BD steps

Paolo Prenassi

February 2019

1 BLAST

La nostra proteina è STE che si trova nel file kinase_input_sequences.txt

Cerchiamo le proteine omologhe usando blast dal sito `https://www.ebi.ac.uk/Tools/common/tools/help/index.html` (suggerito da Marco, capire perchè questo sito e non altri)

Parametri:

- WARNING

  In ogni sezione, cliccare su get per inserire i parametri e su Try out per vedere i risultati

- Choose a tool: NCBI BLAS +

  Suggerito da Marco quindi bisogna capire perchè non abbiamo utilizzato gli altri come per esempio PSI BLAST

- Submitjob: mettere la mail, la sequenza e il database

  Noi abbiamo usato Uniref90 perchè suggerito da Damiano. UniprotKb ho dei duplicati mentre uniref90 no. Capire perchè non abbiamo utilizzato altre cose come uniref100

- Status: jobid

  Copiare il jobid Response body nella sezione Submit job

- Dopo l'ultimo Try it out parte il programma e c'è scritto RUNNING in Response body. Quando ha finito c'è scritto FINISHED. Per veirficare che abbia finito cliccare continuamente sul Try it out della sezione Status

- Result types: jobid

  Mettere il jobid di prima e cliccare su try it out di nuovo

- Result: jobid, resultType

  Mettere sempre lo stesso jobtype e in resultType mettere ids per avere la lista degli id. Aprire su un'altra scheda la requestURL e scaricare il file in ids.txt.

2 Multiple sequence alignment

- Eseguire UniprotSequenceRequest.py che prende dal file ids.txt tutti gli id e fa richiesta al db di uniprot per richiedere le rispettive sequenze. Output: sequences.fasta. Lo facciamo perchè su uniref90 non c'è la sequenza completa in generale.
- Utilizzare `https://www.ebi.ac.uk/Tools/msa/clustalo/` per fare MSA. Scegliere il file name.fasta appena creato ed inserirlo in step1. Impostare output format a "ClustalW with carachters counts" (capire perchè usiamo questo e non altri come t-coffee e muscle). Cliccare submit.
- Salvare MSA.clw
- Eseguire msaAdjust.py per togliere le colonne e le righe (ovvero le proteine) con troppi gap. Per fare questo impostare threshold_c e threshold_r che rappresentano rispettivamente la percentuale di gap nella colonna e nella riga considerata. Decidere questi threshold in base ad un criterio che potrebbe essere quanto è buono l'hmm. Questo programma restituisce un file align_cleaned.clw

  In questo passaggio riteniamo più efficace tenere un threshold più stringente sulle righe piuttosto di un threshold stringente sulle colonne perchè abbiamo provato il contrario e sembra che in quel modo venga un hmm troppo overfittato, ovvero butta via tantissime sequenze che non sono STE ma butta via anche molte STE riconoscendole come "NON_STE", cosa che non vogliamo

3 Hidden Markov Models

Da qui in poi serve il programma HMMER. Meglio se utilizzato su Linux perchè su Windows è un casino. Nel caso Marco e Davide sono dei guru e sanno come fare su Windows.

- Eseguire da linea di comando: *hmmbuild Uniprot_Blast_cleaned.hmm align_cleaned.clw*

  Questo comando creerà un file contenente l'hmm che si chiamerà Uniprot_Blast_cleaned.hmm
- Eseguire da linea di comando: *hmmsearch –tblout kinase_hmmer_cleaned.tblout Uniprot_Blast_cleaned.hmm Kinase_dataset.fa > kinase_hmmer_cleaned.ali*

  Questo comando creerà due file. Noi abbiamo utulizzato solo il .ali (perchè? le info sono riassunte e l'altro non lo sappiamo leggere). Questo file contiene le proteine rankate in base agli score.
- Eseguire ste_classifier.py. Questo programma fa variare il threshold sugli score. Se una proteina sta sopra tale threshold viene classificata come STE, altrimenti no. Al variare di esso plotta un grafico con sensitivity, accuracy e specificity, grazie al quale è possibile scegliere scegliere il thereshold di

score ideale. Noi abbiamo scleto come threshold lo score corrispondente al picco dell'accuracy (e al miglior tradeoff per la specificity). Qui c'è un problema perchè la sensitivity cala. Dobbiamo discuterne. Trovato il threshold, questo programma crea il file ste_kin.fasta contenente le proteine classificate come ste nel dataset del prof.

4 Cose da fare

- Clusterizzare le proteine appena trovate utilizzando CD-HIT online e scaricare l'output.
- L'output fa cagare quindi bisogna scrivere un programa che dai nomi delle proteine nei vari cluster tiri fuori a che gruppo appartengono utilizzando il dataset dato dal prof.
- Ripetere l'operazione per trovare la percentuale migliore di identity da impostare facendo cd-hit (un buon guess è 0.3). Per sceglierla dobbiamo tenere in considerazione il fatto che volgliamo meno cluster possibili e vogliamo che in ogni cluster ci siano proteine appartenenti alla stessa famiglia/sottofamiglia (è a questo che serve il programma di prima). Inoltre non volgiamo troppe (ce ne saranno sicuramente) proteine in cluster singoli
- Commentare i cluster così trovati dicendo che abbiamo trovato delle (nuove) sottofamiglie.
- Utilizzare come prima da linea di comando Uniprot_Blast_cleaned.hmm contro tutte le kinesi umane per classificare come STE quelle sopra il threshold degli score trovato prima.
- Clusterizzare queste con CD-HIT urilizzando la stessa percentuale di identity di prima.
- Commentare i cluster così trovati dicendo che abbiamo trovato delle (nuove) sottofamiglie. A differenza di prima non possiamo dire se il clustering è buono o no perchè le proteine utilizzate (quelle scaricate da uniprot), non sono labellate con la famiglia e la sottofamiglia a cui appartengono.
- Ovviamente alcune tra queste proteine ce le avevamo già nel dataset del prof, quindi siamo in grado di labellare i vari cluster utilizzando le informazioni di quelle proteine. Ovvero se in un cluster c'è una proteina che il prof aveva labellato come AGK allora quel cluster "dovrebbe" contenere delle AGK (secondo i parametri che abbiamo scelto).
- Se ci sono dei cluster che non conosciamo (ovvero che non hanno nessuna proteina labellata dal prof), significa che abbiamo una nuova sottofamiglia.

- Per ogni sottofamiglia ricostruire un hidden markov model che identifichi tale famiglia. Si ragazzi! Se ve lo state chiedendo, bifsogna ricominciare quasi da capo con il procedimento. Qui però ho un dubbio. Devo prendere la proteina che meglio mi rappresenta il cluster (quella che secondo cd-hit ha il 100% di idenity, ovvero la prima elencata in ogni cluster) e ripartire dall'inizio, oppure posso prendere le proteine del singolo cluster e fare un hmm con quelle??????????

- Non so in che modo ma si vedrà molto probabilmente che la proteina da cui siamo partiti per questo progetto non è la miglior rappresentante del gruppo. Per vederlo ci sono dei boxplot da fare, ma non so quali (chiedere al magnifico Marco)

- Da qui in poi non so più cosa fare. Forse inizia la seconda parte del progetto (quella con le malattie, ...)

- per trovare malattie, cazzi e mazzi, fare una rischiesta ad uniprot come avevamo fatto all'inizio per trovare la sequenza completa delle proteine.

5 Kinesi umane

Chiedere a Bacco come le ha trovate.

-uniprotKB -fare ricerca selezionando db uniprotKB e non mettendo nulla nella barra. -selezionare human -download file fasta -fatto