هر یک از اسیدهای آمینه است.

کلاس FitnessFunction در بسته Help که به همراه این تمرین ارائه شده است، حاوی متدهای لازم برای تعیین انرژی هر وضعیت از دنباله اسیدهای آمینه است. در هنگام ایجاد یک شیء از این کلاس، باید اطلاعات ساختاری دنباله اسیدهای آمینه مورد نظر به همراه نام آنها تعیین شود. یک مثال برای دنباله "AFKC" در شکل ۳ نشان داده شده است.

```
with open("AminoAcids/M.json", 'r') as file:
    json_data = file.read()
data = json.loads(json_data)
M_Amino = ReadJsonFile(data)
with open("AminoAcids/F.json", 'r') as file:
    json_data = file.read()
data = json.loads(json_data)
F_Amino = ReadJsonFile(data)
with open("AminoAcids/K.json", 'r') as file:
    json_data = file.read()
data = json.loads(json_data)
K_Amino = ReadJsonFile(data)
with open("AminoAcids/C.json", 'r') as file:
    json_data = file.read()
data = json.loads(json_data)
C_Amino = ReadJsonFile(data)

ResNames = ["Met", "Phe", "Lys", "Cys"]

fitness_obj = FitnessFunction([M_Amino, F_Amino, K_Amino, C_Amino], ResNames)
combin_list, rotat_and_trans_list = fitness_obj.TempInput()
```

شکل ۳. نحوه استفاده از کلاس FitnessFunction برای یک دنباله از اسیدهای آمینه.

¹ X-ray crystallography

همانطور که در مثال بالا دیده می‌شود با فراخوانی متد TempInput دو لیست برگردانده می‌شود. لیست اول (combin_list در مثال بالا) همه پیوندهای ممکن بین اتم‌های اسیدهای آمینه موجود در دنباله را نشان می‌دهد و با صفر مقداردهی اولیه شده است. هر عنصر این لیست مرتبط با یکی از مکان‌های پیوندهای ممکن بین اتم‌ها است و مقدار یک در آن مکان به معنی برقراری پیوند و صفر به معنی عدم برقراری پیوند است. اینکه برقراری کدامیک از این پیوندها منجر به انرژی مناسبتری برای کل دنباله اسیدهای آمینه می‌شود باید در فرآیند بهینه‌سازی مورد بررسی قرار گیرد.

خروجی دوم (rotat_and_trans_list در مثال بالا) لیستی به طول ۶ برابر تعداد اسیدهای آمینه موجود در دنباله است که اگر تعداد اسیدهای آمینه برابر با K باشد ساختاری مشابه زیر دارد:

[ac1_rotX, ac1_rotY, ac1_rotZ, ac1_transX, ac1_transY, ac1_transZ, ..., acK_rotX, acK_rotY, acK_rotZ, acK_transX, acK_transY, acK_transZ]

همینطور که مشاهده می‌شود در این لیست به ترتیب میزان چرخش هر اسید آمینه در امتداد محورهای X، Y و Z و نیز میزان جابجایی آن در راستای محورهای X، Y و Z مشخص می‌شود. این لیست نیز با صفر مقداردهی اولیه شده است و باید در روند بهینه‌سازی مقادیر مناسب برای عناصر مختلف آن پیدا شود.

انرژی یک وضعیت بخصوص برای دنباله اسیدهای آمینه با فراخوانی متد calculateEnergy قابل محاسبه است که در شکل ۴ مثالی از آن نشان داده شده است. دقت کنید که این متد می‌تواند مقادیر منفی نیز برگرداند و وضعیتی پایدارتر است که مقدار انرژی آن به صفر نزدیکتر باشد.

```
combin_list = [0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0]
rotat_and_trans_list = [65, 70, 300, 200, 10, 20, 110, 230, 300, 600, 290, 120,
                        260, 100, 290, 600, 100, 220, 190, 700, 400, 120]
print(fitness_obj.calculateEnergy(combin_list, rotat_and_trans_list))
```

شکل 4. محاسبه انرژی یک وضعیت از ساختار سه بعدی دنباله اسیدهای آمینه

۳ - شناسایی دقیق‌تر پیوندها (اختیاری)

برای تعیین ساختار سه بعدی پایدارتر یک دنباله از اسیدهای آمینه، انتخاب ساختار سه بعدی از هر اسید آمینه که با وضعیت آن اسید آمینه در دنباله پروتئین مورد بررسی همخوانی بیشتری داشته باشد اهمیت زیادی دارد. در این بخش از تمرین دانشجویان باید بجای انتخاب یک ساختار دلخواه از هر اسید آمینه موجود در دنباله اسیدهای آمینه پروتئین مورد بررسی، از بین ساختارهای مختلف موجود برای هر اسید آمینه در این [سایت](#)، ساختاری که منجر به وضعیت پایدارتری برای کل دنباله اسیدهای آمینه می‌شود را پیدا کنند.

بخش دیگر قابل انجام برای پیش‌بینی دقیق‌تر ساختار سه بعدی نهایی پروتئین در نظر گرفتن پیوندهای بین اتم‌های گوگرد (S) است. در صورت وجود اتم‌های گوگرد در زنجیره اتمی اسیدهای آمینه این اتم‌ها نیز می‌توانند با هم تشکیل پیوند داده که در انرژی کل دنباله اسیدهای آمینه تأثیرگذار است. برای اطلاع از مکان‌های ممکن جهت ایجاد پیوند بین اتم‌های گوگرد باید از متد TempInputForSulfur اشیاء کلاس FitnessFunction استفاده کرد (شکل ۵). در این لیست نیز مقدار یک به معنی برقراری پیوند و صفر به معنی عدم برقراری پیوند بین اتم‌های مربوطه است و باید در فرآیند بهینه‌سازی ترکیبی از وجود پیوندها که منجر به وضعیت پایدارتری در دنباله اسیدهای آمینه می‌شود مورد بررسی قرار گیرد. این لیست نیز با صفر مقداردهی اولیه شده است. برای بررسی تأثیر این پیوندها در هنگام محاسبه انرژی باید وضعیت مورد بررسی از این لیست به متد calculateEnergy داده شود. شکل ۵ مثالی از این فراخوانی را نشان می‌دهد.

```
sulfur_combin_list = fitness_obj.TempInputForSulfur()

sulfur_combin_list = [0, 1]
print(fitness_obj.calculateEnergy(combin_list, rotat_and_trans_list,
                                  SulfurCombinations=sulfur_combin_list))
```

شکل ۵. استفاده از لیست پیوندهای ممکن برای اتم‌های گوگرد در محاسبه انرژی دنباله اسیدهای آمینه

۴ – خروجی‌های مورد انتظار

هدف از این تمرین طراحی یک الگوریتم تکاملی است که بتواند برای هر دنباله داده شده از اسیدهای آمینه، ساختار سه بعدی با بیشترین پایداری را پیدا کند. به این منظور باید مؤلفه‌ها و پارامترهای الگوریتم تنظیم شوند یا تطبیق داده شوند. نحوه بدست آوردن یا دلیل انتخاب هر مؤلفه یا مقدار برای پارامترها باید توضیح داده شود. دانشجویان باید موارد زیر را در این تمرین تحویل دهند:

- فایل‌های کد برنامه مورد استفاده برای پیاده‌سازی تمرین در یک پوشه به نام code
- نیازمندی‌ها یا وابستگی‌های نرم‌افزاری برای اجرای کدها به صورت دقیق مشخص شود.
- فایل گزارش با نام Doc.pdf (حتماً در قالب PDF باشد) شامل موارد زیر:
 - هر گونه استفاده از ابزارها یا بسته‌های نرم‌افزاری در پیاده‌سازی باید صراحتاً با ذکر منبع بیان شود.
 - توضیحات و نتایج مرتبط با نحوه انتخاب مؤلفه‌ها و مقادیر پارامترها یا تطبیق آنها
 - هر گونه توضیح اضافی در مورد نحوه انجام تمرین
- فایل‌های کد و گزارش را به صورت یک فایل فشرده در قالب ZIP و با نام EC_HW2_Name_Family در صفحه درس در سایت کوئرا بارگذاری کنید (به جای Name نام و به جای Family نام خانوادگی خود را قرار دهید).

مهلت تحویل تمرین تا پایان روز جمعه ۶ بهمن ماه خواهد بود.

موفق باشید