PYCON KOREA 2014

IPython으로 상호적, 재현 가능한 생물정보학 연구하기

2014.08.30

김가경

1 게놈 시대에 생물정보학 연구의 필요성 대두

- 1960 년대: 계산 생물학이 대두될 무렵 데이터 세트는 작았으며, 단백질 서열은 수작업을 통해 Dayho 도감으로 인쇄된 후 이후 CD-ROM에 배포되었음.
- 1990 년대: 생물정보학도들이 여러 파지와 세균의 게놈을 포함 점점 더 큰 데이터 세트를 분석하는 스프레드 시트 프로그램과 과학적인 소프트웨어 패키지를 사용 함.
- 2003 년: 인간 게놈의 온라인 공개와 함께 전 게놈 시대가 종료됨. 미국 국립 보건원 (NIH) 시퀀싱 관련 생물에 집중적으로 투자하여 주석에 도움을 줌.
- 2000 년대 중반 : Sanger 시퀀싱은 빠르고 저렴한 차세대 시퀀싱 기술에 의해 대체되어 수천에서 전체 게놈의 정보를 몇 시간에서 수일을 통해 가능하게 함. 생산된 빅 데이터에 대해서 자동화 및 확장성이 용이한 계산 도구를 개발중임.
- 미래: 1,000달러 게놈 시대가 다가옴에 따라, 개인 게놈에 대해 공공 데이터베이스가 범람 할 것이며, 다양한 도구와 함께 분석 할 수있는 풍부한 정보 소스를 제공 할 것임. 또한, 단백질체학, 대사체학, 의료 이미징, 환경 조건 데이터의 통합을 위해 쉽게 사용될 수 있음.

2 재현 가능한 생물 정보학 도구

- 필요성: 데이터 분석의 품질을 효율적으로 제어하고, 공동 작업을 용이하게 하며, 어려운 과제 개발 프로그램 및 파이프라인을 장기적으로 유지. 작동을 잘 하고, 읽기 쉬우며 테스트 하기 쉬운 프로그래밍 방법이 필요함.
- 장점: 에러가 지속적으로 발견되고 수정될 수 있으며, 실험실에서 검증을 용이하도록 함.
- IPython: Python, shell 명령어, R을 지원하며, 서로 다른 프로그램과 플랫폼을 통합하고 복잡한 분석을 처음부터 끝까지 수행하게 하는 환경을 위한 상호적 컴퓨팅 환경

3 IPython notebook

- 2011년 Fernando Perz가 IPython 프로젝트를 시작함.
- 대화형 컴퓨팅과 소프트웨어 개발을 위해 최적의 생산성을 얻도록 설계되었으며 최근의 과학계산용 파이썬 프로그램 중에서도 매우 중요한 도구.
- '편집-컴파일-실행' 방식보다 '실행-탐색' 방식을 장려함. 데이터 탐색, 실험, 오류 판독, 반복, 파라미터 최적화 등을 빠르게 처리할 수 있음.
- 운영체제의 쉘, 파일 시스템과도 잘 통합되어 있으며, 다른 프로그래밍 언어를 서포팅 함 (rmagic plugin, RPy2 library).
- IPython notebook은 전자 노트북과 프로그래밍 환경을 통합하여, 각 단계에서 결과를 보고 시각화하고 저장 할 수 있으며, 노트와 코멘트를 상호적으로 첨가 가능하며 클릭 하나로 전체 파이프라인을 실행할 수 있음.
- 따라서 **생물정보 분석의 생산성을 향상시키고 재현성을 돕는다**.

- See details at http://nbviewer.ipython.org/gist/irobii/014b8aa3574090a0d04a

4 BioPython study

- Python을 이용한 생물정보학 스터디
  - Python Korea 지원
  - 구성원: BT, IT, Designer, 심리학자, 국문학자 (차를 아는자와 운전하는 자와의 만남)
  - 자료: http://biopy.github.io http://www.biopython.net

- 바이오스핀 커뮤니티 출범
  - 바이오 + 뇌과학에 관심이 있는, 실습형 스터디 커뮤니티
  - 커뮤니티: https://www.facebook.com/groups/biospin/
  - 페이지: https://www.facebook.com/biospintalk

5 상호적, 재현 가능한 생물 정보 연구의 예

- Cumulative histogram: 마이크로 RNA의 타겟 사이트 유형에 따른 mRNA 발현 패턴을 Kolmogorov-Smirnov (KS) 테스트와 누적 비율로 도식화 한 예
- 구현: http://nbviewer.ipython.org/gist/irobii/8e96a4834dc6993a25e2
- 방법: Ipython notebook에서 시각화를 위해 matplot library와 과학 연산이 가능한 scipy의 ks_2samp 기능을 사용
- 결론: 기존의 엑셀에서의 작업을 자동화 할 수 있었으며, IPython notebook 내에서 결과 값을 확인하고 시각화가 가능함.

감사합니다

email: kakyung.kim@gmail.com