MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi ISSN : 1978-161X(p); 2477-2550(e)
Volume 12, No. 1 (2020), pp 75-78 DOI : 10.18860/mat.v12i1.8918
Received : January25th 2020; **Accepted** : March 9th 2020 ; **Avalaible Online** : April 5th 2020

Peningkatan Performa Algoritma Resampling Based Clustering

Okta Qomaruddin Aziz

***Abstract*— Clustering is one of powerful technique to find a biological mechanism in gene expression. This technique identify a gene that has same expression. Using bootstrap method we can improve the quality of microarray, thus resampling based clustering (RC) is consider one of the improvement. RC use K-means clustering to determine initial parameter and need thousands of iteration to converge. Performance improvement can be done at preprocess, such as normalization and changing the initial parameter. Normalization can remove or lower the bias in microarray. The result show that normalization can improve the accuracy of RC. In addition, for parameter K, a lower value will lower the accuracy of this RC.**

***Index Terms*— Clustering, Mikroarray, Resampling, Normalization.**

***Abstrak*–- Dalam analisa ekspresi gen, clustering adalah salah satu teknik yang bermanfaat untuk menemukan mekanisme biologis dengan mengidentifikasi gen yang memiliki pola ekspresi yang sama. Untuk meningkatkan kualitas dari microarray gen yang didapatkan, dilakukan replika eksprimen berulang-ulang. Algoritma resampling based clustering memiliki performa yang baik dalam kasus ini. Algoritma ini menggunakan K-means clustering dalam menentukan parameter awal. Algoritma ini memerlukan ribuan iterasi untuk mencapai konvergensi. Peningkatan performa dilakukan pada tingkat praproses, yaitu dengan melakukan normalisasi dan perubahan parameter awal. Normalisasi berperan penting dalam menghilangkan bias pada microarray. Hasil yang didapatkan dari percobaan, disimpulkan bahwa normalisasi dapat meningkatkan akurasi yang dicapai oleh algoritma resampling based clustering. Selain itu didapatkan juga bahwa nilai awal K yang kecil dapat mengurangi akurasi dari algoritma resampling based clustering ini.**

Kata Kunci—Clustering, Mikroarray, Resampling, Normalisasi.

I. PENDAHULUAN

Analisa microarray adalah salah satu jalan utama untuk mempelajari proses biologis yang diatur oleh DNA. Berdasarkan asumsi bahwa gen yang sama fungsinya memiliki pola yang sama pada setiap kondisi pengambilan microarray, analisa cluster telah menjadi pendekatan yang sangat bermanfaat dalam mempelajari proses biologis dengan mengindentifikasi ekspresi gen yang sama dalam percobaan. Validitas dari analisa cluster bergantung pada kualitas dari microarray yang dipakai. Pada banyak kasus, ekspresi gen pada microarray memiliki data yang beragam. Hal ini dikarenakan untuk setiap pengambilan data, bergantung pada kondisi eksperimen, gen spesifik, dan yang lainnya. Oleh karena itu salah satu cara untuk memperbaiki kualitas dari microarray (dengan mengurangi noise), salah satu cara yang paling sederhana ialah dengan mengambil data dengan mengulang eksperimen (replikasi eksperimen). Jikalau diambil asumsi bahwa ekspresi setiap gen pada replika dapat ditukar antar gen yang sama, akan diperoleh segala kombinasi dari replika yang ada dari eksperimen yang sama. Hal ini dapat digunakan untuk mencari cluster yang lebih akurat dengan algoritma resampling based clustering[1].

Sebelum melakukan pencarian cluster dengan algoritma yang diinginkan, preprocess dari data microarray memiliki peranan penting dalam mencari cluster yang lebih akurat. Salah satu preprocessing yang sering dipakai adalah normalisasi microarray. Normalisasi ini mempengaruhi kecepatan konvergensi dan akurasi dari algoritma yang dipakai dalam mencari cluster[2]. Normalisasi microarray bertujuan untuk memperbaiki bias yang terjadi pada ekspresi microarray yang dipakai. Dengan melakukan normalisasi diharapkan cluster yang didapat dari algoritma clustering yang dipakai memiliki akurasi dan kecepatan konvergensi yang tinggi. Pada tulisan ini dilakukan preprocessing (normalisasi) pada data microarray yang selanjutnya akan digunakan dalam algoritma resampling based clustering. Normalisasi yang digunakan ialah normalisasi standar yang paling umum untuk dipakai. Untuk menentukan nilai awal parameter – parameter dari algoritma ini digunakan algoritma K-means. Dalam hal ini kita menentukan banyaknya cluster awal K agar dapat menentukan nilai awal parameter –parameter yang akan digunakan dalam iterasi selanjutnya. Pada tulisan ini dibahas mengenai pengaruh nilai K dan normalisasi terhadap tingkat akurasi dari algoritma resampling based Clustering.

Manuscript received March 22, 2020. This work was supported in part by Informatics Engineering Department of Maulana Malik Ibrahim Islamic State University

Okta Qomaruddin Aziz is with the Informatic Engineering Departement of Maulana Malik Ibrahim Islamic State University , Malang, Indonesia (corresponding author provide phone +6282246995242); email okta.qomaruddin@uin-malang.ac..id

II. METODE

2.1 Normalisasi

Sebelum diproses oleh algoritma Resampling based Clustering (RC), data gen terlebih dahulu dinormaliasasikan. Normalisasi dilakukan dengan menerapkan metode normalisasi standar. Dengan mengekspresikan ulang y_{ijr} yang merupakan nilai dari gen ke –i untuk percobaan ke –j dan replika ke –r sebagai nilai standar

$$y_{ijr*} = \frac{y_{ijr} - \overline{y_i}}{\sigma_i} \quad (1)$$

Dengan $\overline{y_i}$ adalah nilai rata – rata gen ke –i dan σ_i adalah simpangan baku dari gen ke –i. Selanjutnya setelah dilakukan normalisasi untuk setiap nilai gen, diterapkan algoritma RC terhadap data gen yang telah dinormalisasi.

2.2 Algoritma Resampling based clustering (RC)

Misalkan kita memiliki gen sebanyak N yang diukur pada sebanyak J percobaan dan setiap percobaan direplikasi sebanyan R kali. Dinotasikan y_{ijr} adalah nilai dari gen ke –i pada percobaan ke –j dan replika ke –r. Jika diasumsikan bahwa replika antar percobaan tersebut bisa ditukar, maka secara intuisi jalan yang bagus untuk melakukan clustering adalah melakukan clustering secara konsensus dari pengambilan sampel secara bootstrap dari data. Untuk melakukannya misalkan $s_i = (s_{i1}, \dots, s_{ij})$ adalah kombinasi replika dengan s_{ij} adalah indeks replika untuk kondisi percobaan ke –j. Dinotasikan $y_i^s = (y_{i1}^s, \dots, y_{ij}^s)$, dimana $y_{ij}^s = y_{ij_{sij}}$ adalah nilai gene ke -i pada percobaan ke –j dan dari kombinasi s_{ij}. Dinotasikan kembali $s = (s_1, \dots, s_N)$ dan $y^s = (y_1^s, \dots, y_N^s)$. Misalkan c_i adalah indikator cluster untuk gen ke –i dengan $c_i \in \{1, \dots, K\}$ dinotasikan kembali sebagai $c = (c_1, \dots, c_N)$. Untuk menemukan clustering secara konsensus, kita harus menemukan probabilitas sebagai berikut

$P(c_i = k|y) = \sum_{s \in S} \frac{1}{H} E(I(c_i = k)|y^s)$ dimana S adalah semua kombinasi dari replika dan H merupakan kardinalitasnya. Sehingga $P(c_i = k|y^s)$ adalah probabilitas clustering dari semua kombinasi replika. Untuk mengestimasinya kita menggunakan *Resampling based Clustering* (RC) yang diapprokmasikan oleh beberapa langkah berikut. Pertama kita menentukan nilai awal dari parameter $\Psi = (\pi, \theta, \omega, \tau, \nu)$ dengan parameter – parameter tersebut adalah parameter berdasarkan model *Gaussian mixture* [1]. Penentuan nilai awal parameter ini berdasarkan K-means clustering dimana nilai dari K awal kita tentukan. Selanjutnya kita jalankan langkah – langkah secara umum untuk algoritma RC untuk jumlah iterasi yang telah ditentukan. Untuk iterasi ke –t,

Langkah 1

Diberikan $c(t-1)$dan $\Psi(t-1)$, kita secara random mengambil sampel $s(t)$

Langkah 2

Diberikan $s(t)$ sehinggga didapat $y^{s(t)}$ kita mengambil sampel $c(t)$

Langkah 3

Diberikan $c(t)$ dan $y^{s(t)}$ kita mengambil sampel $\Psi(t)$

Penghitungan nilai $c(t), s(t), \Psi(t)$ berdasarkan [1]. Selanjutnya setelah didapat nilai c sebagai indikator cluster dari gen yang ada dilakukan penghitungan *pairwise similarity matrix* berdasarkan nilai c yang didapat untuk setiap gen. Sebagai langkah terakhir dilakukan clustering berdasarkan metode *average linkage*. Kode algoritma RC dalam bahasa R didapatkan dari [1]

2.3 Data

Data yang digunakan dalam penelitian ini dibagi menjadi dua buah macam data yaitu data simulated dan data asli. Data simulated merupakan data yang digenerate sedemikian hingga mencerminkan data gene dengan parameter tertentu. Sedangkan data asli yang digunakan adalah data Yeast Galactose.

2.3.1 Data Simulated

Data Simulated digenerate berdasarkan fungsi

$$y_{ijr} = \sin\left(\frac{2\pi j}{10} - \frac{k\pi}{4}\right) + u_i^k + v_{ij}^k + \epsilon_{ijr}^k \quad (2)$$

dimana

$u_i^k \sim N\left(0, {\sigma_g^2}^k\right)$, $v_{ij}^k \sim N\left(0, {\sigma_c^2}^k\right)$ $\epsilon_{ijr}^k \sim N(0, {\sigma_r^2}^k)$

Data 1 menggunakan parameter

$\sigma_g = 0.00$, $\sigma_c = 0.20$, $\sigma_r = 0.50$

Data 2 menggunakan parameter

$\sigma_g = 0.00$, $\sigma_c = 0.00$ $\sigma_r = 0.55$

Parameter tersebut secara berturut turut adalah standar deviasi dari gen tertentu, kondisi eksperimen tertentu, dan replika tertentu untuk setiap kluster ke –k.

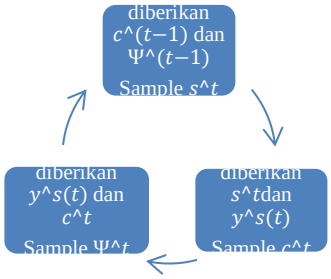
2.3.2 Data Yeast Galactose

Data Yeast Galactose merupakan data yang umum dipakai untuk percobaan algoritma clustering. Data ini sering dipakai karena telah diketahui secara spesifik pembagian cluster untuk setiap gen, sehingga memudahkan evaluasi dari algoritma. Data yang dipakai pada percobaaan kali ini merupakan subset dari data Yeast Galactose yang terdiri dari 205 gen. Data ini dapat diunduh dari website Dr. Yeung (http://expression.washington.edu/publications/kayee/yeungb2003), dimana data yang hilang telah diperbaiki menggunakan KNNimpute. Data ini memiliki level noise yang kecil [1].

2.4 Alur percobaan

Berikut ini adalah alur percobaan yang dibuat

1. Lakukan Normalisasi data dengan menggunakan normalisasi standar
2. Gunakan algoritma K-Means untuk menentukan nilai parameter awal ($\Psi = (\pi, \theta, \omega, \tau, \upsilon)$ dan c). Dimana Ψ adalah parameter parameter berdasarkan model gaussian mixture dan c adalah indikator dari cluster. (Nilai K berubah untuk setiap pengulangan eksperimen)
3. Lakukan iterasi dengan siklis



Gambar 1. Iterasi siklis

Setelah iterasi selesai lakukan hierarchical clustering dengan menggunakan average linkage.

2.5 Evaluasi dan Validasi

Pada eksperimen kali ini digunakan data simulated dan data Yeast Galactose sebagai data microarray yang akan dilakukan clustering. Data ini sama dengan [1]. Evaluasi hasil didapatkan berdasarkan data mengenai microarray yang telah ada dan diketahui detailnya (data Yeast Galactose). Untuk melihat validasi dari cluster digunakan ARI (Advanced Rand Index). ARI nilainya berada pada interval 0 sampai 1 dimana jika nilainya mendekati 1 maka data cluster hasil algoritma RC dan cluster asli pada data relatif sama.

Evaluasi hasil ini sesuai dengan penelitian [1] yang dipakai sebagai acuan utama dari ekperimen.

Untuk melihat validasi dari hasil ARI pada data simulated dan data asli digunakan ukuran standar deviasi (σ) dimana jika nilainya kecil berarti ragam dari hasil percobaan (ARI) untuk nilai parameter awal tertentu setelah pengulangan percobaan, juga kecil. Akibatnya performa dari algoritma tidak berubah – ubah (tetap).

III. HASIL DAN ANALISA

Setelah dilakukan percobaan berulang sebanyak 10 kali untuk tiap tiap nilai awal kluster K, didapatkan hasil sebagai berikut.

3.1 Data Simulated

Iterasi dilakukan sebanyak 1000 iterasi dan nilai K awal dimulai dari 5 sampai dengan 50.

Tabel 1. Nilai ARI untuk Simulated data 1

K	Tanpa	normalisasi
5	0,76 (0,08)	0,82 (0,002)
10	0,93 (0,09)	0,96 (0,08)
15	1 (0)	1 (0)
20	1 (0)	1 (0)
25	1 (0)	1 (0)
30	1 (0)	1 (0)
35	1 (0)	1 (0)
40	1 (0)	1 (0)
45	1 (0)	1 (0)
50	1 (0)	1 (0)

Tabel 2 Nilai ARI untuk Simulated data 2

K	Tanpa normalisasi	normalisasi
5	0,82 (0,002)	0,86 (0,08)
10	1 (0)	1 (0)
15	1 (0)	1 (0)
20	1 (0)	1 (0)
25	1 (0)	`1 (0)
30	1 (0)	1 (0)
35	1 (0)	1 (0)
40	1 (0)	1 (0)
45	1 (0)	1 (0)
50	1 (0)	1 (0)

Nilai dalam kurung () adalah standar deviasi untuk 10 pengulangan. Terlihat bahwa untuk nilai K > 10 ARI yang dicapai adalah 1. Hal ini menunjukkan bahwa cluster yang didapatkan oleh algoritma RC sesuai dengan cluster yang sebenarnya. Terlihat bahwa secara umum dengan menggunakan normalisasi data nilai performa dari RC meningkat. hal ini ditunjukkan dengan peningkatan akurasi (ARI). Selanjutnya juga terlihat bahwa secara umum dengan menggunakan normalisasi performa dari RC akan lebih stabil. Hal ini ditunjukkan dengan nilai standar deviasi yang lebih kecil dibandingkan dengan RC tanpa menggunakan normalisasi. Pada tabel 1 dan tabel 2 terlihat bahwa performa RC akan lebih baik untuk nilai K awal yang lebih besar dari 10.

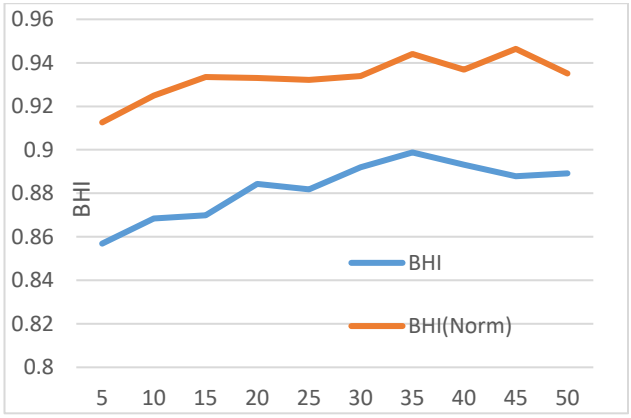
3.2 Data Yeast Galactose

Iterasi dilakukan sebanyak 3000 iterasi dan nilai K awal dimulai dari 5 sampai dengan 50.

Tabel 3. Nilai ARI untuk data Yeast Galactose

K	Tanpa normalisasi	normalisasi
5	0,86 (0,020)	0,91 (0,012)
10	0,87 (0,024)	0,92 (0,011)
15	0,87 (0,025)	0,93 (0,008)
20	0,88 (0.025)	0,93 (0,010)
25	0,88 (0,021)	0,93 (0,015)
30	0,89 (0,020)	0,93 (0,007)
35	0,87 (0,017)	0,94 (0,015)
40	0,89 (0,024)	0,93 (0,015)
45	0,88 (0,023)	0,95 (0,009)
50	0,89 (0,029)	0,94 (0,014)

Nilai dalam kurung () adalah standar deviasi untuk 10 pengulangan. Terlihat bahwa untuk nilai K > 10 ARI yang dicapai lebih baik dibandingkan dengan nilai ARI untuk K<10. Terlihat bahwa secara umum dengan menggunakan normalisasi data nilai performa dari RC meningkat. hal ini ditunjukkan dengan peningkatan akurasi (ARI). Selanjutnya juga terlihat bahwa secara umum dengan menggunakan normalisasi performa dari RC akan lebih stabil. Hal ini ditunjukkan dengan nilai standar deviasi yang lebih kecil dibandingkan dengan RC tanpa menggunakan normalisasi. Jika dibandingkan dengan penelitian [1], iterasi yang dibutuhkan untuk mencapai keakuratan (ARI) sebesar 0,95 adalah 5000 iterasi. Namun dengan menggunakan normalisasi iterasi yang dibutuhkan adalah 3000 iterasi.



Gambar 2 Grafik BHI untuk data Yeast Galactose

RC dengan menggunakan normalisasi sebagai preprocessing dapat meningkatkan performa algoritma baik dalam segi akurasi yang dihitung menggunakan ARI ataupun kestabilan yang dihitung berdasarkan standar deviasi dari akurasi. Hal ini ditunjukkan dengan nilai ARI yang lebih besar dan nilai standar deviasi yang lebih kecil jika dibandingkan dengan RC tanpa menggunakan normalisasi. Namun hal ini belum terbukti untuk data yang memiliki level noise yang tinggi, sebab data yang digunakan pada penelitian ini adalah data yang memiliki noise yang kecil.

IV. KESIMPULAN

Telah dilakukan implementasi RC dengan menggunakan normalisasi sebagai preprocessing dari data. Dari hasil yang diperoleh dapat ditarik kesimpulan bahwa normalisasi meningkatkan performa RC baik dari segi akurasi maupun kestabilan. Patut diperhatikan juga bahwa nilai parameter awal K (banyaknya cluster awal) yang terlalu kecil dapat memperkecil akurasi dari RC. Namun kesimpulan ini tidak dapat dibuktikan pada data dengan level noise yang tinggi, sebab pada penelitian ini digunakan data dengan level noise yang rendah. Untuk selanjutnya Algoritma ini diharapkan dapat diimplementasikan pada level noise yang bervariasi mulai dari data dengan level noise rendah sampai data dengan level noise yang sangat tinggi. Sehingga kita dapat menarik kesimpulan yang lebih umum mengenai pengaruh nilai parameter awal serta normalisasi terhadap performa algoritma RC

REFERENSI

[1] L. Han, L. Chun, H. Jie, F. Xiaodan, “A Resampling based Clustering Algorithm for Replicated Gene Expression Data”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* Vol X no X Februari 2015, DOI: 10.1109/TCBB.2015.2403320, 2015.

[2] H. Jianping, B. Yoganand, C. Yidong, L. James, L. B. Michael, X. Zixiang, S. Edward, R. D Edward, “ Effect of Normalization on Microarray based Classification”, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2006.

[3] A. Sarıkaş, N. Odabaşıoğlu and G. Altay, "Comparison of estimation methods for missing value imputation of gene expression data," 2016 Medical Technologies National Congress (TIPTEKNO), Antalya, 2016, pp. 1-4.

[4] D. Arthur and S. Vassilvitskii. (2007). k means ++ the advantages of careful seeding. in Proc. ACM-SIAM SODA 2007, Soc. Ind. Appl. Math., Philadelphia, PA, USA, 2007, pp. 1027–1035.

[5] Raed Seetan, Jacob Bible, Michael Karavias, Wael Seitan, Sam Thangiah, "Radiation Hybrid Mapping: A Resampling-based Method for Building High-Resolution Maps", *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, pp. 1390.

[6] G. Shao, S. Wu and T. Li, "cDNA microarray image segmentation with an improved moving k-means clustering method," Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), Anaheim, CA, 2015, pp. 306-311.

[7] K. Passi, P. Draper, J. Santala and C. K. Jain, "Microarray Data Analysis of Yeast Data Using Sparse Non-negative Matrix Factorization," 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2017, pp. 1259-1264.