SigWin-detector: A Grid-enabled workflow for discovering enriched windows of genomic features related to DNA sequences

Inda, M.A.; van Batenburg, M.F.; Roos, M.; Belloum, A.S.Z.; Vasunin, D.; Wibisono, A.; van Kampen, A.H.C.; Breit, T.M.

DOI
10.1186/1756-0500-1-63

Publication date
2008

Document Version
Final published version

Published in
BMC Research Notes

License
CC BY

Citation for published version (APA):
Inda, M. A., van Batenburg, M. F., Roos, M., Belloum, A. S. Z., Vasunin, D., Wibisono, A., van Kampen, A. H. C., & Breit, T. M. (2008). SigWin-detector: A Grid-enabled workflow for discovering enriched windows of genomic features related to DNA sequences. BMC Research Notes, 1, Article 63. https://doi.org/10.1186/1756-0500-1-63

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)
**SigWin-detector: a Grid-enabled workflow for discovering enriched windows of genomic features related to DNA sequences**

Márcia A Inda\(^1\), Marinus F van Batenburg\(^2\), Marco Roos\(^3\), Adam SZ Belloum\(^3\), Dmitry Vasunin\(^3\), Adianto Wibisono\(^3\), Antoine HC van Kampen\(^2\) and Timo M Breit*\(^1\)

**Address:** 1Integrative Bioinformatics Unit, Swammerdam Institute for Life Sciences, Faculty of Science, University of Amsterdam, PO Box 94062, 1090 GB Amsterdam, The Netherlands, 2Bioinformatics Laboratory, Academic Medical Center, Meibergdreef 9, 1105 AZ Amsterdam, The Netherlands and 3Institute of Informatics, Faculty of Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

**Email:** Márcia A Inda - inda@science.uva.nl; Marinus F van Batenburg - m.f.vanbatenburg@amc.uva.nl; Marco Roos - roos@science.uva.nl; Adam SZ Belloum - adam@science.uva.nl; Dmitry Vasunin - dVasunin@science.uva.nl; Adianto Wibisono - wibisono@science.uva.nl; Antoine HC van Kampen - a.h.vankampen@amc.uva.nl; Timo M Breit* - breit@science.uva.nl

* Corresponding author

**Abstract**

**Background:** Chromosome location is often used as a scaffold to organize genomic information in both the living cell and molecular biological research. Thus, ever-increasing amounts of data about genomic features are stored in public databases and can be readily visualized by genome browsers. To perform in silico experimentation conveniently with this genomics data, biologists need tools to process and compare datasets routinely and explore the obtained results interactively. The complexity of such experimentation requires these tools to be based on an e-Science approach, hence generic, modular, and reusable. A virtual laboratory environment with workflows, workflow management systems, and Grid computation are therefore essential.

**Findings:** Here we apply an e-Science approach to develop SigWin-detector, a workflow-based tool that can detect significantly enriched windows of (genomic) features in a (DNA) sequence in a fast and reproducible way. For proof-of-principle, we utilize a biological use case to detect regions of increased and decreased gene expression (RIDGES and anti-RIDGES) in human transcriptome maps. We improved the original method for RIDGE detection by replacing the costly step of estimation by random sampling with a faster analytical formula for computing the distribution of the null hypothesis being tested and by developing a new algorithm for computing moving medians. SigWin-detector was developed using the WS-VLAM workflow management system and consists of several reusable modules that are linked together in a basic workflow. The configuration of this basic workflow can be adapted to satisfy the requirements of the specific in silico experiment.

**Conclusion:** As we show with the results from analyses in the biological use case on RIDGES, SigWin-detector is an efficient and reusable Grid-based tool for discovering windows enriched for features of a particular type in any sequence of values. Thus, SigWin-detector provides the proof-of-principle for the modular e-Science based concept of integrative bioinformatics experimentation.
**Findings**

Genomic information is encoded in DNA and as such retained in a fairly steady configuration. In contrast to RNA, proteins, and metabolites, DNA is organized by a limited number of large chromosomes with relatively stable DNA sequences. Therefore, position in the DNA sequence, i.e., chromosome location, provides a convenient and essential scaffold for both the living cell and molecular biological research. In cells, for example, chromosomal organization is important for gene-transcription processes. Expression-profiling studies showed that gene expression is not only controlled at the level of individual genes, but also via autonomous regulation of chromosomal domains [1-5]. This suggests the existence of higher-order transcriptional regulatory mechanisms related to DNA organization or structures. The use of chromosomal organization in the life sciences is exemplified by the popularity of genome browsers that use chromosome location to map many genomic features, such as genes and their products, regulatory elements, gene expression, and epigenetic markers. The search for connections between genomic features is important in unraveling cellular mechanisms.

The pace at which omics experiments continuously keep producing large amounts of data about genomic features for an increasing number of sequenced genomes, creates a need for new high-throughput methods for identification of correlations between DNA related features [6-12]. Therefore, biologists would benefit from tools that could quickly identify enriched regions of genomic features. This would allow extensive, yet convenient in silico experimentation by providing a pipeline for streaming large quantities of data through various algorithms, applications and services.

This paper describes an e-Science based data integration and analysis tool: SigWin-detector. This application can detect clusters with increased (or decreased) density of a genomic feature in a DNA-related sequence in a fast and reproducible way. In the context of the development of a VL, our tool was implemented as a workflow running under WS-VLAM[20,21], a Grid-enabled WMS. A biological use case shows its relevance for biological research. SigWin-detector is based on a method previously used by Versteeg and coworkers [4] to detect regions of increased and decreased gene expression (RIDGEs and anti-RIDGES) in human transcriptome maps (HTM). We improved the original method by i) deriving an analytical formula for computing the new hypothesis probability distribution, which replaces the costly step of estimation by random sampling and ii) developing a new algorithm for computing moving medians. While these improvements radically increase the intrinsic efficiency of the method, implementing SigWin-detector using a generic e-Science approach with access to Grid resources broadens its applicability and makes it amenable to a wide spectrum of experiments on genomic features or in fact on any sequence of values.

**Significant windows and the mmFDR procedure**

Versteeg et al. [4] identified clusters where the median expression level of the genes involved is significantly higher than expected (RIDGEs), using a moving median false discovery rate (mmFDR) procedure (Figure 1). The mmFDR procedure identifies RIDGEs by testing the input gene-expression against the null hypothesis that the position of the genes on the chromosomes does not affect their expression levels. This same procedure can be used to identify significant windows (i.e., windows in the input sequence that have a median value that deviates significantly from expected, if assumed that the ordering of the numbers in the input sequence is random) related to any genomic feature mapped to DNA sequences. In an even wider scope, it can also be used to identify significant windows in any sequence of numbers.

**Avoiding permutations in the mmFDR procedure**

Computationally, the most expensive step in the original mmFDR procedure is the repeated determination of medians over sliding windows of permutations of the input data to estimate the probability function corresponding to the null hypothesis. Our first improvement to the original method was to derive an exact formula for this distribution (see definitions and derivation in Additional file 1):
Figure 1

Using a mmFDR method to detect RIDGEs in a human transcriptome map. Schematic representation of the moving median false discovery rate (mmFDR) procedure identifying regions of high and low density of gene expression (RIDGEs and anti-RIDGEs, respectively) [4]. (A) Input sequence, a human transcriptome map (HTM), i.e., expression values of genes ordered by their chromosome location (cyan; chromosome 6). (B) \( mm(w) \), moving medians of the HTM for a given window size \( S \). (C) Determination of the high and low mmFDR thresholds at a given level \( \alpha \): The high threshold \( m_k \) is the smallest gene expression value for which \( \sum_{m \leq m_k} f(m) / \sum_{m \leq m_k} g(m) \leq \alpha \), here \( f(m) \) is the theoretical probability distribution of \( mm(w) \), and \( g(m) \) is the observed distribution of \( mm(w) \). (In [4], \( f(m) \) is estimated by simple sampling). Similarly, the low threshold \( m_j \) is the largest gene expression value for which \( \sum_{m \leq m_j} f(m) / \sum_{m \leq m_j} g(m) \leq \alpha \). (D) Selection of significant windows in chromosome 6: RIDGEs (in red) all windows for which the median gene expression is higher than or equal to \( m_k \); anti-RIDGEs (in blue) all windows for which the median gene expression is lower than or equal to \( m_j \). (E) Output RIDGEOGRAM of chromosome 6. Each row (y-axis) in the RIDGEOGRAM represents a window size, ranging from \( S = 3 \) to \( S = M \) (the number of genes on the chromosome). Each column (x-axis) represents a sliding window number, ranging from \( w = S/2 \) to \( w = M-S/2 \) (hence the triangular form). Color is used to mark window medians significantly above (red) or below (blue) the genome-wide median. The scheme shows median expression data for window size \( S = 69 \) and FDR thresholds level \( \alpha = 5\% \).
This exact formula reduces the number of cycles of computing moving medians of an input sequence of approximately 25,000 entries from at least 5,000 to 1, giving SigWin-detector the efficiency it needs to be used routinely and for processing and comparing multiple datasets within minutes to hours, instead of days. This efficiency could not be if \( f(m) \) was estimated by sampling the permutation space \( E_N \) and counting the number of times \( m \) was the median value in any sliding window of size \( S \).

\[
\hat{j}(r) = \begin{cases} 
\frac{r-1}{K} \frac{N-r}{S-K-1}, & \frac{N-r}{S-K-1} \\
\frac{N}{S} \frac{N-r}{S-K-1}, & \frac{N-r}{S-K-1} \leq \frac{r-1}{K} \\
\frac{N}{S} \frac{r-1}{K}, & \frac{r-1}{K} \leq \frac{N-r}{S-K-1} \leq \frac{N}{S} \\
\frac{N-1}{S}, & \frac{N-r}{S-K-1} \leq \frac{N-r}{S-K-1} \leq \frac{N}{S} \\
\frac{N}{S}, & \frac{N-r}{S-K-1} \leq \frac{r-1}{K} \leq \frac{N}{S} \end{cases}
\]

This two-phase algorithm is also suitable for parallelization. For example, a two-phase algorithm would start by dividing the input sequence into chunks of size \( 2M \), with \( M \geq 2S_{\text{max}} \) and applying the original algorithm to each chunk separately. Similarly, the second phase computes the medians for the missing sliding windows by dividing the sequence into chunks of the same size, but now using an offset \( M \). This two-phase algorithm is also suitable for parallelization.

\begin{align*}
\text{Additional Figure A1 (Additional file 2) shows a graph comparing our moving medians algorithm with the commonly used Hardle and Steiger's algorithm [22]. While the execution time of their algorithm increases with window size (for a fixed sequence size), the execution time of our algorithm decreases with window size (Figure A1, upper panel). Because SigWin-detector needs to compute moving medians for many window sizes, our algorithm has a clear advantage over Hardle and Steiger's algorithm. In Figure A1, the break-even point of the cumulative computation is for \( S_{\text{max}} \) around 400. The efficiency of our method can be further improved by using a mixed algorithm that uses Hardle and Steiger's algorithm for small window sizes and our algorithm for large window sizes, or by employing a divide-and-conquer approach. For example, a two-phase approach would start by dividing the input sequence into chunks of size \( 2M \), with \( M \geq 2S_{\text{max}} \) and applying the original algorithm to each chunk separately. Similarly, the second phase computes the medians for the missing sliding windows by dividing the sequence into chunks of the same size, but now using an offset \( M \). This two-phase algorithm is also suitable for parallelization.}
\end{align*}

**Bioinformatics application: finding RIDGES in a human transcriptome map**

Once we finished our basic SigWin-detector, we modified it (Additional file 3) for application in our biological use case that aims to find (anti-)RIDGES in transcriptome maps. Figures 4 and 5 show a series of RIDGEOGRAMS for gene expression data for a recent version of the human transcriptome map (HTM) based on the UCSC release hg18 [4], and Table 1 summarizes some RIDGE statistics. Each RIDGEOGRAM displays both RIDGES (red-shades) and anti-RIDGES (blue-shades), the different color shades representing different mmFDR threshold levels. The size of the resulting RIDGEOGRAMS is proportional to the number of genes on a chromosome. We determined i) genome-wide (anti-)RIDGES, i.e., windows for which the median expression is significantly higher (lower) than

First we split the procedure into a collection of workflow components (called modules), each module performing a specific task that may be fine-tuned using parameters. The modules exchange data with each other by means of input and output ports. We then can choose the appropriate modules and compose a workflow suited to our specific needs [16]. Figure 3 describes a basic workflow configuration of SigWin-detector.

The SigWin-detector Config-Basic1 workflow was tested on a Grid computer cluster composed of geographically distributed computational nodes: Distributed ASCI Supercomputer 3 (DAS-3. [23]). Additional Figure A2 (Additional file 2) presents wall clock execution times of the SigWin-detector Config-Basic1 workflow (Figure 3) for input sequences of various sizes.

The basic workflow can be altered by substituting, deleting, or adding modules. For example, we can extend the workflow to get the input sequence from a remote uniform resource identifier (URI) and then put the resulting SigWin-map back into it. We can modify the workflow to generate one SigWin-map per logical subsequence of the input sequence, instead of a single SigWin-map for the complete sequence [16]. We can also expand our workflow by computing significant windows for high median values (e.g., RIDGeS) and significant windows for low median values (e.g., anti-RIDGeS) simultaneously. The SigWin-detector workflow itself can be made into a "composite module" for more complex workflows. Furthermore, interconnection of WS-VLAM with the TAVERNA workbench [19] will permit the use of the existing TAVERNA components in connection with SigWin-detector. At the moment, Grid authentication prevents WS-VLAM workflows being used outside the Grid without the extra step of Grid certification. However, we are working on a Taverna workflow that encapsulates the SigWin detector, to be made available through the myExperiment webpage [24].
**Computing moving medians for many window sizes.** Description of our moving medians algorithm and data structures used. The figure illustrates a computation with input sequence size \( N = 7 \), and window sizes \( S = 3, 5, 7 \). (A) Rank data structure: helps navigation through the sliding windows while keeping track of the median (or any other desired order-statistics). The Rank data structure is a Boolean array used to keep track of the elements that are inside a sliding window by means of crossing out the elements that are outside it. It also has a pointer that keeps track of the \( i \)th remaining element. This pointer is used to track the median. The Marker structure assumes the sequence is in ranked order. For example, if a sliding window of size 3 of a sequence of size 7 contains elements ranked 5, 1, and 6, the corresponding Marker structure has elements ranked 2, 3, and 4 crossed out, and its median pointer points to element ranked 5. (C) Moving median algorithm for window size \( S \). Our algorithm computes the moving medians for window sizes \( S = \text{Smiri}, \text{Smiri}+dS,..., \text{Smiri}+(n\cdot dS) \), starting at \( S = S_{\text{min}} \). When the last sliding window of size \( S \) is reached, the algorithm proceeds to the next window size \( (S+dS) \) by inserting the elements that are in the first sliding window of size \( S \cdot dS \) and crossing out the elements that were in the last sliding window of size \( S \) and setting the new position for the median pointer (which is element \( mm(S+dS) = (S+dS+1)/2 \)). The algorithm stops after computing the medians for the largest window size.

| Window Size | Median Position | Crossing Out | Remaining Elements |
|-------------|-----------------|--------------|--------------------|
| 3           | 1               |              | 10 1 15 1 3 5 20   |
| 5           | 3               | X 1 3 5     | 10 1 15 1 3 5 20   |
| 7           | 4               | X X 1 3 5  | 10 1 15 1 3 5 20   |

### Figure 2

Moving medians algorithm at window size \( S \)

1. Cross out all elements that are not in sliding window \( w=1 \).
2. Set median pointer equal to the remaining element numbered \( mm_S(w)=(S+1)/2 \) (blue arrow).
3. FOR \( w=2 \) TO \( N-S+1 \)
   - Update the Marker structure:
     - insert the element that has entered it
     - cross out the element that has left it.
   - Move the median pointer according to the following rules:
     - Jump up to the next remaining element, if the rank of the element that has left the window is smaller or equal to the pointer value, and the rank of the element that has entered the window is larger than the pointer value.
     - Jump down to the previous remaining element, if the rank of the element that has left the window is smaller than or equal to the pointer value, and the rank of the element that has entered the window is larger than the pointer value.
     - Remain where placed, otherwise.
SigWin-Detector basic workflow using the WS-VLAM workflow composer. Upper: A snapshot of the workflow. Lower: Short description of the functionality of each module, port connections, and output ports. The ports are named by an abbreviation of the module name followed by ‘i’ or ‘o’ (input or output respectively) and the port number. Input ports are colored in blue and output ports in red. The ports are numbered in the same order they appear in the workflow.

| Module functionality | port connections | Output ports description |
|-----------------------|------------------|--------------------------|
| **ColumnReader:** Reads the input sequence $E = \{E_1, E_2, \ldots, E_n\}$ from a selected column of a tab delimited file and transfers it to the output port. | CR1: (Not used) | CRo1: A vector containing the input sequence $E$. |
| **Rank:** Computes the ranks $R = \{R_1, R_2, \ldots, R_n\}$ corresponding to $E$. | R1→CRo1 | Ro1: The Rank structure corresponding to $E$, cf. methods section, and Figure 8. Ro2: A vector containing $R$, a sorted version of $E$. Ro3: A vector containing a sorted version of the non duplicate values of $E$. |
| **SWMedian:** Computes $m_S(w)$, the moving medians of $E$, for window sizes $S = S_{\text{min}}, S_{\text{min}}+\Delta S, \ldots, S_{\text{max}}=S_{\text{min}}+q\Delta S$. Uses the algorithm described in the methods section. | SWM1→Ro1 | SWMo1: The parameters $SW=(N, S_{\text{min}}, S_{\text{max}}, \Delta S)$ corresponding to the sliding window structure. SWMo2: A sliding window structure containing the computed moving medians (i.e., a sequence of vectors. Each containing $m_S(w)$, for $S = S_{\text{min}}, S_{\text{min}}+\Delta S, \ldots, S_{\text{max}}$. |
| **SWMedianProb:** Computes $f_S(m)$, the exact theoretical null hypothesis probability density function corresponding to the moving medians $m_S(w)$ using the analytical formula. | SWMPI1→SWMo1 SWMPI2→Ro2 | SWMPo1: A sequence of vectors. Each containing $f_S(m)$, for $S = S_{\text{min}}, S_{\text{min}}+\Delta S, \ldots, S_{\text{max}}$. |
| **Sample2Freq:** Generates $g_S(m)$, the normalized frequency counts corresponding to the moving medians $m_S(w)$. | S2FI1→SWMo2 S2FI2→Ro3 | S2Fo1: A vector of counts. Each containing $g_S(m)$, for $S = S_{\text{min}}, S_{\text{min}}+\Delta S, \ldots, S_{\text{max}}$. |
| **FDRThreshold:** Uses $g_S(m)$ and $f_S(m)$ to compute $m_{0.05}(or m_{0.1}, \alpha)$, the high (or low) mmFDR thresholds at a given level $\alpha$, corresponding to each window size $S$, for $S = S_{\text{min}}, S_{\text{min}}+\Delta S, \ldots, S_{\text{max}}$. | FDRT1→SWMPo1 FDRT2→S2Fo1 FDRT3→Ro3 | FDRTo1: A sequence of high (or low) mmFDR thresholds $m_{0.05}(or m_{0.1}, \alpha)$, one for each $S$. |
| **SigWinSelect:** Selects the windows for which the median value $m_S(w)$ is above (or below) the FDR threshold $m_{0.05}(or m_{0.1}, \alpha)$. The resulting significant windows are written to a tab-delimited file. | SWSS1→SWMo1 SWSS2→SWMo2 SWSS3→FDRTo1 | SWSo1: Name of the file to which the resulting significant windows were written. SWSo2: (Not used) |
| **SigWinPlotGrace:** Generates an XMGRACE configuration file with instructions of how to plot the resulting SigWin-map. | SWPGI1→SWMo1 SWPGI2→FDRTo1 | SWPGo1: A file containing XMGRACE instructions on how to print the resulting SigWin-map. |
| **XMGrace:** Displays the resulting SigWin-map using XMGRACE. | XMG1→SWSo1 | - |

Figure 3
SigWin-Detector basic workflow using the WS-VLAM workflow composer. Upper: A snapshot of the workflow. Lower: Short description of the functionality of each module, port connections, and output ports. The ports are named by an abbreviation of the module name followed by ‘i’ or ‘o’ (input or output respectively) and the port number. Input ports are colored in blue and output ports in red. The ports are numbered in the same order they appear in the workflow.
expected by considering the whole genome gene expression profile in the mmFDR procedure (Figure 4), and ii) chromosome specific (anti-)RIDGEs, i.e., the same analysis, but considering only the specific chromosome gene expression profile (Figure 5). This distinction has a major effect on the outcome. If the expression values of the genes on a certain chromosome are typically significantly higher than the genome-wide values, then there are less chromosome specific than genome-wide RIDGEs (e.g., chromosome 19 in Figures 4 and 5 and Table 1). Conversely, if the expression values of the genes on a chromosome are typically significantly smaller than the genome-wide values, then there are more chromosome specific RIDGEs (e.g., chromosome 6 in Table 1 and Figures 4 and 5). In the case of anti-RIDGEs the opposite holds (e.g., chromosomes 17 in Table 1 and Figures 4 and 5). This example shows the importance of choosing the right sequence to compute the null hypothesis distribution.

Based on the fact that chromosomes are separate molecules in a cell, one may favor the results from the individual chromosome SigWin-detector analysis to investigate potential higher-order gene expression regulatory mechanisms.

The RIDGEOGRAMS shown in Figures 4 and 5 only take the ordering of the genes into account, and not their actual physical position in the chromosome. However, from a biological perspective it is likely that the higher order gene-expression mechanisms that underlie RIDGEs relate to an actual section of the chromosome rather than a cluster of genes just ordered by their chromosome location. So we used our SigWin-detector to take the physical gene position into account by subdividing the chromosomes in stretches of constant value (250 kb). If a stretch contains the beginning of one or more genes, their average expression value is assigned to that stretch of DNA. For this analysis we used the SigWin-detector Config-Sub2 with preprocessed HTM data and adapted parameters. The resulting RIDGEOGRAMS are proportional to the chromosome's size (Additional Figure A3, Additional file 2).

| chr | median | size | N | gw-R | chr-R | gw-R | chr-R | gw-aR | chr-aR | gw-aR | chr-aR |
|-----|--------|------|---|------|-------|------|-------|-------|-------|-------|-------|
| Y   | 11     | 57772954 | 96 | 0   | 28    | 0    | 9    | 212   | 0    | 54    | 0     |
| 21  | 15     | 46944323 | 318 | 0  | 0     | 0    | 6957 | 0    | 266   | 0     |
| 18  | 16     | 76117153 | 488 | 0  | 8     | 0    | 23329 | 27   | 521   | 0     |
| 13  | 19     | 114142980 | 553 | 0  | 2     | 0    | 32667 | 10123 | 853   | 190   |
| 4   | 23     | 191273063 | 1172 | 0  | 323   | 0    | 121113 | 0    | 5     | 0     |
| 6   | 26     | 170899992 | 1406 | 28327 | 175351 | 873 | 1803 | 73404 | 12884 | 223   |
| 8   | 26     | 146274826 | 1067 | 213 | 61    | 32   | 83720 | 3110 | 176   |
| 10  | 26     | 135374737 | 1123 | 165 | 36813 | 9    | 9239 | 611   | 453   |
| 20  | 26.5   | 62435964 | 738 | 978  | 1350  | 292 | 538 | 4171  | 0     | 7     |
| 2   | 29     | 242951149 | 1908 | 1801 | 22546 | 247 | 52   | 2871  | 376   |
| 5   | 29     | 180857866 | 1276 | 722 | 8875  | 146 | 303 | 35406 | 10231 |
| 3   | 30     | 199501827 | 1581 | 47644 | 97097 | 806 | 1491 | 77920 | 85262 |
| X   | 30     | 154913754 | 893 | 141 | 946   | 106 | 694 | 0     | 3667  |

Table 1: HTM statistical data

chr: chromosome, median: median expression of all genes on a chromosome, size: chromosome size in base pairs, N: number of genes in chromosome, gw-R: number of genome-wide RIDGEs in a chromosome, chr-R: number of chromosome-specific RIDGEs in a chromosome, gw-aR: number of genome-wide anti-RIDGEs in a chromosome, chr-aR: number of chromosome-specific anti-RIDGEs in a chromosome.
The anti-RIDGEs show a lower cut-off caused by the many 0 values in the HTM. The results from the SigWin-detector analysis using chromosome position are substantially different to those using chromosome ordering. This application demonstrated that SigWin-detector is an e-Science tool that allows convenient in-silico experimentation. To prove that this tool is generic, we used our workflow to examine a simple sequential data set: an extended time series of hourly ground level ozone concentration measurements (Additional file 4).

**Availability and requirements**

- **Project name**: SigWin-detector
- **Project home page**: http://mad-db.science.uva.nl/projects/sigwin/
- **Programming language**: C++
- **Other requirements**: SigWin-detector needs the WS-VLAM workflow management system. WS-VLAM has a client distribution and site distribution.

**Authors’ contributions**

MAI carried out the entire research project and wrote the manuscript. MFvB participated in development of the statistical methods. MR was involved in the conceptualization of the analytical formula, in the e-Science approach, and in the coordination of the project. ASZB, DV, and WA...
worked on the development and support of WS-VLAM. AHvK developed the methods for the genomic mapping of expression data and was involved in the development of the statistical methods. TMB conceived the study, participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

Additional material

Additional file 1
Derivation of the exact formula for the probability function \( f(m) \), and detailed description of the mmiFDR-procedure.
Click here for file (http://www.biomedcentral.com/content/supplementary/1756-0500-1-63-S1.pdf)

Additional file 2
Additional Figures.
Click here for file (http://www.biomedcentral.com/content/supplementary/1756-0500-1-63-S2.pdf)

Additional file 3
Description of alternative SigWin-detector workflow configurations.
Click here for file (http://www.biomedcentral.com/content/supplementary/1756-0500-1-63-S3.pdf)

Additional file 4
Applicability of SigWin-detector: periodic time series of air quality data.
Click here for file (http://www.biomedcentral.com/content/supplementary/1756-0500-1-63-S4.pdf)

Additional file 5
This tar file contains the source files of the WS-VLAM modules needed to run the SigWin-detector workflow, and some examples. To uncompress use: \( \text{tar} -xvf \text{SigWin-VLAM.v1.1.tar.gz} \) (Linux users). \( \text{WinZip} \) or a similar tool.
Click here for file (http://www.biomedcentral.com/content/supplementary/1756-0500-1-63-S5.gz)

Acknowledgements

We thank R. Monajemi for assistance with the HTM data sets, R. H. Bischen for checking the mathematics, L. O. Hertzberger for his constant support, and J. Batson for proofreading the paper. This work was carried out in the context of the Virtual Laboratory e-Science project http://www.vl-e.nl and BioRange program of the Netherlands Bioinformatics Centre (NBIC). VL-e is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OCW) and the ICT innovation program of the Ministry of Economic Affairs (EZ). BioRange is supported by a BSIK grant through the Netherlands Genomics Initiative (NGI).

References

1. Spellman PT, Rubin GM: Evidence for large domains of similarly expressed genes in the Drosophila genome. J Biol 2002, 1:5.

2. Boutaeva AM, Kalmykova AI, Shevel'yov Y Y, Nuraminsky DI: Large clusters of co-expressed genes in the Drosophila genome. Nature 2002, 420:666-669.

3. Roy PJ, Stuart JM, Lund J, Kim SK: Chromosomal clustering of muscle-expressed genes in Caenorhabditis elegans. Nature 2002, 418:975-979.

4. Versteeg R, van Schaik BD, van Batenburg MF, Roos M, Monajemi R, Karaoz H, Bussmeijer AH, van Kampen AH: The human transcriptome map reveals extremes in gene density, intron length, GC content, and repeat pattern for domains of highly and weakly expressed genes. Genome Res 2003, 13:1998-2004.

5. Sabo PJ, Kuehn MS, Thurman R, Johnson BE, Johnson EM, Hua C, Man Y, Rosenzweig E, Goldey J, Haydock A, Weaver M, Shafer A, Neri F, Humbert R, Singer MA, Richard TA, O Dorschner M, McArthur M, Hawrylycz M, Green RD, Navas PA, Noble WS, Stamatoyannopoulos JA: Genome-scale mapping of DNase I sensitivity in vivo using tiling DNA microarrays. Nature Methods 2006, 3:S18.
ET, Armengol L, Bird CP, Clark TG, Cooper GM, de Bakker PIW, Kern AD, Lopez-Bigas N, Martin JD, Stranger BE, Thomas DJ, Woodroffe A, Bazzoglou S, Davydov E, Dimas A, Eyras E, Hallgrimsdottir IB, Hardison RC, Huppert J, Sidow A, Taylor J, Trumbower H, Zody MC, Guigo R, Mullikin JC, Abecasis GR, Estivill X, Birney E, Bouffard GG, Guan XB, Hansen NF, Idol JR, Maduro VVB, Maskeri B, McDowell JC, Park M, Thomas PJ, Young AC, Blakesley RW, Muzny DM, Sodergren E, Wheeler DA, Worley KC, Jiang HY, Weinstock GM, Gibbs RA, Graves T, Fulton R, Mardis ER, Wilson RK, Clamp M, Cuff J, Gnerre S, Jaffe DB, Chang JL, Lindblad-Toh K, Lander ES, Koribnine M, Nefedov M, Osoegawa K, Yoshinaga Y, Zhu BL, de Jong PJ: Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature 2007, 447:799-816.

7. Eckhardt F, Lewin J, Cortese R, Rakyan VK, Astwood J, Burger M, Burton J, Cox TV, Davies R, Down TA, Haefliger C, Horton R, Howe K, Jackson DK, Kunde J, Koenig C, Liddle J, Niblett D, Otto T, Peteet R, Sunden S, Thompson C, West T, Rogers J, Olek A, Berlin K, Beck S: DNA methylation profiling of human chromosomes 6, 20 and 22. Nature Genetics 2006, 38:1378-1385.

8. van Steensel B: Mapping of genetic and epigenetic regulatory networks using microarrays. Nature Genetics 2005, 37:518-524.

9. Pollack JR, Perou CM, Alizadeh AA, Eisen MB, Pergamenschikov A, Williams CF, Jeffrey SS, Botstein D, Brown PO: Genome-wide analysis of DNA copy-number changes using cDNA microarrays. Nature Genetics 1999, 23:41-46.

10. Su AI, Cooke MP, Ching KA, Hakak Y, Walker JR, Wiltshire T, Orth AP, Vega RG, Sapinoso LM, Moqrich A, Patapoutian A, Hampton GM, Schultz PG, Hogenesch JB: Large-scale analysis of the human and mouse transcriptomes. Proceedings of the National Academy of Sciences of the United States of America 2002, 99:4465-4470.

11. Yanai I, Benjamin H, Shmoish M, Chalifa-Caspi V, Shklar M, Ophir R, Bar-Evan E, Horn-Saban S, Safran M, Domany E, Lancet D, Shemuel O: Genome-wide midrange transcript expression profiles reveal expression level relationships in human tissue specification. Bioinformatics 2005, 21:850-859.

12. Halasz G, van Batenburg MF, Perusse J, Hua S, Lu XJ, White KP, Bussemaker HJ: Detecting transcriptionally active regions using genomic tiling arrays. Genome Biology 2006, 7:

13. Rauwerda H, Roos M, Hertzberger BO, Breit TM: The promise of a virtual lab in drug discovery. Drug Discovery Today 2006, 11:228-236.

14. Goble C: The low down on e-science and grids for biology. Comparative and Functional Genomics 2001, 2:365-370.

15. Oehmen CS, Stratsmoa TP, Anderson GA, Orr G, Webb-Robertson BJM, Taylor RC, Mooney RW, Baxter DJ, Jones DR, Dixon DA: New challenges facing integrative biological science in the post-genomic era. Journal of Biological Systems 2006, 14:275-293.

16. Inda MA, Belloum ASZ, Roos M, Vasunin D, de Laat CJ, Hertzberger BO, Breit TM: Interactive Workflows in a Virtual Laboratory for e-Bioscience: the SigWin-Detector Tool for Gene Expression Analysis. In Proceedings of the e-Science 2006, Amsterdam IEEE CS Press; 2006.

17. Post LJG, Roos M, Marshall MS, van Driel R, Breit TM: A semantic web approach applied to integrative bioinformatics experimentation: a biological use case with genomics data. Bioinformatics 2007, 23:3080-3087.

18. Ludascher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones M, Lee EA, Tao J, Zhao Y: Scientific workflow management and the Kepler system. Concurrency and Computation: Practice & Experience 2006, 18:i1039-i1065.

19. Hull D, Wolstenen K, Stevens R, Goble C, Pocock MR, Li P, Oinn T: Taverna: a tool for building and running workflows of services. Nucleic Acids Research 2006, 34:W729-W732.

20. Korkhov V, Vasunin D, Wuibsono A, Belloum ASZ, Inda MA, Roos M, Breit T, Hertzberger BLO: VLM-G: Interactive Dataflow Driven Engine for Grid-enabled Resources. Scientific Programming 2007, 15:173-188.

21. WS-VLAM [http://www.science.uva.nl/~gvlam/vlam/]

22. Hardie W, Steiger W: Optimal Median Smoothing. Applied Statistics-Journal of the Royal Statistical Society Series C 1995, 44:258-264.

23. DAS3, The Distributed ASCI Supercomputer [http://www.cs.vu.nl/das3]

24. Goble C, Roure DCD: myExperiment: social networking for workflow-using e-scientists. In Proceedings of the 2nd workshop on Workflows in support of large-scale science: June 25, 2007, Monterey, California, USA ACM Press: 2007:1-2.