Sygnatury genowe chorób - użycie równoległego R i uczenia maszynowego z MLInterfaces

Marek Wiewiórka[1], Alicja Szabelska[2], Michał Okoniewski[3]

[1]Politechnika Warszawska, Instytut Informatyki
[2]Uniwersytet Przyrodniczy w Poznaniu, Katedra Metod Matematycznych i Statystycznych
[3]ETH Zurich, Scientific IT Services

PAZUR, 15-17 październik 2014

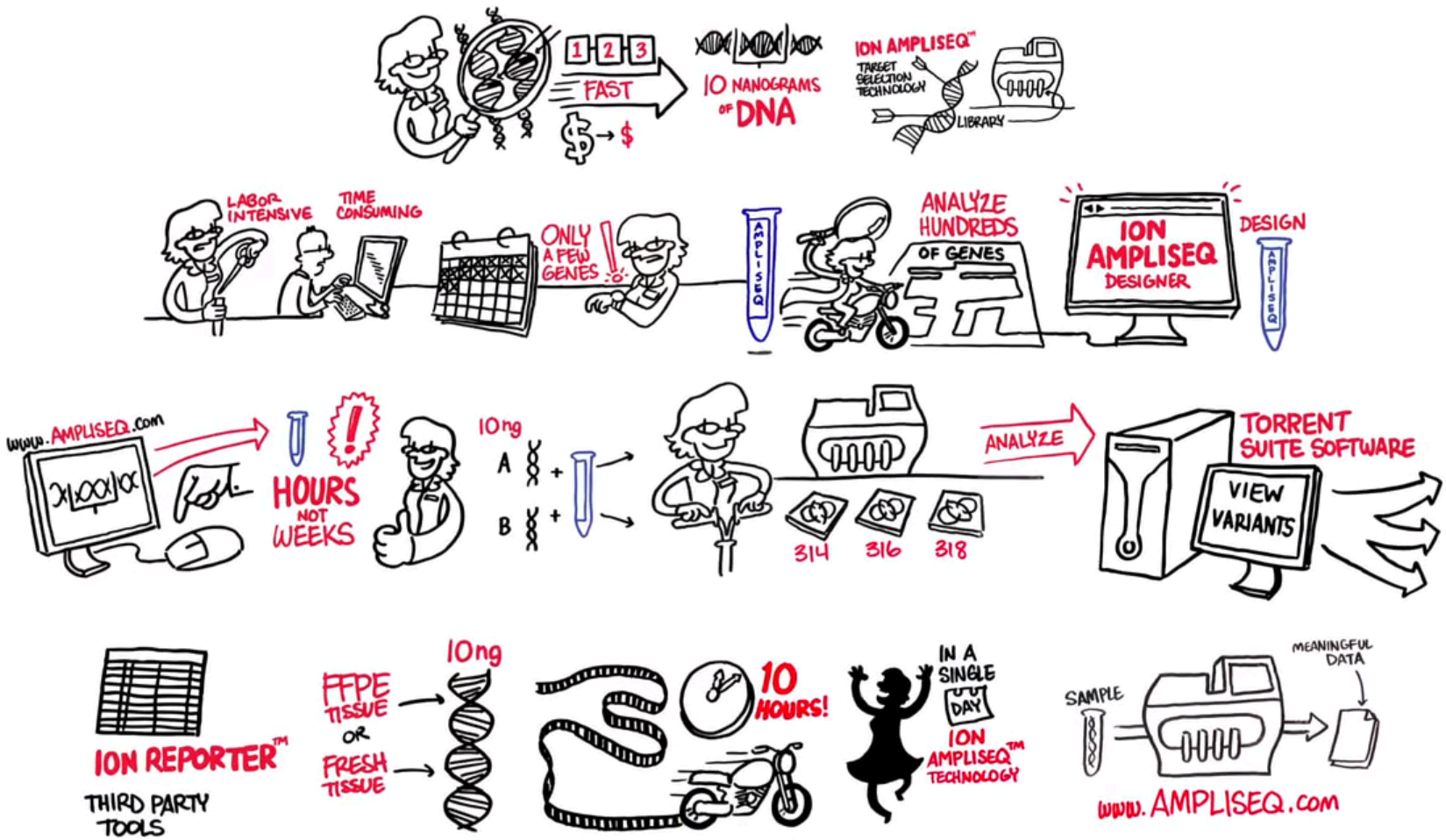
Outline - RNA-Seq

1. Eksperyment
2. Klasyfikacja i uczenie maszyn
3. Wykorzystanie obliczeń równoległych w R

Czego eksperyment dotyczył

- Eksperyment testowy technologii AmpliSeq (LifeTech) - wzbogacanie specyficznych sekwencji RNA przed sekwencjonowaniem (IonTorrent/Proton)
- wzbogacanie odbywa się za pomocą reakcji PCR w typowej maszynie do PCR, które są w powszechnym laboratoryjnym użyciu
- zaprojektowaliśmy prawie 300 amplikonów - krótkich ok 120bp sekwencji z genomu w znaczących miejscach genów odpowiedzialnych za odpowiedź odpornościową organizmu

Technologia AmpliSeq



Zakres eksperymentu

- Próbki biologiczne: krew (PBMC) 8 pacjentów z SM i 8 osób zdrowych odpowiadających wiekiem i płcią
- Sami byliśmy "healthy donors"
- Cel eksperymentu: wykazanie, że za pomocą AmpliSeq można różnicować pacjentów wg choroby i jej typu
- Dziękujemy LifeTech za udostępnienie kitów AmpliSeq (Q1 2013) :)

Wybór zmiennych wykorzystanych w klasyfikacji

- Selekcja istotnych amplikonów na podstawie nieparametrycznego podejścia SAMseq - pakiet samr
- Detekcja SNP jako dodatkowy atrybut do klasyfikacji - kody Marka i pakiet AmpliQueso
- Może jakieś jeszcze charakterystyki warto wziąć pod uwagę?

Analiza dyskryminacyjna

- Weryfikacja klasyfikacji prób do odpowiednich klas na podstawie istotnych amplikonów, informacji o SNP oraz płci
- Predykcja na podstawie wybranych klasyfikatorów:
  - naiwny bayesowski - `naiveBayesI`
  - sieci neuronowe - `nnetI`
  - k najbliższych sąsiadów - `knnI`
  - metoda wektorów nośnych - `svmI`
  - lasy losowe - `RandomForestI`
- Walidacja wyników metodą LOOCV (Leave-one-out cross validation)

Klasyfikacja z MLIterfaces

- Metody uczenia maszyn dla danych w Bioconductorze
- Pakiet wykorzystywany w klasyfikacji i analizie skupień

Funkcja MLearn

```
MLearn(formula, data, .method, trainInd, ... )
```

- formula - formuła R określająca zależność między badaną cechą a pozostałymi zmiennymi
- data - dane zawierające wszystkie zmienne wytypowane do uczenia klasyfikatora oraz badaną cechę
- .method - metoda uczenia maszyn
- trainInd - wektor numeryczny określający próby, które mają być w zbiorze uczącym

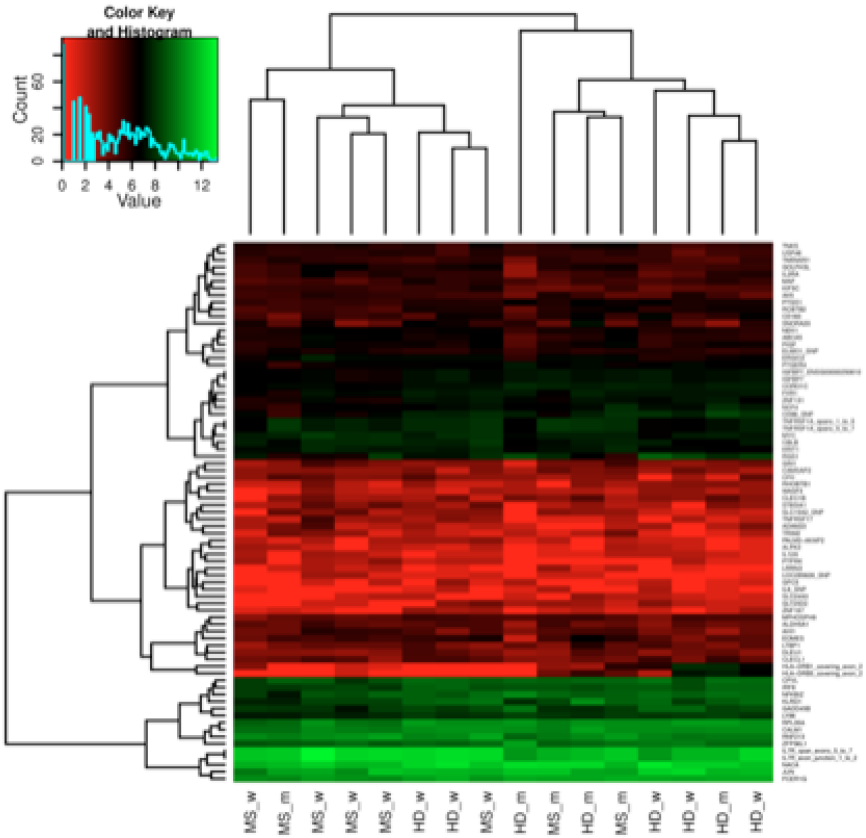
Jak to działa w R?

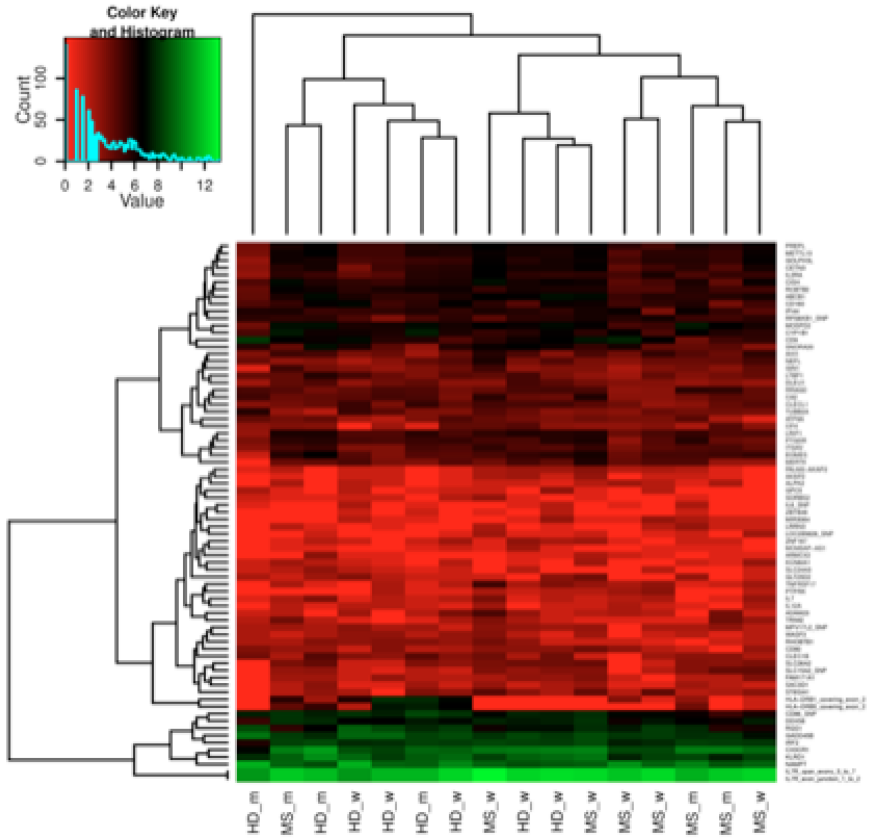
```
library(MLInterfaces)
load("test.Rd")
klasyfikator <- MLearn(klasy ~ ., data=dane, knnI(k=3), xvalSpec("LOO"))
confuMat(klasyfikator)

##       predicted
## given 1 2
##     1 3 5
##     2 7 1

correct.pred <- sum(diag(confuMat(klasyfikator)))/nrow(dane)
correct.pred

## [1] 0.25
```

Wyniki - efektywność klasyfikacji

Zmienne	nB	knn*	svm+	nnet*	rF
100 amplikonów	0.625	0.5	0.4375	0.625	0.56875
100 amplikonów + płeć	0.625	0.5	0.4375	0.625	0.5875
100 amplikonów + płeć + SNPy	0.5625	0.5	0.5	0.625	0.55625
100 amplikonów + SNPy	0.5625	0.5	0.5	0.63125	0.55625
wszystkie amplikony	0.75	0.625	0.75	0.7	0.7
wszystkie amplikony + płeć	0.8125	0.625	0.6875	0.71875	0.725
wszystkie amplikony + płeć + SNPy	0.8125	0.625	0.75	0.71875	0.7125
wszystkie amplikony + SNPy	0.8125	0.625	0.75	0.7375	0.725

Wyniki - mapy ciepła





Obliczenia równoległe w R

- w repozytorium CRAN[1] można znaleźć wiele pakietów umożliwiających wykonywanie obliczeń równoległych i rozproszonych;
- są to zarówno pakiety ogólnego przeznaczenia(np. snow, snowfall itd.), jak do konkretnych zastosowań (np. maanova - analiza danych z mikromacierzy);
- niektóre z nich wymagają zainstalowanych dodatkowych natywnych wersji bibliotek matematycznych(np. Magma, OpenCL, gputools, itd.) i/lub dostępu do dedykowanych koprocesorów arytmetycznych (np. kart graficznych wspierających technologie CUDA/OpenCL)
- historycznie bardzo wiele z nich działało tylko na systemach operacyjnych typu Unix (wykorzystanie mechanizmu rozwidlania procesów – np. parallel, multicore).

[1] `http://cran.r-project.org/web/views/HighPerformanceComputing.html`

Obliczenia równoległe w R

- w repozytorium CRAN[1] można znaleźć wiele pakietów umożliwiających wykonywanie obliczeń równoległych i rozproszonych;
- są to zarówno pakiety ogólnego przeznaczenia(np. snow, snowfall itd.), jak do konkretnych zastosowań (np. maanova - analiza danych z mikromacierzy);
- niektóre z nich wymagają zainstalowanych dodatkowych natywnych wersji bibliotek matematycznych(np. Magma, OpenCL, gputools, itd.) i/lub dostępu do dedykowanych koprocesorów arytmetycznych (np. kart graficznych wspierających technologie CUDA/OpenCL)
- historycznie bardzo wiele z nich działało tylko na systemach operacyjnych typu Unix (wykorzystanie mechanizmu rozwidlania procesów – np. parallel, multicore).

[1] `http://cran.r-project.org/web/views/HighPerformanceComputing.html`

Obliczenia równoległe w R

- w repozytorium CRAN[1] można znaleźć wiele pakietów umożliwiających wykonywanie obliczeń równoległych i rozproszonych;
- są to zarówno pakiety ogólnego przeznaczenia(np. snow, snowfall itd.), jak do konkretnych zastosowań (np. maanova - analiza danych z mikromacierzy);
- niektóre z nich wymagają zainstalowanych dodatkowych natywnych wersji bibliotek matematycznych(np. Magma, OpenCL, gputools, itd.) i/lub dostępu do dedykowanych koprocesorów arytmetycznych (np. kart graficznych wspierających technologie CUDA/OpenCL)
- historycznie bardzo wiele z nich działało tylko na systemach operacyjnych typu Unix (wykorzystanie mechanizmu rozwidlania procesów – np. parallel, multicore).

[1] `http://cran.r-project.org/web/views/HighPerformanceComputing.html`

Obliczenia równoległe w R

- w repozytorium CRAN[1] można znaleźć wiele pakietów umożliwiających wykonywanie obliczeń równoległych i rozproszonych;
- są to zarówno pakiety ogólnego przeznaczenia(np. snow, snowfall itd.), jak do konkretnych zastosowań (np. maanova - analiza danych z mikromacierzy);
- niektóre z nich wymagają zainstalowanych dodatkowych natywnych wersji bibliotek matematycznych(np. Magma, OpenCL, gputools, itd.) i/lub dostępu do dedykowanych koprocesorów arytmetycznych (np. kart graficznych wspierających technologie CUDA/OpenCL)
- historycznie bardzo wiele z nich działało tylko na systemach operacyjnych typu Unix (wykorzystanie mechanizmu rozwidlania procesów – np. parallel, multicore).

[1] `http://cran.r-project.org/web/views/HighPerformanceComputing.html`

Obliczenia równoległe w R

- w repozytorium CRAN[1] można znaleźć wiele pakietów umożliwiających wykonywanie obliczeń równoległych i rozproszonych;
- są to zarówno pakiety ogólnego przeznaczenia(np. snow, snowfall itd.), jak do konkretnych zastosowań (np. maanova - analiza danych z mikromacierzy);
- niektóre z nich wymagają zainstalowanych dodatkowych natywnych wersji bibliotek matematycznych(np. Magma, OpenCL, gputools, itd.) i/lub dostępu do dedykowanych koprocesorów arytmetycznych (np. kart graficznych wspierających technologie CUDA/OpenCL)
- historycznie bardzo wiele z nich działało tylko na systemach operacyjnych typu Unix (wykorzystanie mechanizmu rozwidlania procesów – np. parallel, multicore).

Problemem w takiej sytuacji staje się *przenośność* pakietów, które chcą wykorzystywać obliczenia równoległe!

[1] `http://cran.r-project.org/web/views/HighPerformanceComputing.html`

. . . doParallel[2] – „parallel backend“

- stanowi warstwę abstrakcji dla różnych mechanizmów zrównoleglania;
- w systemach typu Unix wykorzystuje pakiety parallel/multicore, natomiast w Windows snow

```
library(doParallel)
Loading required package: foreach
foreach: simple, scalable parallel programming from Revolution Analytics
Use Revolution R for scalability, fault tolerance and more.
http://www.revolutionanalytics.com
Loading required package: iterators
Loading required package: parallel

#tworzymy ,,klaster‘‘ z 2 rdzeniami/procesorami/malo elastyczne rozwiaznie
cl <- makeCluster(2)
#lepiej uzyc funkcji detectCores() z pakietu parallel
> detectCores()
[1] 8
cl <- makeCluster(detectCores())
registerDoParallel(cl)
stopCluster(cl)
```

[2] http://cran.r-project.org/web/packages/doParallel/index.html

foreach[3] – łatwe zrównoleglanie pętli

```
> system.time(x <- foreach(i=1:10, .combine='cbind') %do% rnorm(10000000))
   user  system elapsed
  7.868   0.140   8.005
> system.time(x <- foreach(i=1:10, .combine='cbind') %dopar% rnorm(10000000))
   user  system elapsed
  1.530   0.586   3.493
```

Bardzo przydatne opcje:

- `.combine` – sposób „sklejania" wyników oblczeń (np. `cbind`,`rbind`)
- `.export` – wyeksportowanie lokalnych zmiennych do równoległych sesji;
- `.packages` – lista pakietów, które zostaną załadowane w każdej równoległej sesji

Przykład:

```
aqSNPL<- foreach( i=1:nrow(covdescT),
.export=c(".callSamPileup","covdescT"),.packages=c("VariantAnnotation","ampliQueso")) %dopar%
{
  .callSamPileup(i, covdescT,minQual,refSeqFile, bedFile  )
}
```

[3] http://cran.r-project.org/web/packages/foreach/index.html

Czego nas to nauczyło

- Można określić ekspresję różnicową dla wielu amplikonów RNA
- Można użyć tej wiedzy do budowania/wykorzystania sygnatur genowych chorób i wariantów chorób
- Ma sens użycie połączonych danych klinicznych, ekspresji genów i SNP
- Biblioteka ampliQueso[4] – analiza danych z małych paneli wzbogacanego sekwencjonowania
- Zrównoleglanie obliczeń w R i użycie klasyfikatorów na ww danych
- Nie pozwolono nam wysłać do recenzji (!) artykułu o eksperymencie - a był napisany ☹

[4] `http://www.bioconductor.org/packages/release/bioc/html/ampliQueso.html`