CORSO S.I.G.A.

"METODI ED APPLICAZIONI DI INFORMATICA PER L'ANALISI DI DATI BIOLOGICI"

Salsomaggiore Terme, 06-10 Maggio 2013

Il corso verrà tenuto dal Dr. Gian Paolo Ciceri (gp.ciceri@gmail.com).

Breve profilo del docente: laureato in Matematica nel 1991, G. P. Ciceri ha un'esperienza più che ventennale nella gestione di sistemi informativi a fini statistici e nell'analisi dati per la business intelligence. Ha lavorato presso società nazionali ed internazionali di assoluta rilevanza: attualmente si occupa, tra l'altro, di aspetti di CRM (Customer Relationship Management) ed applicazioni di data mining in problemi di Loyalty. Ha collaborato con il Centro di Ricerca per la Genomica e la Postgenomica Animale e Vegetale (CRA) di Fiorenzuola d'Arda per la messa a disposizione di infrastrutture di calcolo parallelo legate a strumenti di Next Generation Sequencing. Ha installato il suo primo Linux nei primi anni novanta (kernel 0.99pl12) e ricompilato il suo primo Python più o meno nello stesso periodo (era la versione 1.2). Mantiene un caleidoscopio informale di materiali e contenuti per il corso - ma senza alcuna pretesa di sistematicità né di esaustività - nel blog: http://ockhamrazorback.blogspot.com.

Durante le esercitazioni dedicate ad esempi applicativi di bioinformatica il docente sarà affiancato dal Dr. Moreno Colaiacovo (CRA-GPG, moreno.colaiacovo@entecra.it). Laureato in bioinformatica, M. Colaiacovo ha conseguito il Dottorato in Sistemi Complessi in Medicina e Scienze della Vita. Attualmente si occupa dell'analisi computazionale delle interazioni tra microRNA e mRNA target in specie vegetali e dell'identificazione di particolari motivi di regolazione (feed-forward loop) in cui intervengono sia fattori di trascrizione che microRNA.

- **Non sono richieste conoscenze pregresse dell'ambiente linux né di linguaggi di scripting.**
- **E' richiesto ai singoli partecipanti di avere con sé il proprio portatile.**
- **I partecipanti avranno la possibilità di utilizzare file di dati ottenuti dalla propria attività di ricerca per imparare ad estrarre le diverse tipologie di informazioni in essi contenute.**

PROGRAMMA

Lunedi pomeriggio: 14-18

0. piattaforme tecnologiche di base

Gli strumenti di base per il bioinformatico: le applicazioni portabili (su chiavetta USB) e le macchine virtuali Linux sotto Windows.

0.1. L'autosufficienza informatica, così come resa possibile con le portableapps

0.1.0. Gestione della piattaforma: aggiornamenti e salvataggi

0.1.1. La configurazione della console portatile

0.2. Il miglior sistema di sviluppo è il proprio: come allestire un (microscopico) supercomputer con Virtualbox ed una macchina virtuale Linux

0.2.0. Interazioni con la macchina virtuale: accesso da rete e condivisione di file con l'ambiente ospite (ospitante la macchina virtuale)

Esercitazione:

a. Configurare le portableapps su una chiavetta USB o su un'unità rimuovibile
b. Installare e configurare Virtualbox e una macchina virtuale Linux Ubuntu
c. Installare Git come sistema controllo versione

Martedi mattina: 9-11

1. La bourne-again shell (bash) di Linux

La classica interfaccia a riga di comando è, per Linux, ancora insostituibile porta d'accesso a tutti i programmi e permette di automatizzare i compiti d'uso più frequenti con estrema eleganza.

1.0. due trucchi del mestiere: la gestione dell'input/output e la concatenazione di programmi

1.1. alcuni programmi notevoli: head, tail, cat, tar, grep - e non solo

1.2. cenno alla programmazione della shell: variabili ed istruzioni condizionali

Martedi mattina: 11-13

Esercitazione

a. Formati standard di file per la gestione di dati di sequenziamento
b. Installazione in locale ed utilizzo del programma BLAST per l'analisi di sequenze

Martedi pomeriggio 14-16

0. Espressioni regolari (regex)

Cercare dati particolari all'interno di un file molto grande e magari di formato complesso è un lavoro molto frequente, che per essere svolto con efficacia richiede la conoscenza di qualche tecnica particolare.

2.0. premessa: grep (wildcards ed espressioni regolari)

2.1. regex: cenni storici ed implementazioni

2.2. esempi di semplici espressioni regolari

2.3. le espressioni regolari in python

2.3.0. vari modi di filtrare i dati: re.match vs. re.search vs. re.sub

Martedi pomeriggio 16-18:

Esercitazione

a. Filtraggio dei risultati del BLAST

Mercoledi 9-11

1. Python

Vero e proprio tuttofare dell'elaborazione dati, il linguaggio di scripting Python coniuga la valorizzazione del tempo speso nell'utilizzarlo con l'eleganza della soluzione prodotta

3.0. cenno storico: l'evoluzione dei linguaggi di scripting

3.1. Python: caratteristiche e funzionalità

3.2. iPython, la potente shell iterativa

3.2.0. il profilo -pylab di iPython

3.3. tipi dati nativi: liste, tuple e dizionari (array associativi)

3.4. lavorare con i file in Python

Mercoledì 11-13

Esercitazione:

a. Utilizzo della shell iPython

b. Primi programmi Python

Mercoledi 14-16

3.5. la tracciabilità del codice e dei programmi: unit testing e logging

3.6. Python nelle scienze, panoramica informativa

3.6.0. utilizzi numerici intensivi e matrici: numpy

3.6.1. Cython: l'alta velocità di python (con i tipi dati del linguaggio C)

3.6.2. database ad elevate prestazioni: pytables

3.6.3. algoritmi per la biologia computazionale: Biopython

3.7.4. interazioni tra Python ed R: rpy2

Mercoledì 16-18

Esercitazione

a. Biopython (prima parte)

Giovedi 09-11

4. Elementi di programmazione parallela

Adesso che anche su di un comune pc portatile ci sono almeno due CPU (virtuali), più frequentemente quattro, può valere la pena imparare come dimezzare (potenzialmente) i tempi di elaborazione

4.0. le moderne CPU permettono di eseguire più di un programma per volta

4.0.0. come struttarle: il modulo Python multiprocessing

Giovedì 11-13

Esercitazione

a.Biopython (seconda parte)

Giovedì 14-16

4.1. algoritmi intrinsecamente paralleli

4.2. introduzione alla programmazione parallela con la tecnica del message passing

4.3. Python in parallelo: una panoramica sul modulo pypar

4.4. due modi di programmare in parallelo: i pattern Master/Worker e MapReduce

Giovedì 16-18

Esercitazione:

a. Istallazione ambiente programmazione parallela
b. Esempio di programma Python parallelo

Venerdi 9-11

5. Programmare (anche) per gli altri

Cenni di programmazione professionale. Quando scrivere un programma per se stessi non è più sufficiente e si desidera che altri usino il nostro software

5.0. la software factory di inizio 21-secolo

5.0.1. tanti dati, e così poco tempo per elaborarli

5.1. alla riscoperta del metodo

5.1.0. tracciabilità totale: controllo versione, unit testing, logging

5.2.0. ancora sull'utilizzo del controllo versione per non perdere traccia dell'evoluzione del codice

5.2.1. utilizzo dello unit testing per migliorare la robustezza del codice e dei dati in input

5.2.2. utilizzo del logging per documentare le esecuzioni

Esercitazione:

a. Esempi di programmi che utilizzano queste tecniche

DISCUSSIONE FINALE E CONSEGNA DEGLI ATTESTATI