Multi-RELIEF specificiteitsdetectie met matrix substitutiescores

Sudesh Jethoe

svjethoe@few.vu.nl

1387480

22-06-2009

Samenvatting

We hebben bekeken of we het bestaande multi-RELIEF algoritme zo kunnen aanpassen dat die gebruik maakt van een blosum62 matrix om zo beter functionele sites op te kunnen sporen. Ook hebben we het programma zo aangepast dat deze niet meer gebruikt maakt van een iteratief proces. In plaats daarvan hebben we het algoritme alle sequenties met elkaar laten vergelijken in de verwachting dat dit tot betere resultaten zou leiden. Hierbij hebben we gebruik gemaakt van de programmeertaal Python en de datasets die gebruikt zijn in het originele paper. Na het schrijven van het programma hebben we alle datasets afgedraaid en geplot in ROC-curves. Hieruit bleek helaas dat een volledig algoritme in plaats van een iteratieve niet tot betere resultaten leed. Tevens kwam uit het onderzoek naar voren dat ook het gebruik van een blosum62 matrix in de substitutiemethode niet tot betere en in veel gevallen zelfs slechtere resultaten leed.

Inleiding

In dit onderzoek wordt bekeken of we met behulp van een substitutiematrix beter functiespecifieke sites kunnen vinden in eiwitfamilies dan met de multi-RELIEF methode. Functiespecifieke sites zijn de aminozuren van eiwitten die grotendeels bepalend zijn voor de specifieke werking van een eiwit. Het door middel van laboratoriumonderzoek afgaan van al de aminozuren die een eiwit bevat is een tijdrovend karwij. Het zou dan ook stukken makkelijker zijn als dit met een computerprogramma gedaan zou kunnen worden. Er is al in meerdere onderzoeken geprobeerd om een algoritme te vinden dat deze specifieke sites kan voorspellen. Een van deze algoritmes is het RELIEF-algoritme (Marchiori, Pirovano, Heringa & Feenstra, 2006). Dit algoritme vergelijkt sequenties uit 2 eiwitfamilies met elkaar. Het probeert vervolgens met behulp van de verschillen tussen de sequenties van beide eiwitfamilies de functionele sites te bepalen. Dit algoritme is vervolgens uitgebreid om, om te kunnen gaan met sequenties uit meer dan 2 eiwitfamilies. Het ontstane algoritme is multi-RELIEF genoemd (Ye, Feenstra, Heringa, Ijzerman & Marchiori, 2008). Doordat Multi-RELIEF meerdere eiwitfamilies met elkaar kan vergelijken, zijn functionele sites beter te herkennen. Als bijkomend voordeel wegen fouten door verkeerd ingedeelde sequenties minder zwaar mee in de berekening.

De methode die we nu gaan maken en testen is weer een aanpassing van de multi-RELIEF methode. Hierbij kijken we niet alleen of aminozuren verschillen per site, maar ook wat deze verschillen precies zijn. Door rekening te houden met de specifieke verandering in aminozuur per site hopen we de functionele sites beter te kunnen onderscheiden.

Theorie

Het RELIEF-algoritme is een methode om te voorspellen waar eiwitfamilies sites hebben, die bepalend zijn voor hun specifieke activiteit. Dit wordt gedaan door het scoren van de onderdelen (aminozuren) van eiwitten op mogelijke specifieke functionele activiteit, hierbij krijgen onderdelen van het eiwit die waarschijnlijk nodig zijn voor de specifieke werking van het eiwit een hoge score en onderdelen van het eiwit die waarschijnlijk geen of geen specifieke functie hebben een lage score. Dit werkt als volgt:

Eerst nemen we de eerste sequentie uit de eerste van 2 eiwitfamilies, stel dat deze RRRR is, hier zoeken we dan de dichtstbijzijnde buren bij, waarvan 1 in de eigen eiwitfamilie en 1 in de andere eiwitfamilie. Met buur bedoelen we de sequentie die er het meest op lijkt. Zie tabel 1.

Eiwitfamilie	Sequentie		Afstand		Dichtste buur
C1	1	RRRR			
	2	RARR	0100	1	X
	3	RAGR	0110	2	
C2	4	RAGG	0111	3	X
	5	AFGG	1111	4	

Tabel 1 Voorbeeld met 2 eiwitfamilies van 3 en 2 sequenties

Als de dichtstbijzijnde buren zijn bepaald, worden de bijbehorende afstandsvectoren van elkaar afgetrokken, volgens de formule:

Gewichtenlijst = dichtstebuur_andere_groep – dichtstebuur_zelfde_groep

In dit geval wordt dus sequentie 2 van 4 afgetrokken. Daar komt dan de volgende gewichtvector uit: 0111 – 0100 = 0011. Deze gewichtvector wordt voor alle sequenties in beide groepen berekend en hiervan wordt voor elke eiwitfamilie het gewogen gemiddelde bepaald. De gemiddeldes van beide eiwitfamilies tellen we vervolgens bij elkaar op en hierna bepalen we daar ook weer het gemiddelde van, dit is de RELIEF-score.

Het multi-RELIEF algoritme is ook geschikt om meerdere eiwitfamilies met elkaar te vergelijken. In de originele implementatie (Ye, Feenstra et al., 2008) worden willekeurig gekozen subsets van sequenties van eiwitfamilies met elkaar vergeleken volgens het RELIEF-algoritme. De volgende stap is het bij elkaar optellen van al de RELIEF-scores. Dit gaat zo:

Stel dat we een lijst van sequenties hebben, die onderverdeeld zijn in 4 eiwitfamilies en er is per combinatie van eiwitfamilies een RELIEF-score bepaald, zoals weergegeven in tabel 2. De totaal score wordt dan bepaald door per site (aminozuur positie) eerst te kijken of er positieve scores zijn. Zo ja, dan worden deze bij elkaar opgeteld en hiervan wordt het gemiddelde genomen. Bij site 1 zijn er in totaal 3 positieve waardes, die bij elkaar opgeteld 3 zijn, 3/3=1, dit is dus de uiteindelijke score voor site 1. Als er geen positieve waardes zijn, dan wordt het gemiddelde van alle negatieve waardes genomen en als er geen positieve of negatieve waardes zijn, dan is de uiteindelijke score gewoon 0. Zie ook tabel 2.

Eiwitfamilies	RELIEF score
	s1 s2 s3 s4 s5
C1/C2	-1 0 1 0 1
C1/C3	1 0 1 0 1
C1/C4	-1 -1 -1 0 1
C2/C3	1 0 -1 0 1
C2/C4	0 -1 1 0 1
C3/C4	1 0 0 0 1
Totaal Score	1 -1 1 0 1

Tabel 2 RELIEF score per 2 klasses en de totaal score

De multi-RELIEF methode kan ook rekening houden met de 3D-structuur, hiervoor wordt gebruik gemaakt van lijsten met bekende 3D-informatie van de te bepalen sequenties. Sites die dicht bij elkaar liggen krijgen aan de hand hiervan, na bepaling van de RELIEF-score, bonuspunten.

Methode

Matrix substitutie scores:
Bij deze methode gaan we de gewichten bepalen aan de hand van een substitutiematrix in plaats van de verschillen per site. Het werkt als volgt:
Stel dat we de volgende matrix hebben:

	A	R	N	D
A	4	-1	-2	-2
R	-1	5	0	-2
N	-2	0	6	1
D	-2	-2	1	6

En de sequenties AAAAA en RANDA. De gewichten moeten dan als volgt worden bepaald:
Op positie 1 is de A op positie 1 vervangen door een R, A -> R in de matrix levert een -1 op. Als we dit voor alle posities doen, komen we op de volgende gewichten uit:

Sequentie 1	AAAAA
Sequentie 2	RANDA
Gewichten	-1 4 -2 -2 4

Vervolgens worden deze gewichten volgens de rest van het RELIEF-algoritme afgewerkt om de RELIEF-scores te bepalen. Het multi-RELIEF gedeelte van het nieuwe algoritme is echter aangepast en maakt geen gebruik van willekeurige selecties. In plaats hiervan gaat deze alle sequenties in alle aangeleverde eiwitfamilies af. Ondanks dat dit langer duurt, is dit gedaan om de nauwkeurigheid te vergroten. Tevens krijgen bij de matrixmethode de negatieve RELIEF-scores voorrang op de positieve scores. Dit is gedaan, omdat bij de gebruikte matrix (blosum62, Henikoff, S. & Henikoff, J.G., 1992) uit wordt gegaan van de kans waarbij een aminozuur over het algemeen wordt vervangen door een andere. Als deze kans groter dan willekeurig is, dan is de waarde in de matrix hoog. Is de kans kleiner dan willekeurig, dan is de waarde negatief en als de kans willekeurig is, dan is de waarde 0. Aangezien wij opzoek zijn naar functiespecifieke sites, gaan we er vanuit dat de kans dat de aminozuren op deze sites vervangen worden erg klein is (dit

aangezien ze belangrijk zijn voor de functiespecifieke werking van het eiwit). Daarom moeten we bij het zoeken naar sites kijken naar de laagste scores en niet de hoogste.

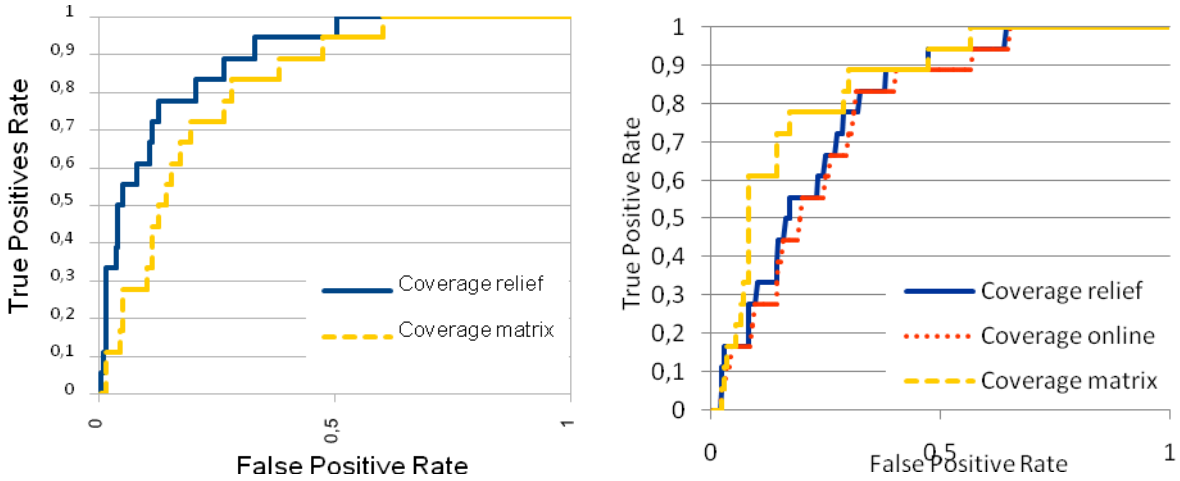
Een derde verandering in het programma is de taal waarin deze geschreven is. Alhoewel de oorspronkelijke multi-RELIEF geschreven was in Octave is er uit praktische overweging besloten om de nieuwe methode in Python te maken. Deze taal is makkelijker te leren, kan beter omgaan met vectoren en wordt het meest gebruikt binnen de vakgroep. Om te bekijken of het nieuwe algoritme beter is, testen we dus 3 programma's, namelijk:

1) online:
   Multi-RELIEF, oorspronkelijke versie, online in Octave
   http://www.ibi.vu.nl/programs/multireliefwww/
   parameters: 1000 iteraties en 100 sequenties per iteratie
2) relief:
   Multi-RELIEF (eigen implementatie in Python)
   parameters: geen
3) matrix:
   Multi-RELIEF (met matrix substitutie scores in Python)
   parameters: de gebruikte matrix is blosum62

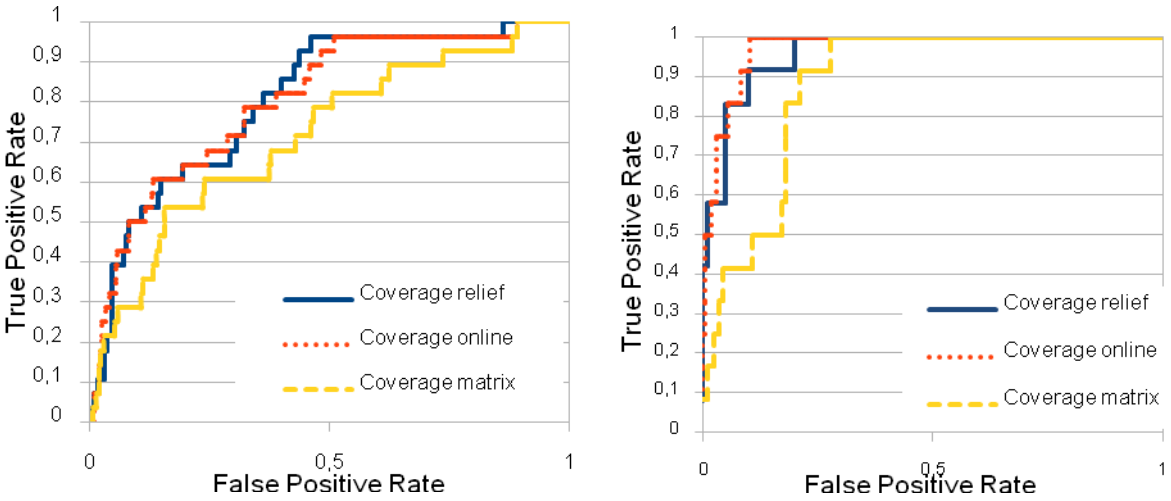
Om te bekijken welke methode beter is maken we gebruik van ROC-plots. In deze plots worden de posities gesorteerd op de berekende waarschijnlijkheid dat dit een functionele site betreft. Vervolgens worden deze posities vergeleken met de lijst met bekende functionele sites. Voor elke treffer neemt de y-waarde toe en voor elke misser neemt de x-waarde toe. Aangezien een goede methode de functionele sites een hogere score (waarschijnlijkheid) geeft, zal de grafiek bij zo'n methode ook sneller het maximum bereiken.

Resultaten

Hieronder voor enkele datasets de ROC-plots:



Figuur 1 ROC-plot van GPCR dataset (links) en GPCR 190 dataset (rechts)



Figuur 2 ROC-plot van LacI dataset (links) en RasRal dataset (rechts)

Wat gelijk opvalt als je naar alle plots kijkt, is dat de matrix-substitutiemethode eigenlijk overal slechter scoort, behalve bij de GPCR-190 set. Ook is te zien dat de online versie en de Python versie ongeveer hetzelfde scoren. Hieronder ook nog voor alle datasets de berekende oppervlaktes van de ROC-curves.

Methode	GPCR	GPCR190	LacI	Rab5/Rab6	Ras/Ral	AQP/GLP	Smad	Gemiddelde
Online		0,75	0,81	0,88	0,97	0,83	0,96	0,867
Relief	0,89	0,78	0,81	0,88	0,96	0,79	0,93	0,858
Matrix	0,82	0,84	0,72	0,86	0,88	0,73	0,93	0,827

Tabel 3 Oppervlaktes onder de grafieken voor alle datasets en methodes (meer is beter)

De GPCR-set kon niet online worden gedraaid, omdat deze te groot was. Hier is dus ook geen data van.

Discussie

Bij de GPCR-190 set zit het verschil er waarschijnlijk in dat er in de normale GPCR-dataset "verkeerde" dichtste buren worden gekozen, waardoor er uiteindelijk verkeerde sites voorrang krijgen. In de GPCR-190 gereduceerde dataset zijn deze "verkeerde" buren waarschijnlijk uit de set gehaald, waardoor er nu betere buren worden geselecteerd. Opvallend is dan echter wel dat dit alleen bij de matrixmethode geldt en dat er bij de multi-RELIEF methode juist slechter gescoord wordt bij een gereduceerde dataset. Bij deze methode zorgt het wegvallen van enkele klassen er dus voor dat de "correcte" sites minder sterk scoren dan de "verkeerde" sites. Dit zou erop kunnen wijzen dat deze methode juist baat heeft bij een zo groot mogelijke dataset. Maar de resultaten zijn natuurlijk ook sterk afhankelijk van de gebruikte matrix en bij het gebruik van een andere matrix zouden deze resultaten misschien wel beter uit kunnen vallen.
Ook is te zien dat de implementatie van het algoritme in Python niet beter (en in veel gevallen zelfs slechter) scoort dan de originele variant. Dit is uiterst vreemd, omdat juist bij deze methode voor alle sequenties de scores worden uitgerekend en niet voor willekeurige selecties een kleine subset, zoals met de online variant gedaan wordt. Waarschijnlijk krijgen de functionele sites door het grote aantal iteraties meer punten en komen ze hierdoor beter naar voren bij de online methode.

Conclusie

In de resultaten is goed te zien dat multi-RELIEF met matrix-substitutiescores, tegen de verwachting in, eigenlijk overal slechter scoort dan de gebruikelijke methode. De Python variant van het algoritme met de aanpassingen scoort over het algemeen hetzelfde als de online variant en soms wat minder goed. Het is dan ook het beste om de online methode zoals deze er nu is, te blijven gebruiken.

Bronnen

Marchiori, E.*, Pirovano, W., Heringa, J. and Feenstra, K.A.* (2006). *A Feature Selection Algorithm for Detecting Subtype Specific Sites for Smad Receptor Binding*, Bio-ICMLA06 (IEEE), 168-173.

Ye, K., Feenstra, K.A., Heringa, J., IJzerman, A..P. and Marchiori, E. (2008). *Multi-RELIEF: a method to recognize specificity determining residues from multiple sequence alignments using a Machine Learning approach for feature weighting*, Bioinformatics, 24(1): 18-25.

Henikoff, S. & Henikoff, J.G. (1992). *Amino acid substitution matrices from protein blocks,* Proc. Natl. Acad. Sci. USA 89, 10915–10919.

www.ibi.vu.nl/programs/multirelief/

www.ncbi.nlm.nih.gov/Class/FieldGuide/BLOSUM62.txt