

Supplemental Data

Supplemental Data Inventory

Supplemental Methods and References.

Figure S1. Illustration of Module generation

Figure S2. CNV and Expression of *RHPN2* in GBM samples from TCGA and Rembrandt databases.

Figure S3. Western Blot and qPCR of *RHPN2* in different cell lines.

Table S1. List of Primers

Table S2. Drivers identified by Multi-Reg

Supplemental Methods

Principles of the Multi-Reg algorithm

A key challenge in identifying driver genes from DNA copy number is that amplification and deletions frequently involve large regions of DNA, each consisting of multiple genes. To pinpoint the driver genes within such genomic regions, we previously developed CONEXIC (1), a computational algorithm that integrates copy number and gene expression data, to identify driver genes and connect these to their expression signatures. A key limitation to our previous approach is that CONEXIC can only identify the one dominant driver for each expression signature.

However, multiple drivers can contribute to the same effect, sometimes acting in parallel. For example, almost all GBM patients have activated RTK signaling and disrupted p53/RB signaling, but each patient has a different combination of deletions, amplifications and mutations in some of the many genes known to influence these signaling pathways (2). For example, loss of function (deletion/under-expression) of some driver genes or gain of function (amplification/over-expression) of other driver genes can result in the same phenotype and expression signature.

Multi-Reg is a significant advance compared to existing methods of network analysis, since it seeks **multiple regulators** for each phenotype. In brief, Multi-Reg begins with regions that are altered in copy number in a statistically significant manner (either amplified or deleted, Fig. 1A). It identifies all genes in each region as candidate driver candidate genes (Fig. 1B). Then for each candidate driver it generates its gene expression signatures. i.e. the list of candidate target genes associated with this driver. Comparing the expression signatures between drivers from the same region allows us to focus on significant drivers (Fig. 1C). The final step involves assigning the predicted set of targets for each candidate driver to a distinct expression signature subtype of GBM

(MES, Proliferative, Proneural). This leads to the testable hypothesis about the biological function that each driver gene influences (Fig. 1D).

In this glioblastoma dataset, we started by generating a list of 747 candidate drivers by identifying regions significantly altered in copy number and taking all genes in each region as candidate drivers.

Applying the complete algorithm, detailed below, resulted in identification of 83 drivers, which can explain the expression of 12,125 targets (above 90% of the 13,223 genes expressed in this dataset).

Processing Expression Data

We used gene expression for 427 samples, measured by University of North Carolina, using the G4502A_07 Agilent chip. We only considered samples that passed our quality control and had low batch effects.

Filtering expression samples by quality control

Motivation: In order to get more accurate results we filtered our samples and concentrated on high quality samples.

Details: We first filtered out samples in a manner very similar to that described in the TCGA paper (2). Briefly, our quality control filtering included the following:

(1) *Net Signal range distributions in the red and green channels*. Samples with large differences between the net signal of red and green channel were flagged, as were those with a high signal in the negative control spots in either the red or green channel.

(2) *Presence of Outliers*. Samples with a high percent of non-uniform features (identified as outliers) were flagged.

(3) *Reproducibility of SpikeIns*. SpikeIns are an internal Agilent control comprising RNA at known concentrations added to the microarray. The linearity between known and measured values of the spiked in RNA is measured. If SpikeIns are reproducible, the R^2 values should be close to 1. Any sample with R^2 less than 0.9 for the SpikeIns was flagged.

Samples were grouped into batches by processing source and time. Batches with a high percentage of flagged samples were removed completely, while batches with low percentage of flagged samples had only the flagged samples removed.

In addition to the quality control steps described above, we also considered the effects each batch had on expression. We evaluated this effect on expression by the number of probes that had ANOVA or Bartlett tests (for mean and standard deviation respectively) with p-value lower than 0.0001. The 4 worst batches were removed.

Results: These filtering steps resulted in 136 samples of high quality gene expression data.

Filtering and unifying expression probes

Motivation: Most genes on the Agilent arrays are measured by more than one probe on the array. We wanted to go from multiple values for each gene to one value per gene, by averaging the probe values.

Details: After filtering batches and samples, we used the ANOVA or Bartlett tests described above to remove probes that still had a substantial batch effect (p-value smaller than 0.0001). These probes were removed from all batches, while keeping all other probes. We then normalized each batch by subtracting the mean expression value of each probe for that batch.

We removed remaining probes if they were not differentially expressed across the different tumor samples (standard deviation smaller than 0.3). We averaged the expression values of multiple probes for the same gene if they agreed and removed all genes measured by inconsistent probes.

We then normalized the expression values of each gene to mean of zero and a standard deviation of one across all samples.

Results: This resulted in a final set of 13,223 genes, measured across 136 microarray samples.

Removing cis effects

Motivation: The expression of each gene is determined both by its copy number (known as *cis-regulatory* effect) and by the regulatory effect of other genes, including drivers (*trans-regulatory* effects). To remove the effect of each gene's copy number, we isolate and separate the *cis-regulatory* effect from the *trans-regulatory* effect, and concentrate on the latter.

Details: We first modeled the gene expression of each gene (g_i) as a linear function of its own copy number (CN_i), and then treated the residual error as the trans-regulatory effect.

That is, we first modeled each gene, g_i as

$$g_i = B_0 + B_1 * CN_i$$

where B_0 and B_1 are constant coefficients that give the best fit, and then calculated the residuals R_i as

$$R_i = g_i - B_0 - B_1 * CN_i$$

To do this, we first calculated the DNA copy number for each gene. Of the 13,223 genes modeled, 12,645 genes have a known chromosomal location using the hg18 build of the genome, and their copy number could be calculated (See Processing Copy Number Data below). A defined genomic location (and therefore copy number) was not available, when using hg18, for 578 genes. For these 578 genes, the original expression (g_i) was taken to represent the "residual", i.e. the trans-regulatory effect. The residuals were normalized to mean of zero and a standard deviation of one for each gene.

Results: For all further stages, when modeling the driver-target relationship we used these normalized residuals as the expression of target genes, but used the original total expression (g_i) of drivers. This was done because the effect of drivers is mediated by the

change in the expression of each driver, regardless of whether this change happened due to cis- or trans-regulation of the drivers.

Processing Copy Number Data

In addition to expression data, we used copy number data from 431 glioblastoma samples collected by the TCGA Glioblastoma project (2) and measured on Agilent CGH arrays by Harvard Medical School.

Copy number of a gene in a specific sample was determined using the maximum or minimum copy-number value (for amplification and deletion, respectively) of relevant markers on the microarray. In general, for each gene we considered all markers that overlap the gene's chromosomal location, as defined by the hg18 build of the genome. If the microarray contained no measurements that overlap this gene we took the copy-number value for the single measurement on the microarrays closest to this gene's chromosomal location. Using the same thresholds as in (3), if the copy number of the gene is above 0.12, the gene is marked as amplified, if the copy number value is below -0.12, it is marked as deleted.

Identification of genes significantly aberrant in copy number

Motivation: First, we aimed to identify genes that are significantly aberrant in copy number (either amplified or deleted) in multiple tumors and consider these genes as candidate drivers. We expected many driver genes to be contained in this candidate list.

Details: We applied the JISTIC algorithm (4) and defined all genes and regions with a q-value threshold below 0.01 as significant.

Result: This resulted in 128 significantly amplified regions (containing 346 genes) and 110 deleted regions (containing 404 genes), giving a total of 747 aberrant genes (3 genes were identified as both amplified and deleted).

Integrating data types and selecting candidate Drivers

At this stage we have 136 samples for which both high quality gene expression and copy number data are available. We now combine the data from these two sources in to improve our ability to identify drivers.

Filtering copy number aberrant genes by differential expression

Motivation: As an initial filter, we required candidate drivers to be differentially expressed across the different tumor samples. This filtering step removes genes that are expressed at a constant level across all tumor samples. This filter removes, among others, genes that are not expressed at all and are therefore unlikely to be drivers.

Details: We defined differentially expressed genes as genes whose expression varied with standard deviation greater than 0.3.

Results: This step resulted in 462 candidate driver genes that reside in significantly amplified or deleted regions, and have variable expression.

Integrating Expression and Copy number

Motivation: We expect that if alteration of copy number for a specific gene was a driving event, then the change in copy number would influence the change in gene expression for that gene. Therefore, we further narrowed down the candidate set, by focusing on genes whose expression is significantly altered by their amplification or deletion status.

Details: We used a scoring method described in the GSEA algorithm (5). We applied this method to each gene, testing if samples where the gene is amplified or deleted have

higher or lower expression respectively than all other samples. Candidate genes that received a statistically significant GSEA score ($p < 0.05$) are considered overexpressed when amplified or underexpressed when deleted. Genes which are underexpressed when amplified or overexpressed when deleted were removed.

Results: 249 genes in total passed this filtering step (126 amplified, 180 deleted, 57 overlap) and were used in the following stages.

Integrating mutated genes

Motivation: Not all driving events are mediated through an effect on driver copy number. To increase our chances of identifying driver genes with no significant copy number alternations, we added mutated genes to the list of candidates.

Details: We downloaded the list of mutations present in each sample from cbio (6). We found 25 genes that are mutated in at least 4 samples, and added them to the list of candidate drivers, except for one mutated gene that is not expressed at all in our samples. Thus, we added 24 genes - an additional 10% of the 249 genes selected by copy number.

Result: This raised the total number of candidate driver genes to 267 genes (since some of the amplified or deleted genes overlap with the mutated genes). Genes added at this step included the well-known oncogene PIK3CG and the gene controlling cell cycle checkpoints CHEK2. PTEN was mutated in our samples, but we had already identified it as a deleted gene.

Associating Candidate Drivers to Targets

Identifying possible driver-target relationships using Mutual Information (MI)

Motivation: We assume that if target genes are affected by the driver biologically, then they will be associated with this driver statistically. We used Mutual Information (MI) to identify associations between each driver and candidate target genes. MI can identify linear and non-linear relationships because it does not require linearity, unlike other methods of association such as Pearson correlation.

Details: Mutual Information (MI) is an information-theory measure of the mutual dependence of two random variables. That is, MI is high when the value of one variable is well predicted by the value of the other. We used the adaptive partitioning estimator, as described in the work by Lian and Wang (7) to estimate Mutual Information (MI).

To identify genes associated to each candidate driver, MI was calculated between both the copy number and gene expression for every candidate driver and the expression of each one of the 13233 genes.

To evaluate which values of MI are significant we use a non-parametric approach to generate a null distribution. We generated 1000 random permutations of each driver and repeated the procedure on the permuted data. Setting a p-value threshold of 0.05, we considered all driver-target combinations that had a smaller p-value as significant.

Results: This step identified lists of candidate targets for each and every driver. Each driver has two lists - one list of targets associated with the driver's copy number, and one list of targets associated with the driver's expression.

Filtering candidate drivers by number of targets

Motivation: We expect real drivers to have a large impact on cellular phenotype and therefore to be associated with a very large number of targets. Therefore, to distinguish between real drivers and spurious MI associations, we kept only the driver genes that were associated with more genes than would be expected by chance.

Details: We kept only drivers where the number of targets was above five percent of the total number of genes expressed, which in this case means above 662 genes out of 13,223.

Results: We calculated the number of targets, as described above, for both copy number and gene expression of each of the candidate drivers. We kept candidate drivers only if both sets of targets were larger than 662. This step resulted in the identification of 213 candidate drivers.

Removing spurious correlations

Motivation: Chromosomal locations that are amplified or deleted in cancer are usually big and contain many genes, where all genes in the same region have similar or identical copy number. But only few of the genes in a given region are likely to be drivers, the rest of the genes most probably are passengers, who might spuriously score well, due to their copy number induced correlation with the drivers. We expect the "real" driver or drivers to predict the expression of targets best.

The driver or drivers whose expression explained a large enough number of candidate targets (above the defined minimum) was selected as the most likely driver or drivers for that region.

Details: We first defined the borders of copy number regions by identifying consecutive areas on the chromosome that are either significantly amplified or significantly deleted (received a q-value less than 0.01 using JISTIC, see above). Genes that have high Mutual Information between their expression and the copy number of any candidate driver in a specific region were identified as tentative targets.

For each target, all possible drivers in the same region receive a score, (see below). The driver with the maximal score for this target is identified, and this can be counted as a "vote" for that driver. The number of targets that "voted" for each driver is counted, and drivers with the number of votes larger than a minimum (662 genes, see above) were picked as real drivers.

Results: Limiting the driver candidates to only drivers with a minimal number of votes resulted in 80 possible drivers, in 69 different chromosomal regions.

Adding mutated genes

Motivation: Mutations can also affect cancer genes, either activating them or inactivating them, in combination with expression. We wanted to consider genes that had less than the minimal number of votes if they were mutated in many of the samples.

Details: Genes that did not have the minimum number of votes but had mutations in more than 5 percent of the glioblastoma samples were added to the list of driver genes above.

Results: This resulted in the addition of 3 genes, including EGFR, NF1 and TP53. Two other mutated genes (RB1 and PTEN) were already included in our list of drivers. Adding mutated genes increased the number of drivers to 83.

Combining candidate drivers and targets to generate modules

While a driver gene may be informative for many target genes, these target genes do not necessarily share an expression pattern (see Fig. S1A). Mutual Information leads us to identify multiple expression patterns in the group of target genes. In order to identify patterns of expression that are more likely to be biologically meaningful, we grouped the target genes with similar behavior using Normal Gamma (see below). This resulted in the identification of groups of co-expressed target genes (called modules).

Scoring all possible split points for a driver

Motivation: There are many ways to split the target gene population in two. We wanted to identify good splits and focus on them. The basic question we ask, repeatedly, is this: given a driver and its target, does the group of target values behave as if it was chosen from one Normal Gaussian population, or is it better to split the target values into two groups, chosen from two separate Normal Gaussian populations (where the two groups are split by a given driver value)?

Details: We ordered the samples according to the value of a given driver (Fig. S1A) and observed the gene expression of all candidate targets (Fig. S1B).

We then began our search for good splits by using each and every driver value as a possible split point. That is, if the driver value is larger than X , we will assign the target to one population; if not, we will assign the target to the other population. We ignored possible splits that would result in a population smaller than a defined minimum (10 samples in our Glioblastoma analysis).

Results: Scoring all possible split points for a given driver (Fig. S1D) allowed us to identify the best possible split for each driver-target combination (Fig. S1E). Target

genes whose maximum score was negative were removed from further analysis, since they more likely to have been drawn from a single Gaussian population.

Selecting split points for groups of targets

Motivation: While we have identified split points that are optimal for each target, our goal was to select split points that would be good for groups of target genes, where these split points identify modules.

Details: Combining the two matrices described above generated two vectors for every driver - sum of scores, and number of targets.

The first vector (Fig. S1H, red) contains the sum of scores of all targets for each and every split point. This is a measure of how good is the split point is for the targets in aggregate - a split point that was good for one gene but bad for all other genes will receive a lower score. This vector was smoothed using cubic smoothing splines, whose derivatives were used to find local maxima.

The second vector (Fig. S1H, blue) contains how many targets selected each split point as a best split point. This vector, which gives a "vote" count for split points, was smoothed using a moving window average, using the minimum population size (10 samples, see above) as the window size. The number of votes for each split point was compared to a null distribution of votes, represented by a uniform distribution (Fig. S1H, green).

Group split points that were local maxima for scores and had more gene "votes" than the null distribution were selected as candidate split points. Candidate split points closer to each other than the minimum population size (10 samples) were unified.

Results: We obtained a list of candidate split points for each driver, and a table showing which split point did each target vote for.

Going from split points to modules

Motivation: Having identified candidate split points, we wanted to use these split points to define modules. All target genes that voted for a candidate split point are grouped together in a module. In a sense a module is defined by the existence of a good split point for a group of targets.

Details: We identified all target genes that voted for the candidate split points in every driver. Genes that voted for a split point within 5 samples (minimal population size divided by 2) of a candidate split point were considered to have voted for that split point. All genes that voted for a given split point were grouped as a module. Each module was then divided into two. All target genes whose expression has a positive correlation with the driver values constitute an upregulated module, while all targets with a negative correlation constitute a downregulated module.

Results: Identifying the target genes based on split points resulted in the identification of several groups of co-expressed target genes, called modules, per driver.

Unifying modules

Motivation: If we end up with many small and very similar modules, it will be difficult to understand what each of them might mean biologically. We want to end up with the smallest number of different modules for each driver. In order to limit the number of modules, we unified, for each driver, modules whose expression is similar.

Details: For each pair of modules from the same driver, we calculated the Normal Gamma score of all the genes in each module. We compared the sum of these two scores to the Normal Gamma score of module containing all genes belonging to at least one of the original modules. If the two modules have a similar expression pattern, the Normal Gamma score of the combined module will be close to (or even larger than) the sum of scores for the two separate modules.

We unified modules from the same driver with
 $\text{ScoreUnified} > \text{Score1} + \text{Score2} - \text{SplitPenalty}$

Where we penalized close split points using the following function:

$$\text{SplitPenalty} = \max(\text{Score1}, \text{Score2}) / \text{abs}(\text{split1} - \text{split2})$$

That is, the maximum score is divided by the number of driver values that separate the split points. In this manner, very close split points are much more heavily penalized, since it makes less sense to keep both of them and generate two separate but only slightly different modules

If there were more than two split points, the algorithm merged them recursively, recalculating the score after merging as necessary.

Results: After processing all drivers, this resulted in 83 candidate drivers and 199 modules - most drivers had one module upregulated and one module downregulated, while only 25 drivers had more than one module of the same type

Normal Gamma Score function

We use a Bayesian scoring approach that maximizes the overall joint probability of both the data and of the model structure. If D represent the data and S represent the structure of the network, then the scoring function is expressed as $\log P(D,S) = \log P(D|S) + \log P(S)$,

where the first term is the likelihood of the data for a given model (in the Bayesian approach we integrate over all possible model parameters) and the second term is the prior on the model structure.

Following the Module Networks approach (8) we use the Normal Gamma distribution (also known as a one dimensional Gaussian-Wishart distribution) for our likelihood function, see Segal et al (9) for full details. Normal Gamma gives a higher score to data with lower variance, which we use to find splits that create two different contexts representing two distinct behaviors. The Normal Gamma score is described below:

$$\begin{aligned} & \text{NormalGamma}(Leaf, \lambda, \alpha) : \\ & N = \text{Size}(Leaf) \\ & \beta = \text{Max}\left(1, \frac{\lambda * (\alpha - 2)}{\lambda + 1}\right) \\ & \beta^+ = \beta + \frac{\text{Var}(Leaf) * N}{2} + \frac{N * \lambda * \overline{Leaf}^2}{2 * (N + \lambda)} \\ & \alpha^+ = \alpha + \frac{N}{2} \\ & \text{Score} = -N * \ln(\sqrt{2\pi}) + \frac{\ln\left(\frac{\lambda}{\lambda + N}\right)}{2} + \ln(\Gamma(\alpha^+)) - \ln(\Gamma(\alpha)) + \alpha * \ln(\beta) - \alpha^+ * \ln(\beta^+) \end{aligned}$$

Leaf is a vector of gene expression values that appear in samples where the candidate driver is above or below the split point value. α, λ and β are parameters. A split is scored

by comparing the score of the data without any splits to the score of the data split into two populations by the driver value.

$$\text{NormalGammaScore} = \text{NormalGamma}(\text{Left_Leaf}) + \text{NormalGamma}(\text{Right_Leaf}) - \text{NormalGamma}(\text{Entire_data})$$

If we have split correctly, the resulting Normal distributions on both sides of the split will have smaller standard deviation than that of a Normal distribution including all data without the split. In this case, the score will be above zero, which means the data is more likely to have come from two Normal distributions than from one Normal distribution.

Analyzing the mouse microarrays

Data processing

We log₂ transformed the microarray data and normalized the transformed data using Quantile Normalization. We removed probes expressed at very low levels (less than 7.5), saturated and non-uniform probes. We unified consistent probes measuring the same gene and removed inconsistent probes, which resulted in 7759 expressed genes.

The glioblastoma signatures (10) were translated from the human to the mouse genome using Homologene release 66. Genes that had more than one match in either mouse or human were removed from further analysis.

Analyzing the response of glioblastoma signatures to RHPN2 overexpression

We normalized each treatment sample by calculating the log of the ratio of the data from each treatment sample and the data generated from the control sample done in parallel with it. We averaged all samples expressing WT RHPN2 protein to get one value per gene for this condition. We used Gene Set Enrichment Analysis (GSEA) (5) to test whether the genes associated with Glioblastoma signatures were significantly upregulated or downregulated. We used the standalone version of GSEA 2.07, available from <http://www.broadinstitute.org/gsea/> and ran it with the default parameters and 100,000 permutations.

References

1. Akavia UD, Litvin O, Kim J, Sanchez-Garcia F, Kotliar D, Causton HC, et al. An Integrated Approach to Uncover Drivers of Cancer. *Cell*. 2010;143:1005-17.
2. Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*. 2008;455:1061-8.
3. Vega FM, Ridley AJ. Rho GTPases in cancer cell biology. *FEBS Lett*. 2008;582:2093-101.
4. Sanchez-Garcia F, Akavia UD, Mozes E, Pe'er D. JISTIC: Identification of Significant Targets in Cancer. *BMC Bioinformatics*. 2010;11:189.
5. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*. 2005;102:15545-50.
6. Cerami E, Gao J, Dogrusoz U, Gross BE, Sumer SO, Aksoy BA, et al. The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer Discov*. 2012;2:401-4.
7. Liang KC, Wang X. Gene regulatory network reconstruction using conditional mutual information. *EURASIP J Bioinform Syst Biol*. 2008:253894.
8. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature genetics*. 2003;34:166-76.
9. Segal E, Pe'er D, Regev A, Koller D, Friedman N. Learning module networks. *J Mach Learn Res*. 2005;6:557-88.

10. Carro MS, Lim WK, Alvarez MJ, Bollo RJ, Zhao X, Snyder EY, et al. The transcriptional network for mesenchymal transformation of brain tumours. *Nature*. 2010;463:318-25.

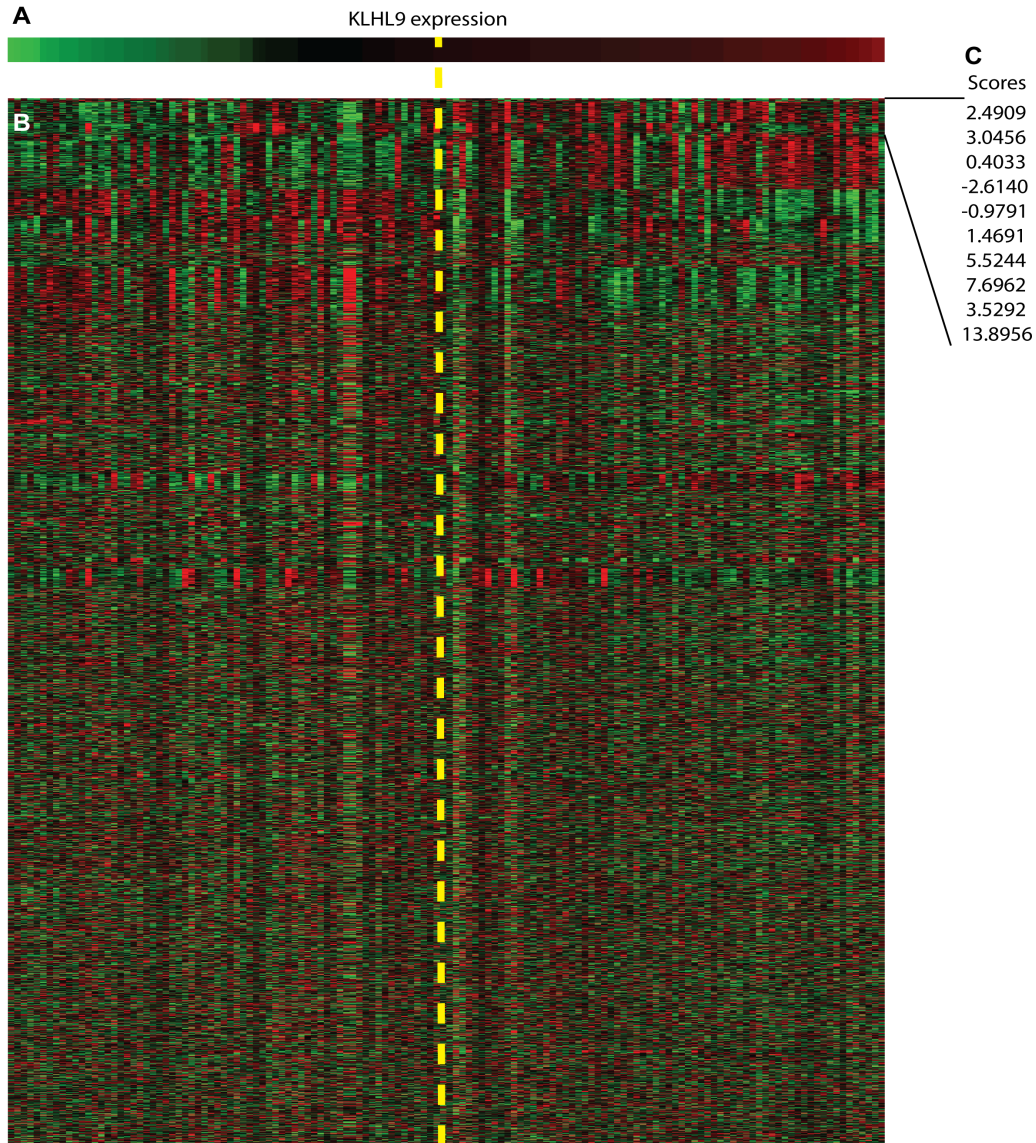


Figure S1 - Illustration of the Module Generating process

Figure S1A-C - Gene expression of all targets as identified by MI (B), sorted by the driver gene expression values (A). Each row represents a target, each column represents a sample, and each cell represents the expression value of this target-sample combination. Red, Green and Black represent overexpression, underexpression and average expression, respectively. A representative split point is shown by the dotted yellow line (B), where the Normal Gamma scores for this split point for the first 10 genes are shown to the right (C).

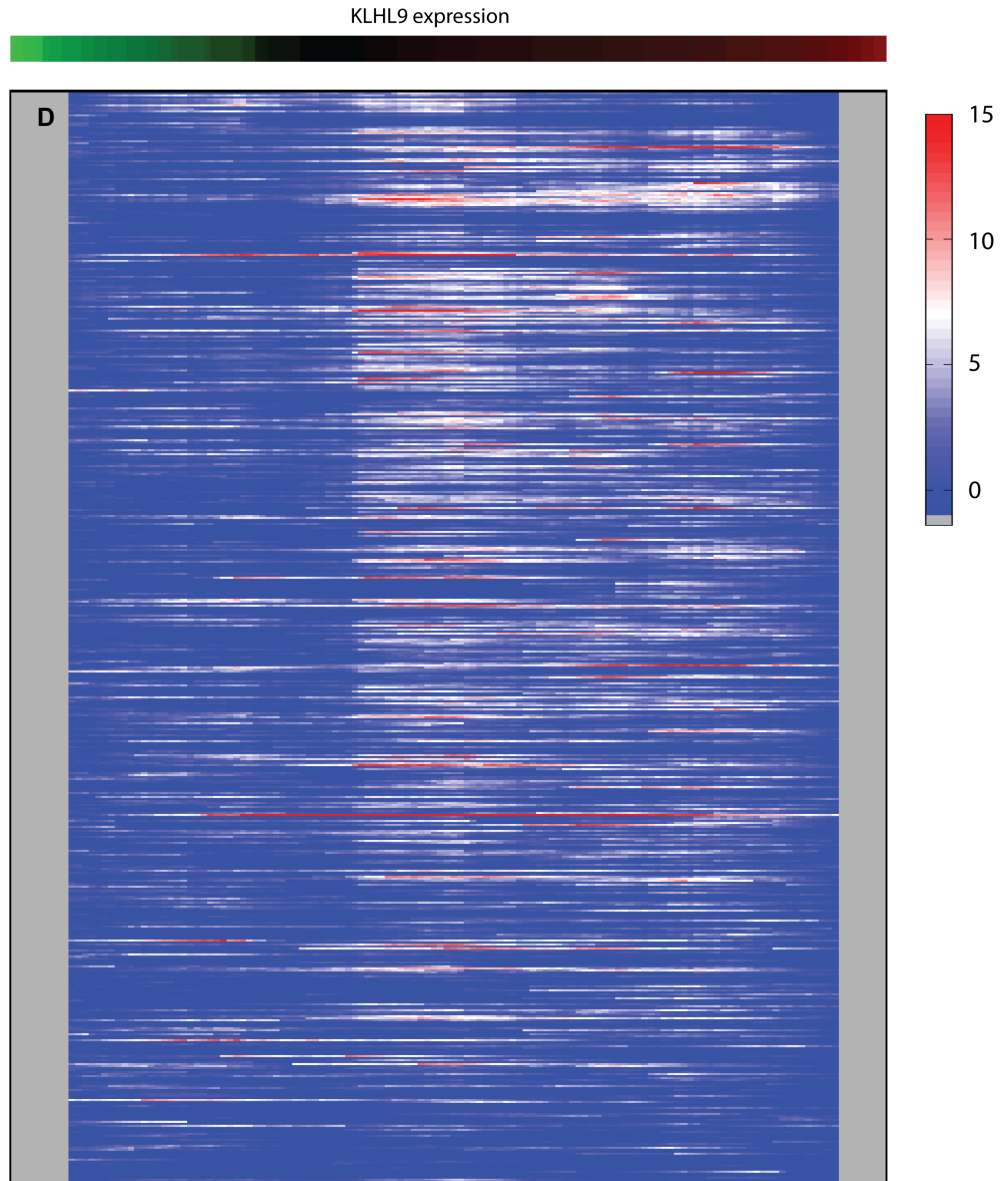


Figure S1D - Normal Gamma score for each split point, same targets as shown in Fig. S1A. Each column represents a choice of split point defined by the driver expression, each row represents a target, and each cell represents the Normal Gamma score given to this combination of split point and target. Red and blue represent high and low scores, while grey represents driver values not considered as split points.

KLHL9 expression



Figure S1E - best split point for each target. Zero indicates that this split-target combination was not maximal for this target, while 1 indicates that the split was the maximal (white and black, respectively).

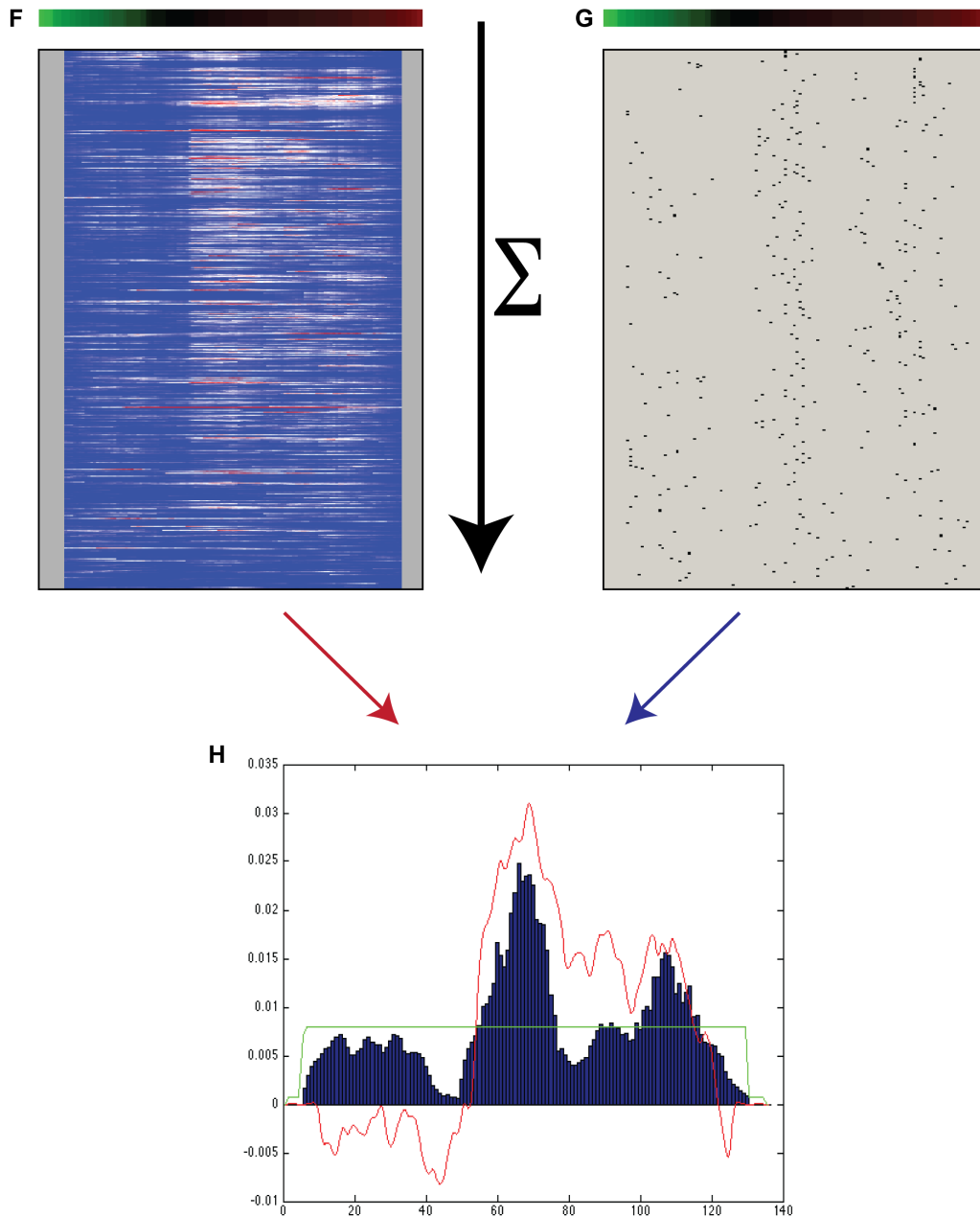


Figure S1F-H - Identifying candidate split points. Summarizing the score matrix (F) and the best split point indicator (G) results in two vectors which are used for split point identification (H). Red represents the sum of scores and indicates the suitability of a split point for a group of targets in aggregate. Blue represents the number of genes voting for each split point, and green represents a null distribution for the votes.

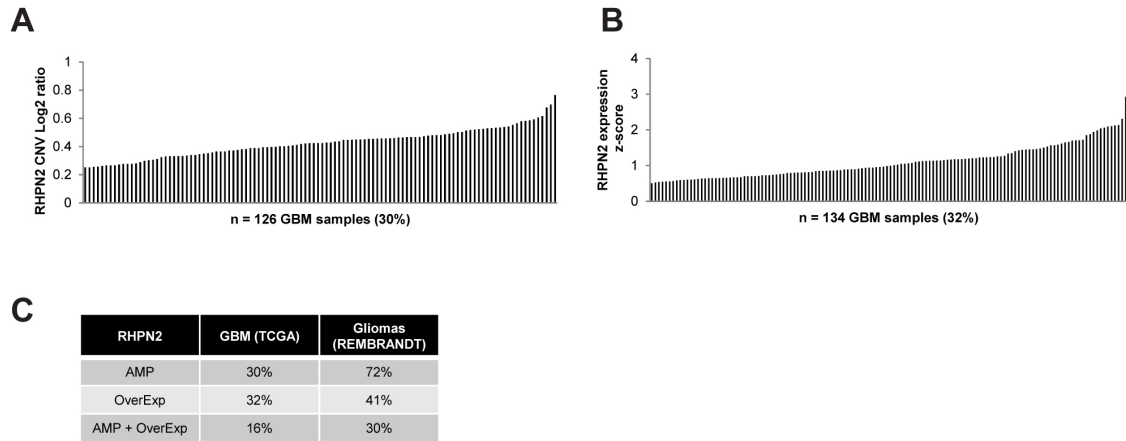


Figure S2

Figure S2. *RHPN2* CNV and Expression in GBM.

(A) and (B) *RHPN2* CNV and Expression plots of GBM samples from TCGA dataset.

(C) *RHPN2* Amplification (AMP), Overexpression (OverExp) and direct correlation AMP and OverExp percentages of GBM samples from TCGA and glioma samples from Rembrandt database.

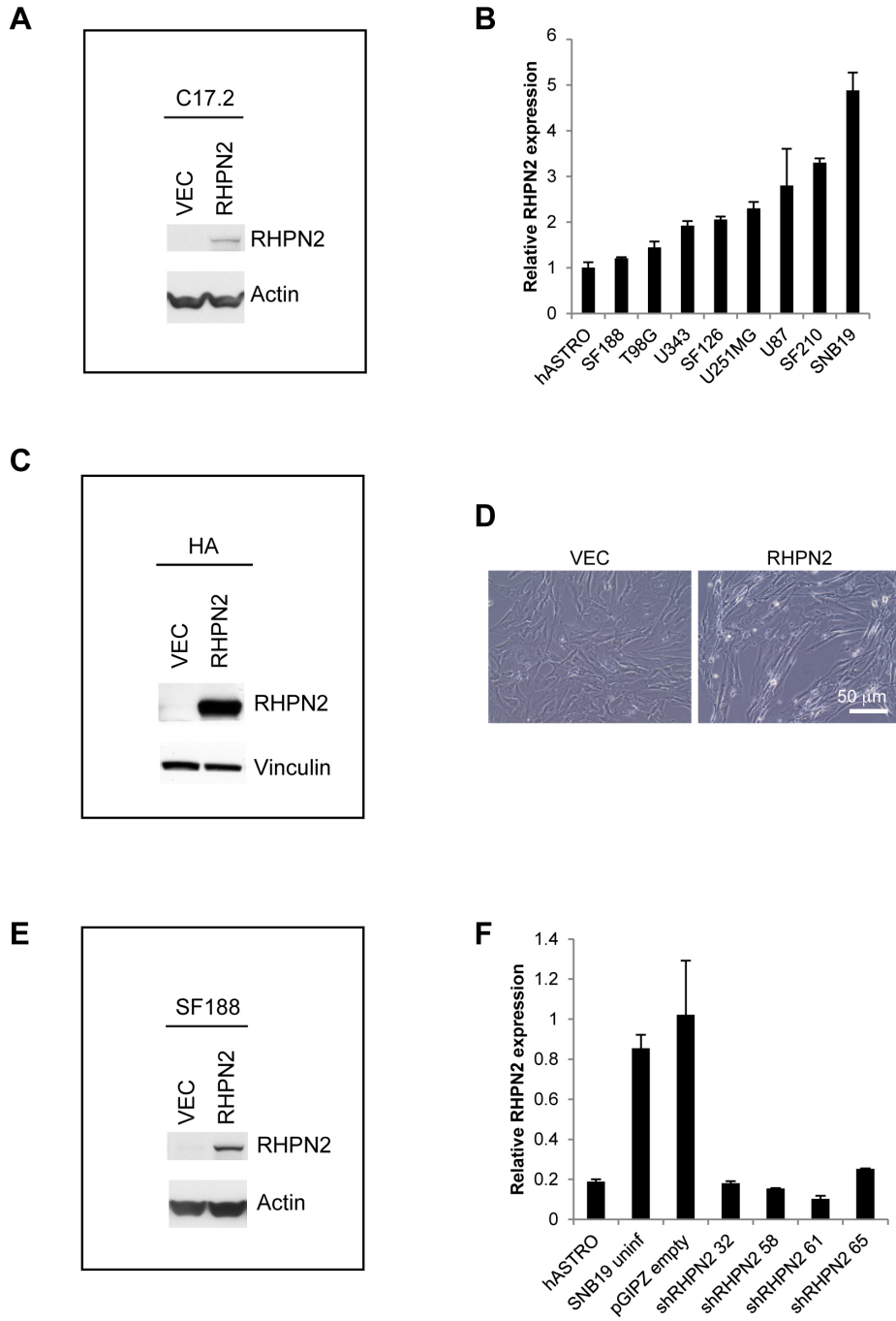


Figure S3

Figure S3. *RHPN2* expression in different cell lines.

(A) Western Blot analysis of ectopic expression of *RHPN2* in C17.2 NSCs.

(B) qRT-PCR analysis of *RHPN2* expression in different glioma cell lines and immortalized human astrocytes (hASTRO).

(C) Western Blot analysis of ectopic expression of *RHPN2* in primary Human Astrocytes (HA).

(D) Morphology of HA upon pLOC VEC and *RHPN2* infection. Scale bar: 50 μm .

(E) Western Blot analysis of ectopic expression of *RHPN2* in SF188 cell line.

(F) Evaluation of *RHPN2* silencing efficiency in SNB19 cells, by qRT-PCR analysis. pGIPZ is control vector; sh*RHPN2* 32, 58, 61, 65 are four different shRNA constructs targeting *RHPN2*.

Table S1. List of primers

Primers mesenchymal genes mouse	Sequence (5'-3')
mActa2_f	GGACGTACAACCTGGTATTGTGC
mActa2_r	CGGCAGTAGTCACGAAGGAAT
mSerpine1_f	CATCCCCCATCCTACGTGG
mSerpine1_r	CCCCATAGGGTGAGAAAACCA
mltga7_qPCR_F1	CTGCTGTGGAAGCTGGGATTC
mltga7_qPCR_R1	CTCCTCCTTGAAGTCTGTCTG
mOsmr_qPCR_f1	CATCCCGAAGCGAAGTCTTGG
mOsmr_qPCR_r1	GGCTGGGACAGTCCATTCTAAA
mGapdh_f	TGACCACAGTCCATGCCATC
mGapdh_r	GACGGACACATTGGGGGTAG
mCtgf_f	GGGCCTCTTCTGCGATTTTC
mCtgf_r	ATCCAGGCAAGTGCATTGGTA
mActn1_f	GACCATTATGATTCCCAGCAGAC
mActn1_r	CGGAAGTCCTCTTCGATGTTCTC
m18s_f	TCAAGAACGAAAGTCGGAGG
m18s_r	GGACATCTAAGGGCATCACA
mC1rl_qPCR_f1	TCGTCCTCCAAGAGCAAAATC
mC1rl_qPCR_r1	TAAGTGTTCCCTGTCTGGTCTG
mTNC_f	ACGGCTACCACAGAAGCTG
mTNC_r	ATGGCTGTTGTTGCTATGGCA

Table S2. Results of Multi-Reg. Chromosomal regions and candidate driver genes identified by Multi-Reg. The table includes information for chromosomal position, CNV, mutations, strand, gene position, GBM subclass and relative p-value. *RHPN2* is highlighted in yellow.