White Paper

Hi-C Data Normalization

What is it, and how should you apply it in your Hi-C analysis pipeline?

Introduction

Based on Sanger shotgun sequence data, the draft human genome was first published over two decades ago. Since that time, next generation sequencing (NGS) technologies have emerged and alternative library preparation techniques are available empowering researchers to analyze genomic complexity at ever increasing depths.

As transformational as NGS has been, it has not come without its technological challenges and limitations. The research communities collective experience in over 20+ years of sequencing has uncovered data biases that must be accounted for during our downstream data analysis and interpretation. Systematic challenges and limitations associated with all sequencing platforms include:

- DNA fragmentation bias.
- GC content bias.
- General mappability challenges.

Biological factors also contribute to bias but vary by specific technique.

Hi-C

One notable library preparation technique is Hi-C (https://cantatabio.com/). This method makes use of proximity ligation technology to enable the capture of 3D information alongside primary sequence data thereby offering insights into how DNA physically interacts in three-dimensional space.

Much like standard whole genome shotgun sequencing, Hi-C data captures genetic alterations including single nucleotide variations (SNVs), small insertion/deletions (indels), copy number variations (CNVs), and structural variations (SVs). However, the addition of 3D information has opened the field of 3D genomics and enabled the identification of topological features such as chromosomal territories, active/inactive compartmentalization, topologically associated domains (TADs), and chromatin loops.

Briefly, the Hi-C methodology involves the following five core steps:

1. Chromatin is crosslinked to “lock” chromatin interactions *in situ*.
2. Chromatin is fragmented to create free ends for ligation.
3. Free ends are ligated.
4. A sequencing compatible library is generated.
5. Paired end sequencing is performed on a compatible NGS system.

When mapped back to the reference, the resulting paired-end data contains 3D structural information. Paired-end reads, that may map at a distance from each other in linear sequence space, are indicative of genomic regions found in close proximity in 3D space. This 3D genomic structure is intimately involved in gene regulation, controlling access of the myriad of regulatory elements found in the genome’s “dark matter” to gene promoters.

Hi-C Data Normalization

While Hi-C data shares all the biases inherent to NGS platforms, the data is sufficiently similar that these biases can be accounted for using the standard

Dovetail GENOMICS
Part of Cantata Bio, LLC

approaches developed for shotgun libraries. However, a source of bias unique to this datatype is the exponential decay of interaction frequency with distance between two genomic regions. That is, the closer any two given regions are in genomic coordinate space, the more likely those regions are to from a chimeric ligation product. Therefore, computational tools designed to make insights into 3D genomic structure need to account for this probabilistic dependency. Enter the need for data normalization.

Analysis standards for Hi-C data are still emerging. While numerous procedures have been developed to remove these systematic biases, widespread community acceptance of what should be the "gold-standard" practice for data processing has not yet fully solidified. The consequence for new, or even experienced genomic scientists, is difficulty navigating the many data processing options available.

Available Tools

Available normalization approaches fall into one of two buckets. Initial attempts at correcting for systematic biases used *explicit* approaches. Designed to directly account for each individual source (GC, fragmentation, mappability, enzyme cut sites etc.), explicit probabilistic models account for the expected interaction frequency between any two pairs of reads. Since explicit approaches assume these biases are known upfront and are accurately accounted for when calculating the correction model, they rely on the main sources of bias being well understood *a priori*.

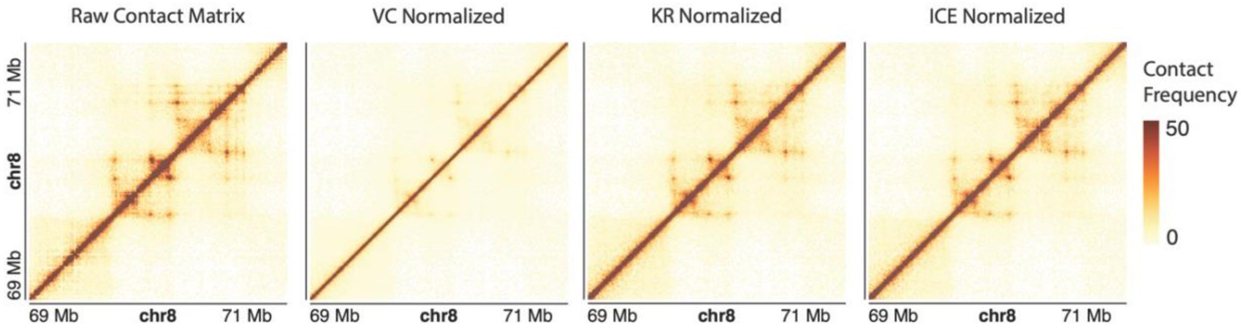
> There are two main options for explicit normalization
>
> - The initial probabilistic model introduced by Yaffe and Tanay[1].
> - An algorithm built upon the same principles called HiCNorm[2].

In contrast, *implicit* models attempt to overcome the need for *a priori* knowledge by making use of an assumption referred to as "equal loci visibility". In essence, implicit approaches assume that any cumulative bias would be captured directly within the sequencing depth of each bin of the contact matrix.

> Examples of the implicit methods are those derived from century-old "matrix balancing" solutions such as Sequential Component Normalization (SCN)[3], Iterative Correction and Eigenvector decomposition (ICE)[4] and the Knight and Ruiz (KR)[5] method. In contrast, chromoR[6] uses a Bayesian approach, and Binless[7], a relatively new algorithm, offers a hybrid approach, where matrix balancing is performed but instead of assuming equal loci visibility, it attempts to build a background correction model followed by negative binomial regression. Other simpler implicit algorithms are vanilla coverage (VC; and the square root supplement – VCSQ), however, these more basic algorithms appear to overcorrect and are not widely used.

So what does this mean for a researcher today preparing their own analysis plan? Which method should they choose? How does this influence what processing pipeline should be chosen?

Practically speaking, the primary difference between the two approaches is the number of user defined parameters – explicitly calculated models require more input parameters (i.e. the list of mapping



Visual Comparison of Normalization Results. Each contact matrix depicts the same 2Mb region on chromosome 8 of a Micro-C library that was sequenced with 800 million read pairs. The matrix was subjected to different normalization approaches and plotted in R. The scale bar is consistently maintained across each normalization approach to better visualize the impact of normalization. This image clearly demonstrates the challenges associated with using coverage alone as a normalization strategy, whereas the more iterative approaches yield a clearer picture of chromatin interactions.

qualities and/or cut sites) than implicit approaches offering greater "tuneability". This makes them more appropriate for less studied organisms where the main sources of biases are likely unknown or poorly described. However, the simplicity of implicit approaches is a benefit of implicit approaches when working with well-studied genomes such and human and mouse making them the *de facto* standard for most Hi-C work.

Thankfully, it appears the field has largely settled on three main data pre-processing pipelines and these pipelines have standardized two data output file formats (i.e. the interaction frequency matrix integral to any Hi-C dataset).

Juicer[8] is an all-in-one pre-processing pipeline developed by the Aiden lab. The main output is a *.hic file. This file is a compressed storage format of the interaction matrices binned at multiple resolutions. A number of downstream computational tools support *.hic files as input and perform the common 3D genomic analyses mentioned above (TADs, loops, etc.). This format also feeds into various visualization tools making it a common choice among 3D genomics researchers.

Cooler[9], a more recent entrant, was developed to make use of the HDF5 data structure. This format offers computational advantages with respect to data structure flexibility. The main output of the cooler pipeline is a *.cool file (and optionally a multidimensional version – *.mcool), and is becoming increasingly adapted for downstream analyses but is not as broadly accepted as the *.hic format.

HiCExplorer[10] is suite of tools which provides a solution for end-to-end Hi-C data analysis and supports both its native matrix format, as well as cool files making it a good option for downstream analyses.

All the above processing pipelines offer multiple options for normalization. As for the most appropriate algorithm to choose, many groups have performed a head-to-head analysis to answer this question. And the answer has generally been...unclear.

One of the first comparisons was performed in a landmark study of the human 3D genome[11] where the authors demonstrated a high concordance for loop and TAD calls no matter what algorithm was used. Ultimately, Rao *et al.* was the first to implement the **KR** method for Hi-C data, having been selected mainly because it was computationally faster than other methods. However, a limitation to the **KR** method is that often fails when the contact matrix is too sparse.

For sparce matrixes, the **ICE** method – which ensures convergence – can be used. Using a robust balancing method, **ICE** does not fail even for sparse matrices but does suffer a reduction in performance in the 1M-500K resolution range.

Another recent study compared six normalization algorithms and came to similar conclusions, noting only minor differences between the algorithms' performance at different resolutions. **SCN** and **KR** was noted to perform admirably for reproducibility of TAD structure. **ChromoR**, while providing a high correlation between technical replicates, appears to break down at lower resolutions (~1M).

Our conclusion… **SCN**, **KR**, and **ICE** (matrix balancing) strategies all perform similarly with only minor differences at the low-resolution range.

Both **KR** and **ICE** are common methods provided internally for most analysis suites. My main choice defaults to **KR** simply because it is offered internally by the **cooler** tools suite, and implemented with a solution for when convergence fails in the sparse matrix scenario. Indeed, the field is still advancing rapidly, and newer algorithms (for example **Binless**) may ultimately shift the consensus in the future once more studies are published so stay tuned.

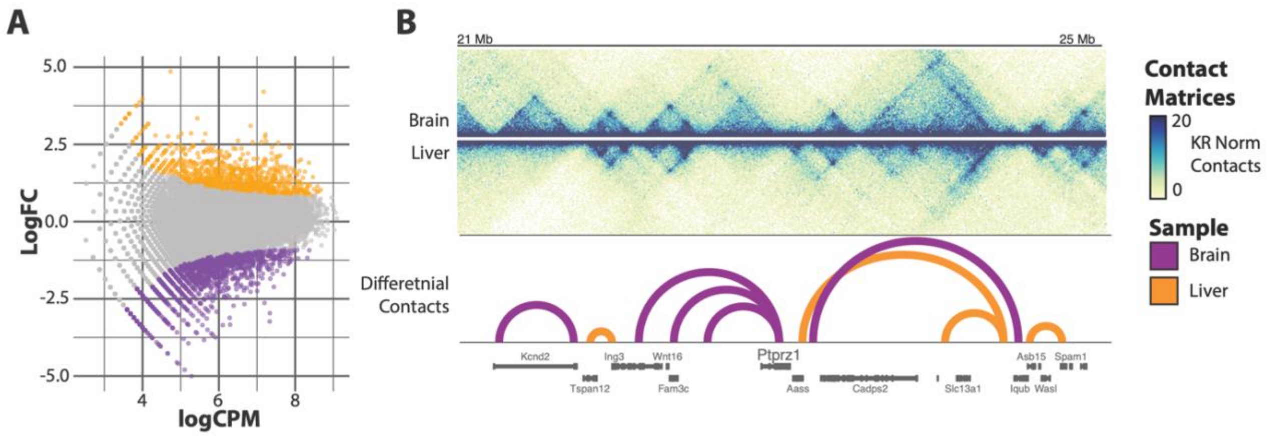
Comparing Two or More Conditions

So far, our discussion has only considered the scenario of normalization *within samples*. However, biological insights are rarely made by analyzing samples in isolation. A well-constructed experiment often includes some form of a control, or more generally, makes a comparison *between* two or more samples or group of samples, as these differences are what often expose meaningful biology. Examples of common types of comparisons could be:

1) Wild-type vs. knockout
2) Drug treated vs. untreated sample
3) Disease vs. normal tissue
4) Within-disease subgroup A vs. subgroup B
5) Time course

For our purpose, we can adapt techniques developed for differential analysis of gene expression. Like Hi-C data, RNA sequencing experiments are predicated on read *abundance* measurements, making it not only important to scale each individual sample by raw sequencing depth, but also important to remove biases *between* samples, such that between sample comparisons can be computed robustly.

Currently, there are only a few methods specifically designed for between sample comparisons for Hi-C data. **MultiHiCcompare**[12] uses a **loess** regression approach, with concepts



Multiple-sample normalization enables robust statistical detection of differential interactions. **A)** By applying concepts from gene expression, in this case, loess normalization, we can detect differences between brain and liver tissues using standard statistical models. **B)** When visualizing these interactions, the resulting differential contacts mirror what we can visually detect in the contact matrices.

largely borrowed and adapted from experiences in the gene expression literature. A more recent approach was developed – **BNBC**[13] – which performs matrix smoothing on individual matrix "bands" prior to batch correction using **ComBat**[13]. Irrespective of the approach, an important consideration with between-sample normalization approaches is that they largely ignore the systematic, sequence-dependent biases such as mappability and GC content, as they are assumed to be common amongst all sample conditions.

What does this mean practically when considering a Hi-C analysis workflow? Does one need to create both within-sample normalized matrix files in addition to the unnormalized, raw interaction frequency matrices? In other words, should we normalize both individually as well as between samples for differential analyses?

Conveniently, the answer to this question is the same whether using *.hic files or *.cool files as the matrix format of choice – **NO!**. One of the features of the multidimensional matrix formats is that, when normalization is performed, the correction weights are stored *independently* of the raw interaction count for each bin. The benefit is that during the processing steps of your Hi-C pipeline, any normalization applied to the contact matrix is calculated "on-the-fly" during downstream feature calling, thus preserving the raw counts. Therefore, one may generate a single, normalized contact matrix and still access the raw counts needed for current between-sample analysis tools.

To our knowledge to date, no comparison has been performed to determine the most robust between-sample normalization approach. However, our internal testing has demonstrated acceptable results when using **multiHiCcompare**.

Take homes

So what have we learned?

1. The most used Hi-C analysis workflow pipelines have built-in methods for individual sample matrix balancing.

2. The **KR** method, common to both **juicer** and **cooler** tools, provides a fast, robust method for individual feature calling that appears to preserve topology.
3. Between sample normalization methods are primarily used for differential region detection and implemented when comparing different sample conditions
4. Between sample normalization depends on raw, unnormalized interaction frequencies, but are readily accessible from both *.hic and *.cool files.

Hopefully, now armed with this knowledge, you will be well on your way towards selecting the analytical pipeline that best fits your needs. Should you still have questions, however, please feel free to reach out to our support team (support@cantatabio.com) for further guidance.

Citations:

1. Yaffe, Eitan, and Amos Tanay. "Probabilistic Modeling of Hi-C Contact Maps Eliminates Systematic Biases to Characterize Global Chromosomal Architecture." *Nature Genetics*, vol. 43, no. 11, 16 Oct. 2011, pp. 1059–1065, https://doi.org/10.1038/ng.947. Accessed 28 Jan. 2021.
2. Hu, Ming, et al. "HiCNorm: Removing Biases in Hi-C Data via Poisson Regression." *Bioinformatics*, vol. 28, no. 23, 27 Sept. 2012, pp. 3131–3133, https://doi.org/10.1093/bioinformatics/bts570. Accessed 18 Sept. 2022.
3. Cournac, Axel, et al. "Normalization of a Chromosomal Contact Map." *BMC Genomics*, vol. 13, no. 1, 30 Aug. 2012, https://doi.org/10.1186/1471-2164-13-436. Accessed 20 Dec. 2022.
4. Imakaev, Maxim, et al. "Iterative Correction of Hi-C Data Reveals Hallmarks of Chromosome Organization." *Nature Methods*, vol. 9, no. 10, 2 Sept. 2012, pp. 999–1003, https://doi.org/10.1038/nmeth.2148. Accessed 6 Jan. 2022.
5. Knight, Philip G, and Daniel Ruiz. "A Fast Algorithm for Matrix Balancing." *IMA Journal of Numerical Analysis*, vol. 33, no. 3, 1 July 2013, pp. 1029–1047, https://doi.org/10.1093/imanum/drs019. Accessed 21 May 2023.
6. Yoli Shavit, and Píetro Lió. "Combining a Wavelet Change Point and the Bayes Factor for Analysing Chromosomal Interaction Data." *Molecular BioSystems*, vol. 10, no. 6, 1 Jan. 2014, pp. 1576–1585, https://doi.org/10.1039/c4mb00142g. Accessed 26 Oct. 2023.
7. Spill, Yannick G., et al. "Binless Normalization of Hi-C Data Provides Significant Interaction and Difference Detection Independent of Resolution." *Nature Communications*, vol. 10, no. 1, 26 Apr. 2019, https://doi.org/10.1038/s41467-019-09907-2. Accessed 28 Jan. 2021.
8. Durand, Neva C., et al. "Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments." *Cell Systems*, vol. 3, no. 1, 1 July 2016, pp. 95–98, www.ncbi.nlm.nih.gov/pmc/articles/PMC5846465/, https://doi.org/10.1016/j.cels.2016.07.002. Accessed 26 Feb. 2021.
9. Abdennur, Nezar, and Leonid A Mirny. "Cooler: Scalable Storage for Hi-C Data and Other Genomically Labeled Arrays." *Bioinformatics*, 10 July 2019, https://doi.org/10.1093/bioinformatics/btz540. Accessed 30 Nov. 2020.
10. Ramírez, Fidel, et al. "High-Resolution TADs Reveal DNA Sequences Underlying Genome Organization in Flies." *Nature Communications*, vol. 9, no. 1, 15 Jan. 2018, https://doi.org/10.1038/s41467-017-02525-w. Accessed 23 Aug. 2022.
11. Rao, Suhas S.P., et al. "A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping." *Cell*, vol. 159, no. 7, Dec. 2014, pp. 1665–1680, https://doi.org/10.1016/j.cell.2014.11.021.
12. Stansfield, John C, et al. "MultiHiCcompare: Joint Normalization and Comparative Analysis of Complex Hi-C Experiments." *Bioinformatics*, vol. 35, no. 17, 22 Jan. 2019, pp. 2916–2923, https://doi.org/10.1093/bioinformatics/btz048. Accessed 26 Oct. 2023.
13. Kipper Fletez-Brant, et al. "Removing Unwanted Variation between Samples in Hi-C Experiments." *BioRxiv (Cold Spring Harbor Laboratory)*, 6 Nov. 2021, https://doi.org/10.1101/214361. Accessed 8 Sept. 2023.